

BLDE Association's
S.B. Arts and K.C.P. Science College, Vijayapur

Department of Computer Science

Lab Manual

Course Code: 21BSC1C1CS1P

Course Title: C Programming Lab

Credits 02

Formative Assessment Marks: 15

Summative Assessment Marks: 35

2023-24

Instructions :**Practice Labs**

1. The following activities be carried out/ discussed in the lab during the initial period of the semester.

- 1 Basic Computer Proficiency
 - a. Familiarization of Computer Hardware Parts
 - b. Basic Computer Operations and Maintenance.
 - c. Do's and Don'ts, Safety Guidelines in Computer Lab
2. Familiarization of Basic Software – Operating System, Word Processors, Internet Browsers, Integrated Development Environment (IDE) with Examples.
3. Type Program Code, Debug and Compile basic programs covering C Programming fundamentals discussed during theory classes.

Part A:

- 1 Write a C Program to read radius of a circle and to find area and circumference
- 2 Write a C Program to read three numbers and find the biggest of three
- 3 Write a C Program to demonstrate library functions in math.h
- 4 Write a C Program to check for prime
- 5 Write a C Program to generate n primes
- 6 Write a C Program to read a number, find the sum of the digits, reverse the number and check it for palindrome
- 7 Write a C Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers
- 8 Write a C Program to read percentage of marks and to display appropriate message (Demonstration of else-if ladder)
- 9 Write a C Program to find the roots of quadratic equation (demonstration of switch-case statement)
- 10 Write a C program to read marks scored by n students and find the average of marks (Demonstration of single dimensional array)
- 11 Write a C Program to remove Duplicate Element in a single dimensional Array
- 12 Program to perform addition and subtraction of Matrices

//Program 1:**/*Program to read radius of a circle and to find area and circumference */**

```
#include<stdio.h>
int main( )
{
    float r, area, circumference;
    printf("enter radius \n");
    scanf("%f", &r );

    area=3.142 * r * r;
    circumference = 2 * 3.142 * r ;
    printf("Area of Circle = %f\n", area);
    printf("Circumference of Circle = %f\n", circumference);
}
```

//Program 2:**/* Program to find maximum between three numbers*/**

```
#include<stdio.h>

int main( )
{
    int a,b,c;
    printf("Enter the numbers\n");
    scanf("%d%d%d",&a,&b,&c);

    if(a>b)
    {
        if(a>c)
            printf("%d is maximum",a);
        else
            printf("%d is maximum",c);
    }
    else if(b>c)
        printf("%d is maximum",b);
    else
        printf("%d is maximum",c);
}
```

//Program 3:

//C Program to demonstrate library functions in math.h

```
#include <stdio.h>
#include <math.h>

int main ( )
{
double x,y, r1, r2,r3,r4,r5;
x = 2.7;

r1 = log(x);
printf("log(%lf) = %lf", x, r1);

    r2 = log10(x);
printf("log10(%lf) = %lf\n", x, r2);

    r3= sqrt(x);
printf("Square root of %lf is %lf\n", x,r3);

y=3;
r4= pow(x, y);
printf("Value %lf ^ %lf = %lf\n", x, y, r4);

r5= exp(x);
printf("The exponential value of %lf is %lf\n", x, r5);

return(0);
}
```

//Program 4:**//C Program to check for prime**

```
#include <stdio.h>
int main( )
{
    int n, i, flag = 0;
    printf("Enter a positive integer: ");
    scanf("%d", &n);

    for (i=2; i<=n/2; ++i)
    {
        if (n%i==0)
        {
            flag=1;
            break;
        }
    }

    if(n==1)
    {
        printf("1 is neither prime nor composite.");
    }
    else
    {
        if(flag==1)
            printf("%d is not a prime number.", n);
        else
            printf("%d is a prime number.", n);
    }

    return 0;
}
```

//Program 5:

//C Program to generate n prime numbers

#include<stdio.h>

int main()

{

int n, count, flag, i=2, j;

printf("Enter how many prime numbers? \n");

scanf("%d", &n);

/* Generating prime numbers */

count=1;

while(count<=n)

{

flag = 0;

for(j=2; j<=i/2; j++)

{

if(i%j==0)

{

flag=1;

break;

}

}

if(flag==0)

{

printf("%d\t", i);

count++;

}

i++;

}

return 0;

}

//Program 6:

/*C Program to read a number, find the sum of the digits, reverse the number and check it for palindrome */

/*Problem Description

This C program accepts an integer, reverse it and also checks if it is a palindrome or not.

Problem Solution

1. Take the number which you have to reverse as the input.
2. Obtain its quotient and remainder.
3. Multiply the separate variable with 10 and add the obtained remainder to it.
4. Do step 2 again for the quotient and step 3 for the remainder obtained in step 4.
5. Repeat the process until quotient becomes zero.
6. When it becomes zero, check if the reversed number is equal to original number or not.
7. Print the output and exit.*/

```
#include <stdio.h>
int main()
{
    int num,temp,remainder,reverse=0;
    printf("Enter an integer \n");
    scanf("%d", &num);

    /* original number is stored at temp */

    temp=num ;
    while(num>0)
    {
        remainder=num%10;
        reverse=reverse*10+remainder;
        num=num/10;
    }
    printf("Given number is = %d\n", temp);
    printf("Its reverse is = %d\n", reverse);

    if (temp==reverse)
        printf("Number is a palindrome \n");
    else
        printf("Number is not a palindrome \n");
}
```

//Program 7:

/*Write a C Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers */

```
#include <stdio.h>
int main()
{
    int num, sum=0;
    printf("Enter the numbers to be added\n");
    do
    {
        scanf("%d", &num);
        if(num<0||num==999)
            continue;
        sum=sum+num;
    }while(num!=999);

    printf("sum=%d",sum);
}
```

//Program 8:

/*Write a C Program to read percentage of marks and to display appropriate message (Demonstration of else-if ladder)*/

```
#include<stdio.h>
int main()
{
    int per;
    printf("Enter your Percentage ");
    scanf("%d", &per);
    if(per<0 || per>100)
        printf("Wrong Entry");
    else if(per>=35 && per<50)
    {
        printf("Grade F");
    }
    else if(per>=50 && per<60)
    {
        printf("Grade D");
    }
    else if(per>=60 && per<70)
    {
        printf("Grade C");
    }
    else if(per>=70 && per<80)
    {
        printf("Grade B");
    }
    else if(per>=80 && per<90)
    {
        printf("Grade A");
    }
}
```

```
    else if(per>=90&&per<=100)
    {
        printf("Grade A+");
    }
    else
        printf("Fail");
}
```

//Program 9:

/*C Program to find the roots of quadratic equation (demonstration of switch-case statement)*/

The mathematical representation of a Quadratic Equation is $ax^2+bx+c = 0$.

Condition 1: If discriminant = 0 then Two Equal and Real Roots will exists

$$Root1 = Root2 = \frac{-b}{2a}$$

Condition 2: If discriminant > 0 then Two Distinct Real Roots will exist for this equation

$$Root1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad Root2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Condition 3: if discriminant < 0 then Two Distinct Complex Roots will exist

$$Root1 = \frac{-b}{2a} + i \frac{\sqrt{-(b^2 - 4ac)}}{2a} \qquad Root2 = \frac{-b}{2a} - i \frac{\sqrt{-(b^2 - 4ac)}}{2a}$$

Input : a = 1, b = -2, c = 1

Output : Roots are real and equal
1

Input : a = 1, b = 7, c = 12

Output : Roots are real and different
-3, -4

Input : a = 1, b = 1, c = 1

Output : Roots are complex
-0.5 + i1.73205
-0.5 - i1.73205

```
#include<math.h>
int main()
{
    double a,b,c,x,d,r1,r2;
    int choice;
    printf("Enter the values of a, b and c n");
    scanf("%lf %lf %lf", &a, &b, &c);

    x=(b*b)-4*a*c;
    d=sqrt(x);
    if(a==0)
    {
        printf("Invalid");
        return 0;
    }
    if(d==0)
        choice=1;
    else if(d>0)
        choice=2;
    else
        choice=3;
    switch(choice)
    {
        case 1:  r1=-b/2*a;
                 r2=-b/2*a;
        printf("\n The roots are real and equal and the roots are r1=%lf and r2=%lf\n",r1,r2);
                 break;
        case 2: r1=(-b+d)/2*a;
                 r2=(-b-d)/2*a;
        printf("The roots are real and different and the roots are r1=%lf and r2=%lf \n", r1, r2);
                 break;
        case 3: r1=-b/(2*a)+d;
                 r2=-b/(2*a)-d;
        printf("The roots are imaginary and unequal and the roots are %lf+%lfi\n", r1, r2);
                 break;
    }
    return 0;
}
```

//Program 10:

**/*Write a C program to read marks scored by n students and find the average of marks
(Demonstration of single dimensional array)*/**

```
#include<stdio.h>
int main()
{
    int n, i;
    float a[100], sum=0.0, avg;

    printf("Enter the numbers of elements: ");
    scanf("%d", &n);

    for(i=0; i<n; i++)
    {
        printf("Enter the marks obtained for subject%d:", i+1);
        scanf("%f", &a[i]);
    }

    for(i=0; i<n; i++)
    {
        sum=sum+a[i];
    }
    avg=sum/n;
    printf("Average=%2f", avg);
    return 0;
}
```

//Program 11:**//Write a C Program to remove Duplicate Element in a single dimensional Array**

```
#include<stdio.h>
int main()
{
int i, j, k, flag = 0, a[10], n;
printf("Enter array size: ");
scanf("%d",&n);
for(i = 0;i < n;i++)
scanf("%d",&a[i]);
printf("\n");
printf("Distinct Array Elements are:\n");
for(i=0;i<n;i++)
    {
        for(j=i-1;j>=0;j--)
        {
            if(a[i]==a[j])
            {
                flag=1;
            }
        }

        if(flag==0)
        {
            printf("%d\t", a[i]);
        }
        flag=0;
    }
return 0;
}
```

//Program 12:**//Program to perform addition and subtraction of Matrices**

```
#include<stdio.h>
int main()
{
    int m, n, i, j;
    double A[50][50];
    double B[50][50];
    double C[50][50];
    double D[50][50];

    printf("Enter the size of the matrices:\nNo. of rows (m): ");
    scanf("%d", &m);
    printf("\nNo. of columns(n): ");
    scanf("%d", &n);

    printf("\nEnter the elements of matrix A:\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%lf", &A[i][j]);
        }
    }
    printf("\nEnter the elements of matrix B:\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%lf", &B[i][j]);
        }
    }
    //Addition of two matrices
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
```

```
        C[i][j]=A[i][j]+B[i][j];
    }
}

printf("\nThe sum of the matrices A and B is:\n");
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        printf("%lf \t", C[i][j]);
    }
    printf("\n");
}

//Subtraction of two matrices
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            D[i][j]=A[i][j]-B[i][j];
        }
    }

printf("\n The Subtraction of the matrices A and B is:\n");
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        printf("%lf \t", D[i][j]);
    }
    printf("\n");
}
}
```

PART B:

- 1 Write a C Program to find the length of a string without using built in function
- 2 Write a C Program to demonstrate string functions.
- 3 Write a C Program to demonstrate pointers in C
- 4 Write a C Program to check a number for prime by defining isprime()function
- 5 Write a C Program to read, display and to find the trace of a square matrix
- 6 Write a C Program to read, display and add two m x n matrices using functions
- 7 Write a C Program to read, display and multiply two m x n matrices using functions
- 8 Write a C Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
- 9 Write a C Program to Reverse a String using Pointer
- 10 Write a C Program to Swap Two Numbers using Pointers
- 11 Write a C Program to demonstrate student structure to read & display records of n students.
- 12 Write a C Program to demonstrate the difference between structure &union.

//Program 1:**//Write a C Program to find the length of a string without using built in function**

```
#include <stdio.h>
#include<linux/string.h>
int main()
{
    char str1[50];
    int i, len = 0;

    printf("Input a string : ");
    scanf("%s", str1);

    for (i=0;str1[i]!='\0'; i++)
    {
        len++;
    }
    printf("The string contains %d number of characters. \n", len);
    printf("So, the length of the string %s is : %d\n\n", str1, len);
}
```

//Program 2:

// Write a C Program to demonstrate string functions.

#include<stdio.h>

#include<linux/string.h>

int main()

{

char str1[100];

char str2[100];

char str3[100];

int length;

printf("Enter the first string\n");

scanf("%s",str1);

printf("Enter the first string\n");

scanf("%s",str2);

printf("\n\nString Operations\n");

/*length of given string*/

length=strlen(str1);

printf("Length of first string = %d\n", length);

printf("\nComparing two strings\n");

```
if(strcmp(str1,str2)==0)
{
    printf("They are equal");
}
else
{
    printf("They are not equal");
}

printf("\nCopy first string to new string object\n");
strcpy(str3,str1);
printf("\nCopied string is %s\n",str3);
//Concatenate two strings
strcat(str1,str2);
printf("Concanited string is %s\n",str1);
}
```

//Program 3:**// Write a C Program to demonstrate pointers in C**

```
#include<stdio.h>
int main()
{
    int a=10;
    int *p;
    p=&a;
    printf("Address stored in a variable p is:%p\n", p);
    printf("Value stored in a variable p is:%d\n",*p);
    return 0;
}
```

//Program 4:**// Write a C Program to check a number for prime by defining isprime()function**

```
#include<stdio.h>

int main()
{
    int n;
    int isPrime(int);
    printf("enter the value of n\n");
    scanf("%d", &n);
    isPrime(n);
}

int isPrime(int n)
{
    int flag=0,i;
    for(i=2; i<=(n/2); i++)
    {
        if(n%i==0)
        {
            flag=1;
            break;
        }
    }
    if(flag==1)
        printf("given number is not a prime number");
    else
        printf("given number is a prime number");
}
```

//Program 5:**// Write a C Program to read, display and to find the trace of a square matrix**

#include<stdio.h>

int main()

{

int a[5][5],r,c,i,j,sum=0;

printf("Enter a number of rows and columns:-");

scanf("%d%d",&r,&c);

if(r==c)

{

printf("\nEnter %d*%d elements in matrix:-\n",r,c);

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

scanf("%d",&a[i][j]);

}

printf("The Array Elements are:\n");

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

for(i=0;i<r;i++)

sum=sum+a[i][i];

printf("\nTrace of the matrix = %d", sum);

}

else

printf("Not a square matrix. It is not possible to find trace."); return 0; }

//Program 6:**// Write a C Program to read, display and add two m x n matrices using functions**`#include<stdio.h>``int m=2,n=2;``int AddMatrix(int [m][n],int [m][n]);``int main( )``{``int A[20][20];``int B[20][20];``int i,j;``printf("Enter the elements of matrix A\n ");``for(i=0;i<m;i++)``{``for(j=0;j<n;j++)``{``scanf("%d",&A[i][j]);``}``}``printf("Enter the elements of matrix B\n ");``for(i=0;i<m;i++)``{``for(j=0;j<n;j++)``{``scanf("%d",&B[i][j]);``}``}`

```
AddMatrix(A,B);
}
int AddMatrix(int A[m][n], int B[m][n])
{
    int i,j;
    int C[20][20];
    printf("The resultant matrix formed on addition of the two matrices:\n")

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            C[i][j]=A[i][j]+B[i][j];
            printf("%d\t",C[i][j]);
        }
        printf("\n");
    }
}
```

//Program 7:

/*Write a C Program to read, display and multiply two m x n matrices using functions */

#include<stdio.h>

void multiply(int A[10][10], int B[10][10], int , int);

int main()

{

int A[10][10], B[10][10];

int i, j, k, m, n, p;

printf("Enter the number of rows and columns for 1st matrix\n");

scanf(" %d%d",&m, &n);

printf("Enter the elements of the 1st matrix\n");

for(i=0;i<m;i++)

{

for(j=0;j<n;j++)

{

scanf(" %d",&A[i][j]);

}

}

printf("Enter the elements of the 2nd matrix\n");

for(i=0;i<n;i++)

{

for(j=0;j<n;j++)

{

scanf(" %d", &B[i][j]);

}

}

```
printf("The 1st matrix\n");
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        printf(" %d\t", A[i][j]);
    }
    printf("\n");
}

printf("The 2nd matrix\n");
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        printf(" %d\t", B[i][j]);
    }
    printf("\n");
}

multiply(A, B, m, n);
}

void multiply(int A[10][ 10], int B[10][10], int m, int n)
{
    int C[12][12], i, j, k;
    for(i=0;i<m;i++)
    {
```

```
for(j=0;j<n;j++)
{
    C[i][j]=0;
    for(k=0;k<n;k++)
    {
        C[i][j]=C[i][j]+A[i][k]*B[k][j];
    }
}

printf("The resultant matrix formed on multiplying the two matrices\n");

for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        printf(" %d\t", C[i][j]);
    }
    printf("\n");
}
```

//Program 8:

/*Write a C Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters. */

```
#include<stdio.h>
#include<linux/string.h>
int main( )
{
    char str[100];
    int i, alphabets = 0, digits = 0, splchar = 0, spaces = 0;
    int vowels=0, consonants = 0;
    printf("Enter the string : ");
    scanf("%s", str);
    for(i=0;str[i];i++)
    {
        if ((str[i]>=65 && str[i]<=90) || (str[i]>=97 && str[i]<=122) )
            alphabets++;
    }
    for(i=0; str[i]!='\0'; i++)
    {
        if(str[i]=='a'||str[i]=='e'||str[i]=='i'||str[i]=='o'||str[i]=='u'||str[i]=='A'||str[i]=='E'||str[i]=='I'||str[i]=='
'O'||str[i]=='U')
        {
            vowels++;
        }
        else if((str[i]>='a'&&str[i]<='z')||(str[i]>='A'&&str[i]<='Z'))
        {
            consonants++;
        }
    }
}
```

```
else if(str[i]>='0'&&str[i]<='9')
{
digits++;
}
else if(str[i]==' ')
{
spaces++;
}
else
{
splchar++;
}
}

printf("Alphabets = %d\n", alphabets);
printf("Vowels = %d\n", vowels);
printf("Consonants = %d\n", consonants);
printf("Digits = %d\n", digits);
printf("Vowels = %d\n", vowels);
printf("Spaces = %d\n", spaces);
printf("Special characters = %d", splchar);
return 0;
}
```

//Program 9:**//Write a C Program to Reverse a String using Pointer**

```
#include<stdio.h>
#include <stdlib.h>
#include<linux/string.h>
int main( )
{
char str1[100],str2[100];
char *p1,*p2;
printf("Enter a String\n");
scanf("%s", str1);
p1=str1+strlen(str1)-1;
p2=str2;
while(p1>=str1)
{
    *p2=*p1;
    p2++;
    p1--;
}
*p2='\0';
printf("Original String: %s\n",str1);
printf("Reverse String: %s",str2);
return 0;
}
```

//Program 10:**//Write a C Program to Swap Two Numbers using Pointers**

```
#include<stdio.h>

int main()
{
    int x, y, *a, *b, temp;
    printf("Enter the value of x and y\n");
    scanf("%d%d", &x, &y);
    printf("Before Swapping\n x = %d\n y = %d\n", x, y);
    a = &x;
    b = &y;

    temp = *b;
    *b = *a;
    *a = temp;
    printf("After Swapping\n x = %d\n y = %d\n", x, y);
    return 0;
}
```

//Program 11:

/*Write a C Program to demonstrate student structure to read & display records of n students. */

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    struct stud
    {
        int rno;
        char name[20];
        float percentage;
    };
    int i,n;
    struct stud *info;
    printf("Enter the total number of students");
    scanf("%d",&n);
    info=(struct stud*)malloc(n*sizeof(struct stud));
    printf("Enter the information of n students:\n");
    for(i=0;i<n;i++)
    {
        printf("Enter the roll no of student%d = ",i+1);
        scanf("%d",&(info+i)->rno);
        printf("Enter the name of student%d = ",i+1);
        scanf("%s", (info+i)->name);

        printf("Enter the percentage of student%d = ",i+1);
        scanf("%f",&(info+i)->percentage);
    }
    printf("\nStudent Information :\n");
    printf("Rno      Name Percentage\n");
    for(i=0;i<n;i++)
    {
        printf("%d\t%s\t%f\n", (info+i)->rno, (info+i)->name, (info+i)->percentage);
    }
}
```

```
[bldea@localhost ~]$ gcc ExpB11CProg.c
```

```
[bldea@localhost ~]$ ./a.out
```

```
Enter the total number of students
```

```
3
```

```
Enter the information of n students:
```

```
Enter the roll no of student1 = 12
```

```
Enter the name of student1 = anil
```

```
Enter the percentage of student1 = 89
```

```
Enter the roll no of student2 = 23
```

```
Enter the name of student2 = nikhita
```

```
Enter the percentage of student2 = 90
```

```
Enter the roll no of student3 = 30
```

```
Enter the name of student3 = arun
```

```
Enter the percentage of student3 = 87
```

```
Student Information :
```

```
Rno      Name      Percentage
```

```
12       anil       89.000000
```

```
23       nikhita    90.000000
```

```
30       arun       87.000000
```

//Program 12:

//C Program to demonstrate the difference between structure &union.

Differences:

	STRUCTURE	UNION
Keyword	The keyword struct is used to define a structure	The keyword union is used to define a union.
Size	When a variable is associated with a structure, the compiler allocates the memory for each member. The size of structure is greater than or equal to the sum of sizes of its members.	when a variable is associated with a union, the compiler allocates the memory by considering the size of the largest memory. So, size of union is equal to the size of largest member.
Memory	Each member within a structure is assigned unique storage area of location.	Memory allocated is shared by individual members of union.
Value Altering	Altering the value of a member will not affect other members of the structure.	Altering the value of any of the member will alter other member values.
Accessing members	Individual member can be accessed at a time.	Only one member can be accessed at a time.
Initialization of Members	Several members of a structure can initialize at once.	Only the first member of a union can be initialized.

//Program 12:**//C Program to demonstrate the difference between structure &union.**

```
#include <stdio.h>
#include <linux/string.h>

struct student
{
    int rno;
    float percentage;
    char name[20];
};
union dept
{
    char deptno[10];
    char name[50];
};

int main()
{
    struct student s={18,78,"ABC"};
    union dept u={"C31"};
    printf("structure data:\n Rno: %d\n Percentage: %.2f\n name:
    %s\n",s.rno, s.percentage, s.name);
    printf("\nunion data:\n Dept No.: %s\n name:
    %s\n",u.deptno,u.name);

    // difference two and three
    printf("\nsizeof structure : %ld\n", sizeof(s));
    printf("sizeof union : %ld\n", sizeof(u));
    // difference five
    printf("\n Accessing all members at a time:");
    s.rno=183;
    s.percentage=90;
    strcpy(s.name, "sneha");
    printf("structure data:\n Rno: %d\n Percentage: %.2f\n name:
    %s\n",s.rno, s.percentage, s.name);
    //u.deptno = "C41";
    strcpy(u.name, "Statistics");
    printf("\nunion data:\n Dept No.: %s\n name: %s\n",u.deptno, u.name);
    //difference four

    printf("\nAltering a member value:\n");
```

```
s.rno = 120;
printf("structure data:\n Rno: %d\n Percentage: %.2f\n name:
%s\n",s.rno,s.percentage, s.name);
//u.deptno = "C34";
printf("\nunion data:\n Dept No.: %s\n name: %s\n", u.deptno,
u.name);
}
```

B.L.D.E Association's

S.B. Arts and K.C.P. Science College, Vijayapur

Department of Computer Science

LABORATORY MANUAL

Class : BSc II Semester

Subject : Computer Science (Data Structure using C)

Paper Code :

Part A:

1. Write a C Program to find GCD using recursive function
2. Write a C Program to display Pascal Triangle using binomial function
3. Write a C Program to generate n Fibonacci numbers using recursive function.
4. Write a C Program to implement Towers of Hanoi.
5. Write a C Program to implement dynamic array, find smallest and largest element of the array.
6. Write a C Program to create two files to store even and odd numbers.
7. Write a C Program to create a file to store student records.
8. Write a C Program to read the names of cities and arrange them alphabetically.
9. Write a C Program to sort the given list using selection sort technique.
10. Write a C Program to sort the given list using bubble sort technique.

/* Program to find GCD of given two numbers*/

```
#include<stdio.h>
```

```
main()
```

```
{  
    int nmtr,dmtr,gcd;  
    int compGCD(int,int);
```

```
    printf("Enter two numbers for GCD\n");  
    scanf("%d%d",&nmtr,&dmtr);
```

```
    gcd=compGCD(nmtr,dmtr);
```

```
    printf("GCD for %d and %d is %d",nmtr,dmtr,gcd);
```

```
}
```

```
int compGCD(int n1,int n2)
```

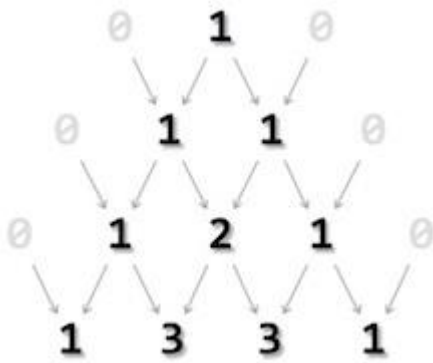
```
{  
    int r;  
    while(n2!=0)  
    {  
        r=n1%n2;  
        n1=n2;  
        n2=r;  
    }
```

```
    return n1;
```

```
}
```

/*C Program to display Pascal Triangle using binomial function*/**Pascal triangle**

Pascal triangle is a triangular array of binomial coefficients. In pascal's triangle, each number is the sum of the two numbers directly above it. It has many interpretations. One of the famous one is its use with binomial equations.



All values outside the triangle are considered zero (0). The first row is 0 1 0 whereas only 1 acquire a space in pascal's triangle, 0s are invisible. Second row is acquired by adding (0+1) and (1+0). The output is sandwiched between two zeroes. The process continues till the required level is achieved.

/*C Program to display Pascal Triangle using binomial function*/

```
#include <stdio.h>
long factorial(int);
int main()
{
    int i, n, j;
    printf("Enter the number of rows \n");
    scanf("%d", &n);
    for (i = 0; i < n; i++)
    {
        for (j = 0; j <= (n - i - 2); j++)
            printf(" ");
        for (j = 0; j <= i; j++)
            printf("%ld ", factorial(i) / (factorial(j) * factorial(i - j)));
        printf("\n");
    }
    return 0;
}

long factorial(int n)
{
    int k;
    long result = 1;
    for (k = 1; k <= n; k++)
        result = result * k;
    return result;
}
```

Output 1:

Enter the number of rows: 5

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

Output 2:

Enter the number of rows: 7

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
```

/*Program to find n Fibonacci numbers using recursion

```
#include<stdio.h>
int main()
{
    int i,n;
    printf("Enter n");
    scanf("%d",&n);
    printf("Fibonacci numbers are:\n");
    for(i=0;i<n;i++)
    {
        printf("%d\n",fib(i));
    }
}

int fib(int n)
{
    if(n==0)
        return 0;
    if(n==1)
        return 1;
    else
        return fib(n-1)+fib(n-2);
}
```

```
[bldea@localhost ~]$ gcc dsprog7.c
[bldea@localhost ~]$ ./a.out
Enter n
10
Fibonacci numbers are:
0
1
1
2
3
5
8
13
21
34
```

```
/* Program to solve Tower of Hanoi problem using recursion.*/
```

```
#include<stdio.h>
```

```
void tower_hanoi( int, char, char, char);
```

```
int main( )
```

```
{
```

```
int  n;
```

```
printf("Enter the number of disks");
```

```
scanf("%d",&n);
```

```
tower_hanoi(n, 'A', 'B', 'C');
```

```
return 0;
```

```
}
```

```
int tower_hanoi(int n , char src, char dest, char tmp)
```

```
{
```

```
if(n>0)
```

```
{
```

```
/* Moving n-1 disks from A to C peg*/
```

```
tower_hanoi(n-1,src,tmp,dest);
```

```
/* Moving n disks from A to B peg*/
```

```
printf("Moves disk %d from %c to %c \n", n, src,dest);
```

```
/* Moving n-1 disks from C to B peg*/
```

```
tower_hanoi(n-1,tmp,dest , src);
```

```
}
```

```
return;
```

```
}
```

/*Write a C Program to implement dynamic array, find smallest and largest element of the array*/

```
#include<stdio.h>

int main()
{
int a[50],i,n,large,small;
printf("How many elements:");
scanf("%d",&n);
printf("Enter the Array:");

for(i=0;i<n;++i)
scanf("%d",&a[i]);
large=small=a[0];
for(i=1;i<n;++i)
{
if(a[i]>large)
large=a[i];
if(a[i]<small)
small=a[i];
}
printf("The largest element is %d",large);
printf("\nThe smallest element is %d",small);

return 0;
}
```

/*Write a C Program to create two files to store even and odd numbers.*/

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int size,a[100],i,n;
    FILE *fp1,*fp2;

    printf("Enter the size : \n");
    scanf("%d",&size);

    fp1=fopen("even", "wb");
    fp2=fopen("odd", "wb");

    if(fp1==NULL || fp2==NULL)
    {
        printf("Error opening file\n");
        exit(1);
    }
    printf("enter the numbers\n");
    for(i=0;i<size;i++)
    {
        scanf("%d",&n);
        if(n%2==0)
            fprintf(fp1,"%d\n", n);
        else
            fprintf(fp2,"%d\n", n);
    }
    fclose(fp1);
    fclose(fp2);
}
```

```
[bldea@localhost ~]$ gcc file_even_odd.c
[bldea@localhost ~]$ ./a.out
Enter the size :
5
enter the numbers
1
2
3
4
5
```

```
[bldea@localhost ~]$ cat even
2
4
```

```
[bldea@localhost ~]$ cat odd
1
3
5
```

```
/*C Program to create a file to store student records*/

#include<stdio.h>
#include<stdlib.h>
int main()
{
    FILE *fp;
    char name[20];

    int m1,m2,m3,m4,m5,rno;

    fp=fopen("StudentRecord.txt","w");
    if(fp==NULL)
    {
        printf("Cannot open file");
        exit(1);
    }
    fprintf(fp,"Role No.\t NAME\t  Kannada\n\t English\t ComputerScience\t Maths\t Physics\n");
    printf("\nEnter name :");
    scanf("%s",&name);
    printf("Enter your Roll No.:");
    scanf("%d",&rno);
    printf("Enter marks for Kannada:");
    scanf("%d",&m1);
    printf("Enter marks for English:");
    scanf("%d",&m2);
    printf("Enter marks for CS:");
    scanf("%d",&m3);
    printf("Enter marks for Maths:");
    scanf("%d",&m4);
    printf("Enter marks for Physics:");
    scanf("%d",&m5);
    fprintf(fp,"%d\t %s\t %d\t %d\t %d\t %d\t %d \n",rno,
        name, m1,m2,m3,m4,m5);
}
fclose(fp);

}
```

```
[bldea@localhost ~]$ gcc student.c
[bldea@localhost ~]$ ./a.out
```

```
Enter name : Sneha
Enter your Role No.:12
Enter marks for Kannada:89
Enter marks for English:90
Enter marks for CS:90
Enter marks for Maths:95
Enter marks for Physics:82
```

```
[bldea@localhost ~]$ cat StudentRecord.txt
```

Roll No.	NAME	Kannada	English	ComputerScience	Maths	Physics
12	sneha	89	90	90	95	82

/*C Program to read the names of cities and arrange them alphabetically.*/

```
#include <stdio.h>
#include <string.h>
void main()
{

    char name[10][8], tname[10][8], temp[8];
    int i, j, n;

    printf("Enter the value of n \n");
    scanf("%d", &n);
    printf("Enter %d names n \n", n);

    for (i = 0; i < n; i++)
    {
        scanf("%s", name[i]);
        strcpy(tname[i], name[i]);
    }

    for (i = 0; i < n - 1 ; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (strcmp(name[i], name[j]) > 0)
            {
                strcpy(temp, name[i]);
                strcpy(name[i], name[j]);
                strcpy(name[j], temp);
            }
        }
    }

    printf("\n-----\n");
    printf("Input NamesSorted names\n");
    printf("-----\n");
```

```
for (i = 0; i < n; i++)
{
    printf("%s\t\t%s\n", tname[i], name[i]);
}
printf("-----\n");
}
```

/*C Program to sort the given list using selection sort technique*/

```
int SelSort(int array[],int n);
int main()
{
int array[100], n,i;
printf("Enter number of elementsn");
scanf("%d", &n);
printf("Enter %d Numbersn", n);
for(i = 0; i < n; i++)
scanf("%d", &array[i]);
SelSort(array,n);
return 0;
}
```

```
int SelSort(int array[], int n)
{
int i, j, position, swap;
for(i = 0; i < (n - 1); i++)
{
position=i;
for(j = i + 1; j < n; j++)
{
if(array[position]>array[j])
position=j;
}
if(position != i)
{
swap=array[i];
array[i]=array[position];
array[position]=swap;
}
}
return 0;
}
printf("Sorted Array:n");
for(i = 0; i < n; i++)
printf("%d\n", array[i]);
}
```

/*Program to sort a list of N elements using Bubble Sort Technique*/

```
#include<stdio.h>
int main()
{
int a[10],i,j,n,temp;
printf("Enter the value of N:\n");
scanf("%d",&n);
printf("Enter the elements:\n");
for(i=0;i<n;i++)
scanf("%d",&a[i]);

for(i=0;i<n;i++)
{
for(j=0;j<n-1;j++)
{
if(a[j]>a[j+1])
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
}
printf("\n sorted order is:\n");
for(i=0;i<n;i++)
printf("%d\n",a[i]);
return 0;
}
```

```
[bldea@localhost ~]$ gcc bubblesort.c
```

```
[bldea@localhost ~]$ ./a.out
```

```
Enter number of elements
```

```
5
```

```
Enter the elements
```

```
67
```

```
45
```

```
90
```

```
12
```

```
56
```

```
Array elements are
```

```
12
```

```
45
```

```
56
```

```
67
```

```
90
```

PART - B

1. Write a C Program to sort the given list using insertion sort technique.
2. Write a C Program to sort the given list using quick sort technique.
3. Write a C Program to sort the given list using merge sort technique.
4. Write a C Program to search an element using linear search technique.
5. Write a C Program to search an element using recursive binary search technique.
6. Write a C Program to implement Stack.
7. Write a C Program to convert an infix expression to postfix.
8. Write a C Program to implement simple queue.
9. Write a C Program to implement linear linked list.
10. Write a C Program to display traversal of a tree

/*Program to sort a list of N elements using Quick Sort Technique*/

```
#include<stdio.h>
int main()
{
    int a[10],low,high,i,n;
    int quicksort(int a[],int low,int high);
    int partttition(int a[],int low,int high);
    printf("enter the value of N:\n");
    scanf("%d",&n);
    printf("enter the elements:\n");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    low=0;
    high=n-1;
    quicksort(a,low,high);
    printf("the sorted elements are:\n");
    for(i=0;i<n;i++)
        printf("%d\n",a[i]);
    return 0;
}

int quicksort(int a[],int low,int high)
{
    int p;
    if(low<high)
    {
        p=partition(a,low,high);
        quicksort(a,low,p-1);
        quicksort(a,p+1,high);
    }
    return;
}

int partition(int a[],int low,int high)
{
    int key,flag,i,j,tmp;
    key=a[low];
    i=low+1;
    j=high;
    flag=1;
```

```
while(flag)
{
    while(key>a[i]&& i<high)
        i=i+1;
    while(key<a[j])
        j=j-1;
    if(i<j)
    {
        tmp=a[i];
        a[i]=a[j];
        a[j]=tmp;
    }
    else
        flag=0;
}
tmp=a[low];
a[low]=a[j];
a[j]=tmp;
return j;
}
```

```
[bldea@localhost ~]$ gcc quicksort.c
[bldea@localhost ~]$ ./a.out
enter the value of N:
5
enter the elements:
67
89
12
54
33
the sorted elements are:
12
33
54
67
89
```

/*Program to sort a list of N elements using Merge Sort Technique*/

```
#include<stdio.h>
int mergesort(int a[],int,int);
int merge(int a[],int low,int mid,int high);

int main()
{
    int a[10],i,n,low,high;
    printf("enter the value of N:\n");
    scanf("%d",&n);
    printf("enter the elements:\n");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    low=0;
    high=n-1;
    mergesort(a,low,high);
    printf("Sorted elements are:\n");
    for(i=0;i<n;i++)
        printf("%d\n",a[i]);
}

int mergesort(int a[],int low,int high)
{
    int mid;
    if(low<high)
    {
        mid=(low+high)/2;
        mergesort(a,low,mid);
        mergesort(a,mid+1,high);
        merge(a,low,mid,high);
    }
    return;
}
```

```
int merge(int a[],int low,int mid,int high)
{
    int i=low,j=mid+1,k=low,c[10];
    while((i<=mid)&&(j<=high))
    {
        if(a[i]<a[j])
        {
            c[k]=a[i];
            k++;
            i++;
        }
        else
        {
            c[k]=a[j];
            k++;
            j++;
        }
    }
    while(i<=mid)
    {
        c[k]=a[i];
        k++;
        i++;
    }

    while(j<=high)
    {
        c[k]=a[j];
        k++;
        j++;
    }
    for(i=low;i<=k-1;i++)
    a[i]=c[i];
    return 1;
}
```

```
[bldea@localhost ~]$ gcc mergesort.c
[bldea@localhost ~]$ ./a.out
Enter number of elements
7
Enter the elements
89
5
67
34
90
12
4
Sorted Elements are
4
5
12
34
67
89
90
```

/*Program to search an element using linear search technique*/

```
#include<stdio.h>

int main()
{
    int a[10],i,n,key,flag=0;
    printf("enter the value of N:\n");
    scanf("%d",&n);
    printf("enter the elements:\n");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    printf("enter the search element:\n");
    scanf("%d",&key);
    for(i=0;i<n;i++)
    {
        if(a[i]==key)
        {
            flag=1;
            break;
        }
    }
    if(flag==1)
        printf("element is found\n");
    else
        printf("element is not found\n");
    return;
}
```

Output :

```
[bldea@localhost ~]$ gcc seqsearch.c
[bldea@localhost ~]$ ./a.out
enter the value of N:
5
enter the elements:
45
67
12
90
55
enter the search element:
12
element is found
```

```
[bldea@localhost ~]$ ./a.out
enter the value of N:
5
enter the elements:
23
11
45
89
33
enter the search element:
10
element is not found
```

/*Program to search for an element in an array using Binary Search */

```
#include<stdio.h>
int main()
{
    int a[10],i,n,X,low,high,mid,flag=0;
    printf("enter the value of N\n");
    scanf("%d",&n);
    printf("enter the elements\n");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    printf("enter the search elements");
    scanf("%d",&X);
    low=0;
    high=n-1;
    while(low<=high)
    {
        mid=(low+high)/2;
        if(a[mid]==X)
        {
            flag=1;
            break;
        }
        if(a[mid]<=X)
            low=mid+1;
        else
            high=mid-1;
    }
    if(flag==1)
        printf("element is found");
    else
        printf("element is not found");
}
```

```
[bldea@localhost ~]$ gcc binary.c
[bldea@localhost ~]$ ./a.out
enter the value of N
5
enter the elements
12
34
56
78
90
enter the search elements
34
element is found
```

```
[bldea@localhost ~]$ ./a.out
enter the value of N
5
enter the elements
13
45
67
89
99
enter the search elements 11
element is not found
```

```
/*Program to demonstrate the working of stack of size N using an
array. The elements of the stack may assume to be of type integer
or real, the operations to be supported are 1.PUSH 2.POP
3.DISPLAY.*/
```

```
#include<stdio.h>
#define MAXSIZE 5
int top=-1,s[20],item;

void PUSH()
{
    if(top==MAXSIZE-1)
    {
        printf("Stack overflow");
        return;
    }
    else
    top=top++;
    printf("Enter the element to be inserted into stack:");
    scanf("%d",&s[top]);
}

int POP()
{
    if(top==-1)
    {
        printf("Stack underflow");
        return;
    }
    else
        printf("The popped element is %d\n",s[top]);
        top--;
}

void display()
{
    int i=0;
        if(top==0)
        {
            printf("Stack underflow");
            return;
        }
}
```

```
else
for(i=top;i>=0;i--)
printf("%d\n",s[i]);
}

int main()
{
    int ch;
    char ok;
    do
    {
        printf("\nMENU\n1.Push\n2.Pop\n3.Display\n4.Exit\n");
        printf("Enter your choice");
        scanf("%d",&ch);

        switch(ch)
        {
            case 1: PUSH();
                    break;
            case 2: POP();
                    break;
            case 3: display();
                    break;
            default: return;
        }
        printf("Do you wish to continue(Y/N):");
        scanf("%c",&ok);
    }
    while(ok!='n');
    return;
}
```

```
[bldea@localhost ~]$ gcc stack1.c
[bldea@localhost ~]$ ./a.out
```

MENU

- 1.Push
- 2.Pop
- 3.Display
- 4.Exit

Enter your choice 1

Enter the element to be inserted into stack:11

Do you wish to continue(Y/N) :

MENU

- 1.Push
- 2.Pop
- 3.Display
- 4.Exit

Enter your choicel

Enter the element to be inserted into stack:22

Do you wish to continue(Y/N) :

MENU

- 1.Push
- 2.Pop
- 3.Display
- 4.Exit

Enter your choicel

Enter the element to be inserted into stack:33

Do you wish to continue(Y/N) :

MENU

- 1.Push
- 2.Pop
- 3.Display
- 4.Exit

Enter your choicel

Enter the element to be inserted into stack:45

Do you wish to continue(Y/N) :

MENU

- 1.Push
- 2.Pop
- 3.Display
- 4.Exit

Enter your choicel

Enter the element to be inserted into stack:56

Do you wish to continue(Y/N) :

MENU

- 1.Push
- 2.Pop
- 3.Display
- 4.Exit

Enter your choice1

Stack overflow

Do you wish to continue(Y/N) :

MENU

- 1.Push
- 2.Pop
- 3.Display
- 4.Exit

Enter your choice 2

The popped element is 56

Do you wish to continue(Y/N) :

MENU

- 1.Push
- 2.Pop
- 3.Display
- 4.Exit

Enter your choice2

The popped element is 45

Do you wish to continue(Y/N) :

MENU

- 1.Push
- 2.Pop
- 3.Display
- 4.Exit

Enter your choice2

The popped element is 33

Do you wish to continue(Y/N) :

MENU

- 1.Push
- 2.Pop
- 3.Display
- 4.Exit

Enter your choice2

The popped element is 22

Do you wish to continue(Y/N) :

MENU

- 1.Push
- 2.Pop
- 3.Display
- 4.Exit

```
Enter your choice2
The popped element is 11
Do you wish to continue(Y/N) :
MENU
1.Push
2.Pop
3.Display
4.Exit
Enter your choice2
Stack underflow
```

```
Do you wish to continue(Y/N) :
MENU
1.Push
2.Pop
3.Display
4.Exit
Enter your choice 4
```

```
//Program to convert infix expression to postfix expression

#include<stdio.h>
#include<stdlib.h>
#define size 100
char stack[size];
int top=-1;
void push(char item)
{
    if(top==size-1)
    {
        printf("stack is overflow");
    }
    else
        top++;
        stack[top]=item;
}
char pop()
{
    char item;

    item=stack[top];
    top--;
    return item;
}
int isoperator(char symbol)
{
    if(symbol=='^' || symbol=='/' || symbol=='*' || symbol=='+' || symbol=='-')
        return 1;
    else
        return 0;
}
int preceed(char symbol)
{
    if(symbol=='^')
        return 3;
    else if(symbol=='*' || symbol=='/')
        return 2;
    else if(symbol=='+' || symbol=='-')
        return 1;
    else
        return 0;
}
int main()
{
    char infix[size], postfix[size], item, temp;
    int i=0, j=0;
    printf("enter the expression");
    scanf("%s", &infix);
```

```
while(infix[i]!='\0')
{
    item=infix[i];
    if(item=='(')
    {
        push(item);
    }

else if(item>='A'&&item<='Z' || item>='a'&&item<='z')
    {
        postfix[j]=item;
        j++;
    }
else if(isoperator(item)==1)
    {
        temp=pop();
        while(isoperator(temp)==1&&preced(temp)>=preced(item))
        {
            postfix[j]=temp;
            j++;
            temp=pop();
        }
        push(temp);
        push(item);
    }

else if(item==')')
    {
        temp=pop();
        while(temp!='(')
        {
            postfix[j]=temp;
            j++;
            temp=pop();
        }
    }
else
    {
        printf("invalid expression");
        exit(0);
    }
    i++;
}
while(top>-1)
{
    postfix[j]=pop();
    j++;
}
postfix[j]='\0';
printf("the output is");
puts(postfix); }
```

```
[bldea@localhost ~]$ gcc infixtopostfix.c
[bldea@localhost ~]$ ./a.out
enter the expression a+b*c/d
the output is abc*d/+
```

```
[bldea@localhost ~]$ ./a.out
enter the expression x/y*z+m*n
the output is xy/z*mn*+
```

```
/* Program to simulate the working of an ordinary Queue using an array*/
```

```
#include<stdio.h>
```

```
#define MAX 5
```

```
void Qinsert();
```

```
void Qdelete();
```

```
void Qdisplay();
```

```
int q[MAX],front=-1,rear=-1,ch;
```

```
int main()
```

```
{
```

```
while(1)
```

```
{
```

```
printf("\n 1.insert\n 2.Delete \n 3.Display \n 4.Exit \n");
```

```
printf("Enter your choice:-\n");
```

```
scanf("%d",&ch);
```

```
switch(ch)
```

```
{
```

```
case 1:Qinsert();break;
```

```
case 2:Qdelete();break;
```

```
case 3:Qdisplay();break;
```

```
default:return;
```

```
}
```

```
}
```

```
}
```

```
void Qinsert()
```

```
{
```

```
int n;
```

```
if(rear==MAX-1)
```

```
{
```

```
printf("\n Queue is overflow \n");
```

```
return;
```

```
}
```

```
else
```

```
{
```

```
printf("Enter the element:\n");
```

```
scanf("%d",&n);
```

```
rear++;
```

```
q[rear]=n;
if(front==-1)
front++;
}
return;
}

void Qdelete()
{
int n;
if(front>rear)
{
printf("\n Queue is underflow\n");
return;
}
else
n=q[front];
printf("\n deleted elements is=%d",n);
front++;
return;
}

void Qdisplay()
{
int i;
if(front>rear)
printf("\n Queue is underflow\n");
else
{
printf("\n queue elements are:\n");
for(i=front;i<=rear;i++)
printf("%4d",q[i]);
}
}
```

```
[bldea@localhost ~]$ gcc que.c
[bldea@localhost ~]$ ./a.out
```

```
1.insert
2.Delete
3.Display
4.Exit
Enter your choice:-1
Enter the element:
12
```

```
1.insert
2.Delete
3.Display
4.Exit
Enter your choice:-1
Enter the element:
23
```

```
1.insert
2.Delete
3.Display
4.Exit
Enter your choice:-1
Enter the element:
34
```

```
1.insert
2.Delete
3.Display
4.Exit
Enter your choice:-1
Enter the element:
45
```

```
1.insert
2.Delete
3.Display
4.Exit
Enter your choice:-1
Enter the element:
56
```

```
1.insert
2.Delete
3.Display
4.Exit
```

Enter your choice:-1

Queue is overflow

- 1.insert
- 2.Delete
- 3.Display
- 4.Exit

Enter your choice:-3

queue elements are:
12 23 34 45 56

- 1.insert
- 2.Delete
- 3.Display
- 4.Exit

Enter your choice:-2

deleted elements is=12

- 1.insert
- 2.Delete
- 3.Display
- 4.Exit

Enter your choice:-2

deleted elements is=23

- 1.insert
- 2.Delete
- 3.Display
- 4.Exit

Enter your choice:-2

deleted elements is=34

- 1.insert
- 2.Delete
- 3.Display
- 4.Exit

Enter your choice:-2

deleted elements is=45

- 1.insert
- 2.Delete
- 3.Display
- 4.Exit

Enter your choice:-

2

deleted elements is=56

- 1.insert
- 2.Delete
- 3.Display
- 4.Exit

Enter your choice:-

2

Queue is underflow

```
//Program to construct a singly linked list
```

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

struct node
{
    int rno;
    char name[10];
    struct node *p;
};
typedef struct node *Node;
Node insert(Node head,int rno,char *nm);
Node del(Node head,int rn);
void search(Node head,int rno);
void display(Node head);

Node getnode()
{
    Node tmp;
    tmp=(Node)malloc(sizeof(struct node));
    tmp->p=NULL;
    return tmp;
}

Node insert(Node head,int rno,char*nm)
{
    Node tmp;
    tmp=getnode();
    tmp->rno=rno;
    strcpy(tmp->name,nm);
    tmp->p=head;
    head=tmp;
    return head;
}

Node del(Node head,int rn)
{
    Node tmp=head,q;
    if(head==NULL)
    {
        printf("List is Empty\n");
        return NULL;
    }
}
```

```
while (tmp!=NULL&&tmp->rno!=rn)
{
    q=tmp;
    tmp=tmp->p;
}
if (tmp==NULL)
{
    printf("\n Roll Number is Not Found");
    return head;
}
if (q==NULL)
    head=head->p;
else
    free(tmp);
return head;
}

void search(Node head,int rn)
{
    Node tmp=head;
    if (tmp==NULL)
    {
        printf("list is Empty\n");
        return;
    }
    while (tmp!=NULL&&tmp->rno!=rn)
    {
        tmp=tmp->p;
    }
    if (tmp==NULL)
    {
        printf("Not Found\n");
        return;
    }
    else
        printf("\nFound\n%d\t%s\n", tmp->rno, tmp->name);
}

void display(Node head)
{
    Node tmp=head;
    if (tmp==NULL)
    {
        printf("List is Empty\n");
        return;
    }
    printf("\nStudent Information:\n");
    printf("\n_____ \n");

    while (tmp!=NULL)
    {
        printf("\n%d", tmp->rno);
```

```
printf("%s",tmp->name);
tmp=tmp->p;
}
printf("\n_____ \n");
}

void main()
{
    Node head=NULL;
    int ch,rno;
    char name[100];

while(1)
{
    printf("\n1.Insert\n 2.Delete\n 3.search\n 4.Display\n 5.Exit\n");
    printf("Enter your choice:\n");
    scanf("%d",&ch);

    if(ch==1)
    {
        printf("Enter The Roll Number:\n");
        scanf("%d",&rno);
        printf("Enter The Name:\n");
        scanf("%s",name);
        head=insert(head,rno,name);
    }
    else
        if(ch==2)
        {
            printf("Enter Roll No:\n");
            scanf("%d",&rno);
            head=del(head,rno);
        }
        else
            if(ch==3)
            {
                printf("Enter Roll No:\n");
                scanf("%d",&rno);
                search(head,rno);
            }
            else
                if(ch==4)
                    display(head);
                if(ch==5)
                    exit(0);
            }
        }
}
```

```
[bldea@localhost ~]$ gcc link1.c
[bldea@localhost ~]$ ./a.out
```

```
1.Insert
2.Delete
3.search
4.Display
5.Exit
```

Enter your choice:

1

Enter The Roll Number:

1

Enter The Name:

ABC

```
1.Insert
2.Delete
3.search
4.Display
5.Exit
```

Enter your choice:

1

Enter The Roll Number:

2

Enter The Name:

neha

```
1.Insert
2.Delete
3.search
4.Display
5.Exit
```

Enter your choice:

1

Enter The Roll Number:

3

Enter The Name:

amit

```
1.Insert
2.Delete
3.search
4.Display
5.Exit
```

Enter your choice:

1

Enter The Roll Number:

4

Enter The Name:

sunita

- 1.Insert
- 2.Delete
- 3.search
- 4.Display
- 5.Exit

Enter your choice:

1

Enter The Roll Number:

5

Enter The Name:

snehal

- 1.Insert
- 2.Delete
- 3.search
- 4.Display
- 5.Exit

Enter your choice:

4

Student Information:

5snehal

4sunita

3amit

2neha

1ABC

- 1.Insert
- 2.Delete
- 3.search
- 4.Display
- 5.Exit

Enter your choice:

3

Enter Roll No:

3

Found

3 amit

- 1.Insert
- 2.Delete
- 3.search
- 4.Display
- 5.Exit

Enter your choice:

2

Enter Roll No:

4

Java Lab Manual

**BLDEA's
S.B.ARTS AND K.C.P. SCIENCE COLLEGE, VIJAYAPUR
DEPARTMENT OF COMPUTER SCIENCE**

Lab Manual

Course Code: 21BSC3C3CS1P

Course Title: JAVA LAB

Credits 02

Formative Assessment Marks: 25

Summative Assessment Marks: 25

BSC-III Semester

2022-23

Practice Labs:

1. Program to print the following triangle of numbers

1
12
123
1234
12345

```
class pyramid
{
    public static void main(String[] args)
    {
        int rows = 5;

        for (int i = 1; i <= rows; ++i)
        {
            for (int j = 1; j <= i; ++j)
            {
                System.out.print(j + " ");
            }
            System.out.println();
        }
    }
}
```

2. Program to simple java application, to print the message,“Welcome to java”

```
class Welcome
{
    public static void main(String as[])
    {
        System.out.println("Welcome to Java");
    }
}
```

3. Program to find the area of rectangle.

```
class RectArea1
{
public static void main(String as[])
{
    int L=3,B=5;
    int area=L*B;

    System.out.println("Area of Rectangle = "+area);
}
}
```

4. Program to create a user defined exception say Pay Out of Bounds.

```
public class exp2
{
    public static void main (String args[])
    {
        int array[] = {20,20,40};
        try
        {
            // assign i value >=3 to check out of bounds exception
            for(int i = 2; i >=0; i--)
            {
                System.out.println("The value of array is" +array[i]);
            }
        }
        catch (ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Error. Array is out of Bounds" +e);
        }
    }
}
```

```
}}
```

Part A: Programming Lab – Java Fundamentals – OOPS in JAVA

- 1 Program to assign two integer values to X and Y. Using the “if statement” the output of the program should display a message whether X is greater than Y.**

```
import java.io.*;

class Big
{
    public static void main (String[] args)
    {

        int X = 50, Y = 20;

        if ( X == Y )
            System.out.println ("both are equal");
        else if (X > Y )
            System.out.println (X + " is greater");
        else
            System.out.println (Y + " is greater");
    }
}
```

Java Lab Manual

Output 1 :

```
bsc-cs@bsccs-OptiPlex-3070:~$ javac Big.java  
bsc-cs@bsccs-OptiPlex-3070:~$ java Big
```

50 is greater

Output 2 :

```
bsc-cs@bsccs-OptiPlex-3070:~$ javac Big.java  
bsc-cs@bsccs-OptiPlex-3070:~$ java Big
```

both are equal

- 2 Program to list the factorial of the numbers 1 to 10. To calculate the factorial value, use while loop. (Hint: Fact of 4 = 4*3*2*1)**

```
import java.io.*;

class Factorial
{
    public static void main(String[] args)
    {
        int number = 10;
        long fact = 1;
        int i = 1;
        while(i<=number)
        {
            fact = fact * i;
            System.out.println("Factorial of "+number+" is: "+fact);
            i++;
        }
    }
}
```

Java Lab Manual

Output :

```
bsc-cs@bsccs-OptiPlex-3070:~$ javac Factorial.java
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ java Factorial
```

Factorial of 1 is: 1

Factorial of 2 is: 2

Factorial of 3 is: 6

Factorial of 4 is: 24

Factorial of 5 is: 120

Factorial of 6 is: 720

Factorial of 7 is: 5040

Factorial of 8 is: 40320

Factorial of 9 is: 362880

Factorial of 10 is: 3628800

3 Program to find the area and circumference of the circle by accepting the radius from the user.

```
import java.io.*;
import java.util.Scanner;

public class Circle
{
    public static void main(String[] args)
    {
        double radius, area, circumference;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter Radius of Circle:");
        radius = in.nextDouble();

        // Calculate area and circumference of circle
        area = 3.142 * radius * radius;
        circumference = 2 * 3.142 * radius;

        System.out.println("Area of Circle : " + area);
        System.out.println("Circumference of Circle : " + circumference);
    }
}
```

Java Lab Manual

Output :

```
bsc-cs@bsccs-OptiPlex-3070:~$ javac Circle.java
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ java Circle
```

```
Enter Radius of Circle: 2.5
```

```
Area of Circle : 19.6375
```

```
Circumference of Circle : 15.71
```

- 4 Program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use function overloading.

```
public class OverloadingTest1
{
    //addition of integers
    void addition(int a, int b)
    {
        int sum=a+b;
        System.out.println("Sum of two integer numbers is "+sum);
    }

    //addition of float numbers

    void addition(float a, float b)
    {
        float sum=a+b;
        System.out.println("Sum of two floating point numbers is "+sum);
    }

    public static void main(String args[])
    {
        OverloadingTest1 d = new OverloadingTest1();
        d.addition(5,10);
        d.addition(1.5,8.5);
    }
}
```

Java Lab Manual

Output :

```
bsc-cs@bsccs-OptiPlex-3070:~$ javac OverloadingTest1.java  
bsc-cs@bsccs-OptiPlex-3070:~$ java OverloadingTest1
```

Sum of two integer numbers is 15

Sum of two floating point numbers is 10

Java Lab Manual

- 5 Write a program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the super class. MulDiv should have methods to multiply and divide. A main function should access the methods and perform the mathematical operations.

```
class Addsub
{
    int num1,num2;

    Addsub(int n1, int n2)
    {
        num1 = n1;
        num2 = n2;
    }

    int add()
    {
        return num1+num2;
    }
    int sub()
    {
        return num1-num2;
    }
}
class Multdiv extends Addsub
{
    public Multdiv(int n1, int n2)
    {
        super(n1, n2);
    }
    int mul()
    {
        return num1*num2;
    }
    float div()
    {
        return num2/num1;
    }
}
```

Java Lab Manual

```
public class MathematicalOP
{
    public static void main(String arg[])
    {
        Addsub r1=new Addsub(50,20);
        int ad = r1.add();
        int sb = r1.sub();
        System.out.println("Addition =" +ad);
        System.out.println("Subtraction =" +sb);
        Multdiv r2 =new Multdiv(4,20);
        int ml = r2.mul();
        float dv =r2.div();
        System.out.println("Multiply =" +ml);
        System.out.println("Division =" +dv);
    }
}
```

Java Lab Manual

Output :

```
bsc-cs@bsccs-OptiPlex-3070:~$ javac MathematicalOP.java
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ java MathematicalOP
```

```
    Addition =70
```

```
    Subtraction =30
```

```
    Multiply =80
```

```
    Division =0.2
```

6. Program with class variable that is available for all instances of a class. Use static variable declaration. Observe the changes that occur in the object's member variable values.

```
class Student
{
int rollNo;
String name;
static String collegeName;           //static variable
}

class StaticVar
{
public static void main(String[] args)
{
//create 3 object which will share collegeName value
Student s1= new Student();
Student s2= new Student();
Student s3= new Student();

//assign value to static variable collegeName

Student.collegeName="Topper Skills";

//assign values to instance variables
s1.rollNo=1;
s1.name="stud1";

s2.rollNo=2;
s2.name="stud2";

s3.rollNo=3;
s3.name="stud3";

//Print the values of the objects
System.out.println("S1 Roll No.= "+s1.rollNo+" S1 Name=" +s1.name+" S1 College Name="
"+s1.collegeName );
System.out.println("S2 Roll No.= "+s2.rollNo+" S2 Name= "+s2.name+" S2 College Name="
"+s2.collegeName );
System.out.println("S3 Roll No.= "+s3.rollNo+" S3 Name= "+s3.name+" S3 College Name="
"+s3.collegeName );
```

Java Lab Manual

```
//if one object change the value of static variable then it will reflect into all objects
```

```
s2.collegeName="my college";
```

```
s2.name="Devavrat";
```

```
//Print the values of the objects after change
```

```
System.out.println("S1 Roll No.= "+s1.rollNo+" S1 Name= "+s1.name+" S1 College Name= "+s1.collegeName );
```

```
System.out.println("S2 Roll No.= "+s2.rollNo+" S2 Name= "+s2.name+" S2 College Name= "+s2.collegeName );
```

```
System.out.println("S3 Roll No.= "+s3.rollNo+" S3 Name= "+s3.name+" S3 College Name= "+s3.collegeName );
```

```
}
```

```
}
```

Java Lab Manual

Output :

```
bsc-cs@bsccs-OptiPlex-3070:~$ javac StaticVar.java
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ java StaticVar
```

```
S1 Roll No.= 1  S1 Name= stud1  S1 College Name= Topper Skills
```

```
S2 Roll No.= 2  S1 Name= stud2  S2 College Name= Topper Skills
```

```
S2 Roll No.= 3  S1 Name= stud3  S3 College Name= Topper Skills
```

```
S1 Roll No.= 1  S1 Name= stud1  S1 College Name= my college
```

```
S2 Roll No.= 2  S1 Name= stud2  S2 College Name= my college
```

```
S2 Roll No.= 3  S1 Name= stud3  S3 College Name= my college
```

7. Program to create a student class with following attributes; EnrollmentNo: Name, Markofsub1 , Markofsub2, markofsub3, TotalMarks. Total of the three marks must be calculated only when the student passes in all three subjects .The passing mark for each subject is 50. If a candidate fails in any one of the subjects his total mark must be declared as zero. Using this condition write a constructor for this class. Write separate functions for accepting and displaying student details. In the main method create an array of three student objects and display the details.*

```
import java.util.Scanner;
class Student
{
    int enroll;
    String name=new String();
    int sub1,sub2,sub3;
    int total;

    void accept()
    {
        Scanner ip=new Scanner(System.in);

        System.out.println("Enter the Name");
        name=ip.nextLine();

        System.out.println("Enter the Enrollment Number");
        enroll=ip.nextInt();

        System.out.println("Enter three Subject Marks");
        sub1=ip.nextInt();
        sub2=ip.nextInt();
        sub3=ip.nextInt();
    }

    void display()
    {
        System.out.println("Enrollment No: "+enroll);
        System.out.println("Name: "+name);
        System.out.println("Sub1 Marks: "+sub1);
        System.out.println("Sub2 Marks: "+sub2);
        System.out.println("Sub3 Marks: "+sub3);

        if(sub1<50 || sub2<50 || sub3<50)
```

Java Lab Manual

```
total=0;
else
total=sub1+sub2+sub3;
System.out.println("Total Marks: "+total);
}
public static void main(String args[])
{
Student s[]=new Student[3];

for(int i=0;i<3;i++)
{
System.out.println("Student Details of "+(i+1));
s[i]=new Student();
s[i].accept();
s[i].display();
}
}
}
```

output :

```
bsc-cs@bsccs-OptiPlex-3070:~$ javac StudentNew.java
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ java Student
```

Student Details of 1

Enter the Name

anita

Enter the Enrollment Number

123

Enter three Subject Marks

78

89

77

Enrollment No: 123

Name: anita

Sub1 Marks: 78

Sub2 Marks: 89

Sub3 Marks: 77

Total Marks: 244

Student Details of 2

Enter the Name

sunil

Enter the Enrollment Number

121

Enter three Subject Marks

67

89

17

Enrollment No: 121

Name: sunil

Sub1 Marks: 67

Sub2 Marks: 89

Sub3 Marks: 17

Total Marks: 0

Student Details of 3

Enter the Name

neha

Enter the Enrollment Number

150

Enter three Subject Marks

98

67

88

Enrollment No: 150

Name: neha

Sub1 Marks: 98

Sub2 Marks: 67

Sub3 Marks: 88

Total Marks: 253

8. Illustrate creation of thread by

a. extending Thread Class

```
class A extends Thread
{
    public void run()
    {
        for(int i=1; i<=5; i++)
        {
            System.out.println("\t From Thread A : i = "+i);
        }
        System.out.println("exit from A ");
    }
}

class B extends Thread
{
    public void run()
    {
        for(int j=1; j<=5; j++)
        {
            System.out.println("\t From Thread B : j = "+j);
            if(j==3)
                stop();
        }
        System.out.println("exit from B ");
    }
}

class C extends Thread
{
    public void run()
    {
        for(int k=1; k<=5; k++)
        {
            System.out.println("\t From Thread C : k = "+k);
            if(k==1)
            try
            {
                sleep(1000);
            }
            catch(Exception e)
            {
            }
        }
    }
}
```

Java Lab Manual

```
    System.out.println("exit from C ");  
    } }
```

```
class Threadmethods
```

```
{  
    public static void main(String args[])  
    {  
        A threadA = new A();  
        B threadB = new B();  
        C threadC = new C();  
        System.out.println("Start thread A");  
        threadA.start();  
        System.out.println("Start thread B");  
        threadB.start();  
        System.out.println("Start thread C");  
        threadC.start();  
        System.out.println("End of main thread");  
    }  
}
```

```
C:\Java50\bin>javac Threadmethods.java
```

```
C:\Java50\bin>java Threadmethods
```

```
Start thread A
```

```
Start thread B
```

```
Start thread C
```

```
    From Thread A : i = 1
```

```
End of main thread
```

```
    From Thread A : i = 2
```

```
    From Thread B : j = 1
```

```
    From Thread C : k = 1
```

```
    From Thread A : i = 3
```

```
    From Thread B : j = 2
```

```
    From Thread A : i = 4
```

```
    From Thread B : j = 3
```

```
    From Thread A : i = 5
```

```
exit from A
```

```
    From Thread C : k = 2
```

```
    From Thread C : k = 3
```

```
    From Thread C : k = 4
```

```
    From Thread C : k = 5
```

```
exit from C
```

b. Implementing Runnable Interface

class A implements Runnable

```
{  
    public void run()  
    {  
        for(int i=1; i<=5; i++)  
        {  
            System.out.println("\t From Thread A : i = "+i);  
        }  
        System.out.println("exit from A ");  
    }  
}
```

class B implements Runnable

```
{  
    public void run()  
    {  
        for(int j=1; j<=5; j++)  
        {  
            System.out.println("\t From Thread B : j = "+j);  
        }  
        System.out.println("exit from B ");  
    }  
}
```

class C implements Runnable

```
{
    public void run()
    {
        for(int k=1; k<=5; k++)
        {
            System.out.println("\t From Thread C : k = "+k);
        }
        System.out.println("exit from C ");
    }
}
```

class RunnableMain

```
{
    public static void main(String args[])
    {
        A threadA = new A();
        Thread ta=new Thread(threadA);
        B threadB = new B();
        Thread tb=new Thread(threadB);
        C threadC = new C();
        Thread tc=new Thread(threadC);
        ta.start();
        tb.start();
        tc.start();
    }
}
```

Java Lab Manual

```
bsc-cs@bsccs-OptiPlex-3070:~$ javac RunnableMain.java
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ java RunnableMain
```

```
    From Thread A : i = 1
```

```
    From Thread B : j = 1
```

```
    From Thread C : k = 1
```

```
    From Thread B : j = 2
```

```
    From Thread A : i = 2
```

```
    From Thread B : j = 3
```

```
    From Thread B : j = 4
```

```
    From Thread C : k = 2
```

```
    From Thread B : j = 5
```

```
    From Thread A : i = 3
```

```
exit from B
```

```
    From Thread C : k = 3
```

```
    From Thread A : i = 4
```

```
    From Thread C : k = 4
```

```
    From Thread A : i = 5
```

```
    From Thread C : k = 5
```

```
exit from A
```

```
exit from C
```

9. Write a program to demonstrate multiple inheritance and use of Implementing Interfaces

```
import java.util.Scanner;

interface Department
{
    void getattendance();
}

class Student
{
    int sno;
    String sname, sclass;

    void getvalue(int s,String n,String c)
    {
        sno=s;
        sname=n;
        sclass=c;
    }
    void display()
    {
        System.out.println("Student Information");
        System.out.println("sno :"+sno);
        System.out.println("name :"+sname);
        System.out.println("class :"+sclas);
    }
}

class Exam extends Student implements Department
{
    int peratt, att;
    Scanner sc=new Scanner(System.in);

    public void getattendance()
    {
        System.out.println("Enter no. of classes attended");
        att=sc.nextInt();
    }
}
```

```
float calattendance()
{
    peratt=(att*100)/50;

    return 0;
}
boolean eligible()
{
    display();
    System.out.println("Student Attendance in Percentage:"+peratt);
    System.out.println("Whether student is eligible for Exam?");
    if(peratt>=75)
        return true;
    else
        return false;
}
}
class StudentAttendance
{
    public static void main(String args[])
    {
        boolean c;
        Exam E=new Exam();
        E.getvalue(12,"ABC","BSCV");
        E.getattendance();
        E.calattendance();
        c=E.eligible();
        System.out.println(c);
    }
}
```

Output :

```
C:\Java50\bin>javac StudentAttendance1.java
```

```
C:\Java50\bin>java StudentAttendance
```

```
Enter no. of classes attended
```

```
35
```

```
Student Information
```

```
sno :12
```

```
name :ABC
```

```
class :BSCV
```

```
Student Attendance in Percentage:58
```

```
Whether student is eligible for Exam?
```

```
false
```

Part – B

1. Program to catch Negative Array Size Exception. This exception is caused when the array size is initialized to negative values.
2. Program to demonstrate exception handling with try, catch and finally.
3. Program which create and displays a message on the window
4. Program to draw several shapes in the created window
5. Program to create a 4x4 grid and fills it in with 15 buttons, each labeled with its index.
6. Program which creates a frame with two buttons father and mother. When we click the father button the name of the father, his age and designation must appear. When we click mother button similar details of mother also appear.
7. Create a frame which displays your personal details with respect to a button click
8. Program to create a window with Text Fields and Buttons. The "ADD" button adds the two integers and display the result. The "CLEAR" button shall clear all the text fields.
9. Program to create a window, when we press M or m, the window displays "good morning", Aora, the window displays " Good Afternoon", Eore, the window displays "good morning", Norn, the window displays "good morning"
10. Demonstrate the various mouse handling events using suitable example

/*1. Program to catch NegativeArraySize Exception.This exception is caused when the array size is initialized to negative values.*/

```
import java.util.Scanner;

class NegativeSize
{
    public static void main(String as[])
    {
        try
        {
            Scanner ip=new Scanner(System.in);

            System.out.println("Enter the Size of the Array");

            int size,i;

            size=ip.nextInt();

            int a[]=new int[size];

            System.out.println("Enter the elements of the array");

            for(i=0;i<size;i++)

                a[i]=ip.nextInt();

            System.out.println("Enter the elements of the array");

            for(i=0;i<size;i++)

                System.out.println(a[i]);

        }

        catch(NegativeArraySizeException ne)

        {
```

Java Lab Manual

```
        System.out.println("Caught "+ne);  
    }  
}
```

Output :

```
bsc-cs@bsccs-OptiPlex-3070:~$ javac NegativeSize.java
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ java NegativeSize
```

Enter the Size of the Array

5

Enter the elements of the array

23

12

45

67

89

Enter the elements of the array

23

12

45

67

89

```
bsc-cs@bsccs-OptiPlex-3070:~$ java NegativeSize
```

Enter the Size of the Array

-5

Caught java.lang.NegativeArraySizeException: -5

//2. Program to demonstrate exception handling with try,catch and finally.

```
import java.lang.Exception;
import java.util.Scanner;
class MyException extends Exception
{
    MyException(String message)
    {
        super(message);
    }
}
class TestMyException
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int x,y;
        System.out.println("Enter value of x and y");
        x=sc.nextInt();
        y=sc.nextInt();

        try
        {
            float z=(float)x/(float)y;
            if(z<0.01)
            {
                throw new MyException("Number is too small");
            }
        }
        catch(MyException e)
        {
            System.out.println("Caught my exception");
            System.out.println(e.getMessage());
        }
        finally
```

Java Lab Manual

```
{  
    System.out.println("finally block - I am always here");  
}  
}
```

Output :

```
C:\Java50\bin>java TestMyException
```

```
Enter value of x and y
```

```
50 100
```

```
finally block - I am always here
```

```
C:\Java50\bin>java TestMyException
```

```
Enter value of x and y
```

```
5 1000
```

```
Caught my exception
```

```
Number is too small
```

```
finally block - I am always here
```

3. Program which create and displays a message on the window

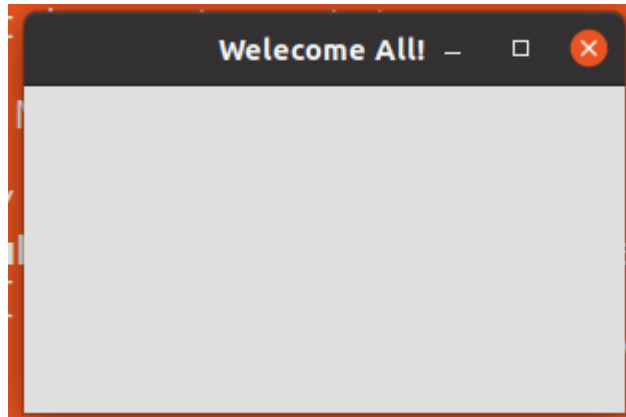
```
import java.awt.*;
import java.awt.event.*;
class Message {
    // The method we wish to call on exit.
    public static void showMessage()
    {
        System.out.println("Bye Bye!");
    }
    public static void main(String[] args)
    {
        Frame f = new Frame("Welecome All!");
        f.addWindowListener( new WindowAdapter()
        {
            public void windowClosing(WindowEvent we)
            {
                showMessage();
                System.exit(0);
            }
        } );
        f.setSize(300,200);
        f.setLocationByPlatform(true);
        f.setVisible(true);
    } }
```

Java Lab Manual

output :

```
bsc-cs@bsccs-OptiPlex-3070:~$ javac Message.java
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ java Message
```



4. Program to draw several shapes in the created window

//Program to draw several shapes in the created window

```
import java.awt.*;

import javax.swing.JFrame;

public class DisplayGraphics extends Canvas
{
    public void paint(Graphics g)
    {
        g.drawString("Hello",40,40);
        setBackground(Color.WHITE);
        g.fillRect(130, 30,100, 80);
        g.drawOval(30,130,50, 60);
        setForeground(Color.RED);
        g.fillOval(130,130,50, 60);
        g.fillRoundRect(175,150,70,50,25,50);
        g.drawArc(30, 200, 40,50,90,60);
        g.fillArc(30, 130, 40,50,180,40);

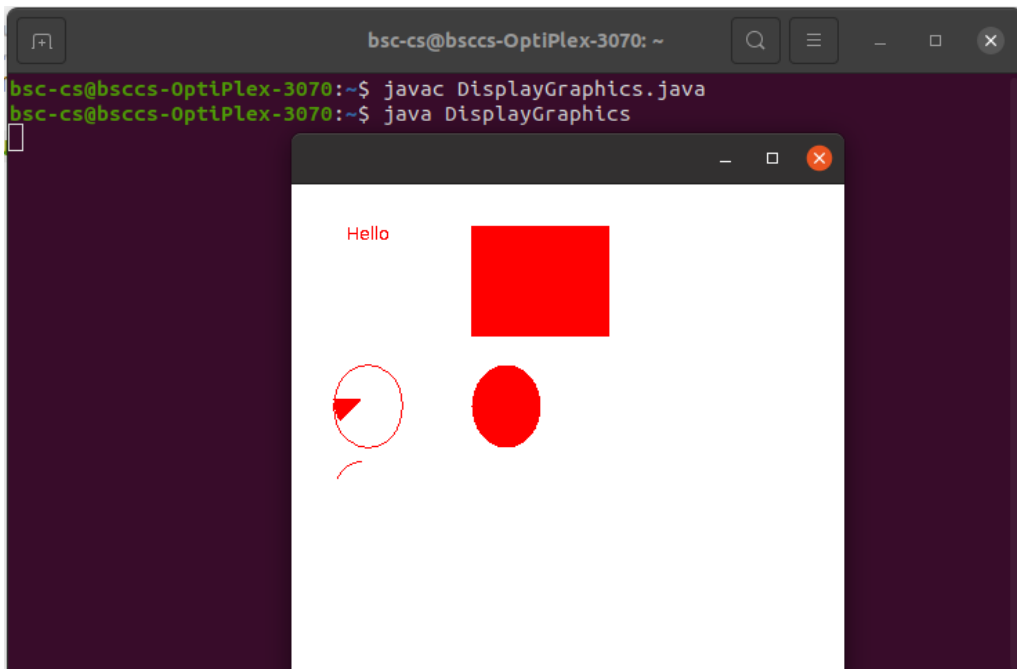
    }
    public static void main(String[] args)
    {
        DisplayGraphics m=new DisplayGraphics();
        JFrame f=new JFrame();
        f.add(m);
        f.setSize(400,400);
        //f.setLayout(null);
        f.setVisible(true);
    }
}
```

Java Lab Manual

Output :

```
bsc-cs@bsccs-OptiPlex-3070:~$ javac DisplayGraphics.java
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ java DisplayGraphics
```



/*5. Program to create a 4×4 grid and fills it in with 15 buttons, each labeled with its index.*/

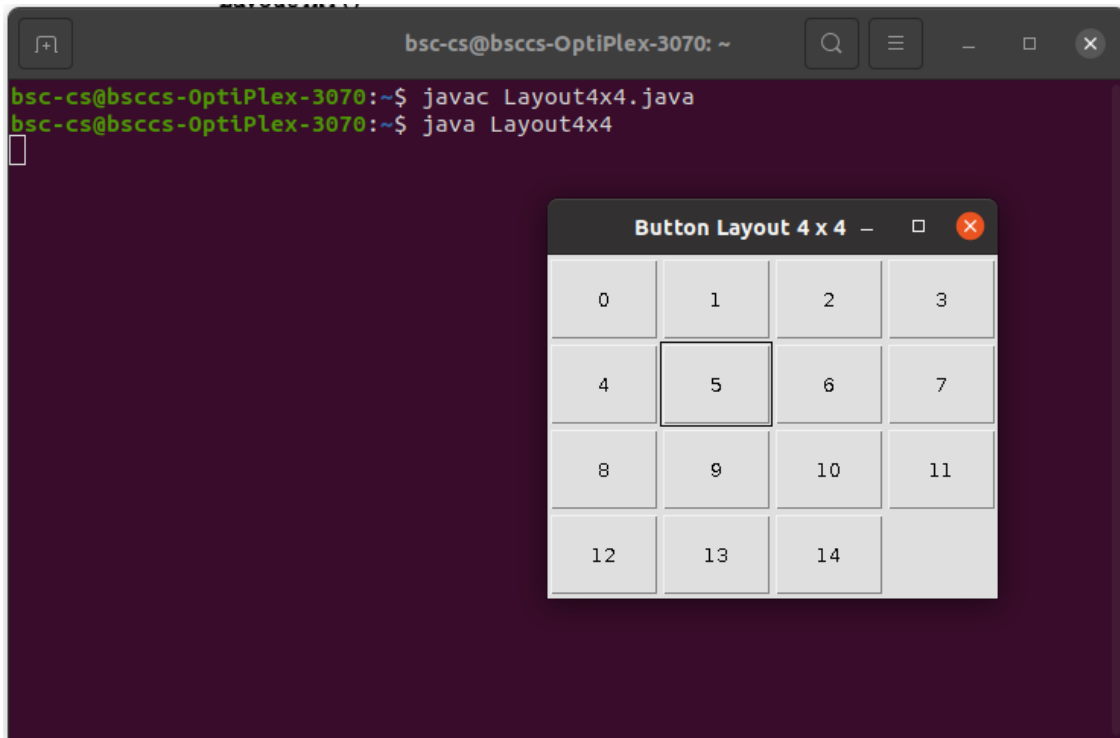
```
import java.awt.*;
import java.awt.event.*;
public class Layout4x4 extends Frame
{
    Layout4x4()
    {
        GridLayout gl=new GridLayout(4,4);
        setLayout(gl);
        setSize(200,200);
        setTitle("Button Layout 4 x 4");
        setVisible(true);
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });
        Button b[]=new Button[15];
        for(int i=0;i<15;i++)
        {
            String s=String.valueOf(i);
            b[i]=new Button(s);
            add(b[i]);
        }
    }
    public static void main(String as[])
    {
        Layout4x4 f=new Layout4x4();
    }
}
```

Java Lab Manual

Output :

```
bsc-cs@bsccs-OptiPlex-3070:~$ javac Layout4x4.java
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ java Layout4x4
```



6. Program which creates a frame with two buttons father and mother. When we click the father button the name of the father, his age and designation must appear. When we click mother button similar details of mother also appear.

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
public class FMdetails extends Frame implements ActionListener
```

```
{
```

```
    Label n1,n2,n3;
```

```
    Button b1,b2;
```

```
    FMdetails()
```

```
    {
```

```
        setLayout(new GridLayout(2,3));
```

```
        setSize(500,200);
```

```
        setTitle("Details");
```

```
        setVisible(true);
```

```
        addWindowListener(new WindowAdapter()
```

```
        {
```

```
            public void windowClosing(WindowEvent e)
```

```
            {
```

```
                System.exit(0);
```

```
            }
```

```
        });
```

```
        n1=new Label();
```

```
        n2=new Label();
```

```
        n3=new Label();
```

```
        b1=new Button("Mother");
```

```
        b2=new Button("Father");
```

```
        add(n1);
```

```
        add(n2);
```

```
        add(n3);
```

```
        add(b1);
```

```
        add(b2);
```

```
        b1.addActionListener(this);
```

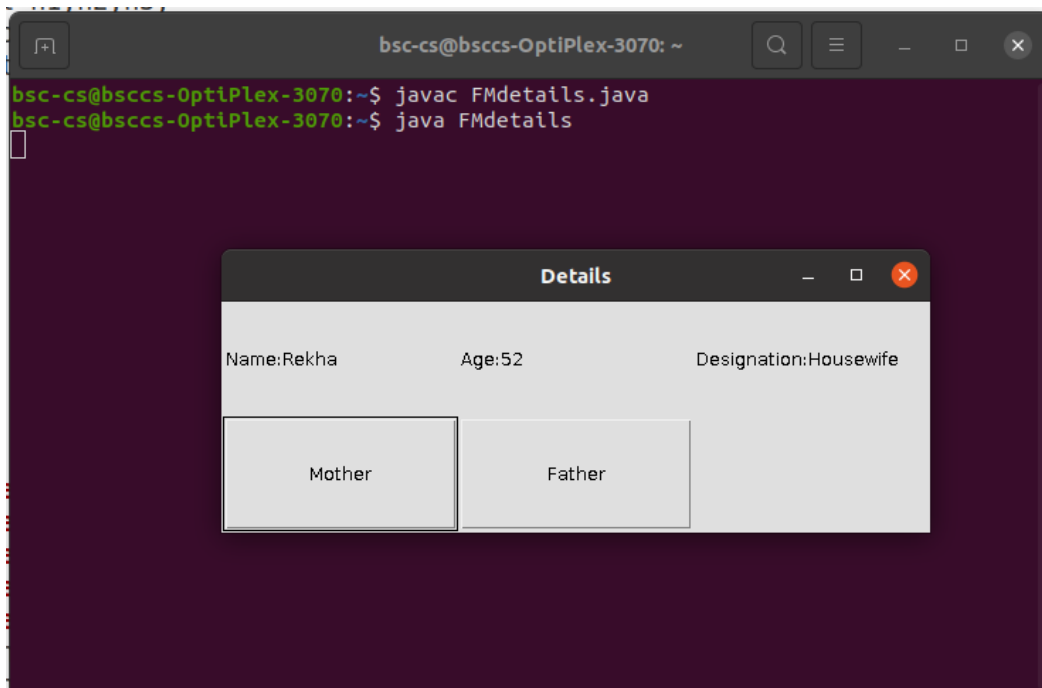
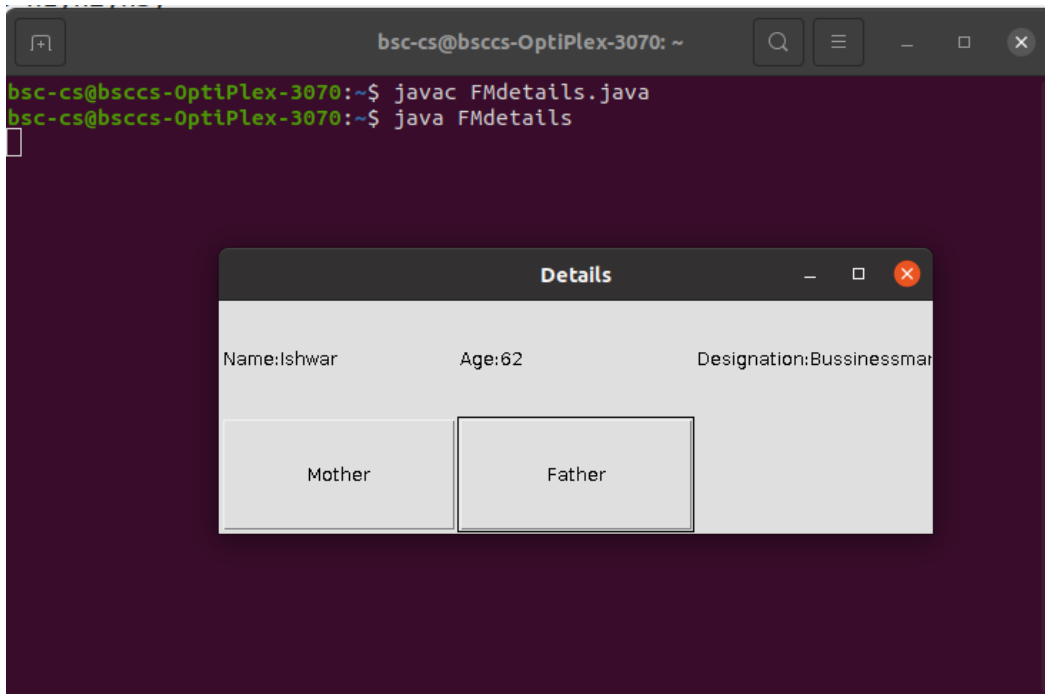
```
        b2.addActionListener(this);
```

```
    }
```

```
public void actionPerformed(ActionEvent e)
{
    if(e.getSource().equals(b1))
    {
        n1.setText("Name:Rekha");
        n2.setText("Age:52");
        n3.setText("Designation:Housewife");
    }
    if(e.getSource().equals(b2))
    {
        n1.setText("Name:Ishwar");
        n2.setText("Age:62");
        n3.setText("Designation:Bussinessman");
    }
}
public static void main(String as[])
{
    FMdetails f=new FMdetails();
}
}
```

Java Lab Manual

Output :



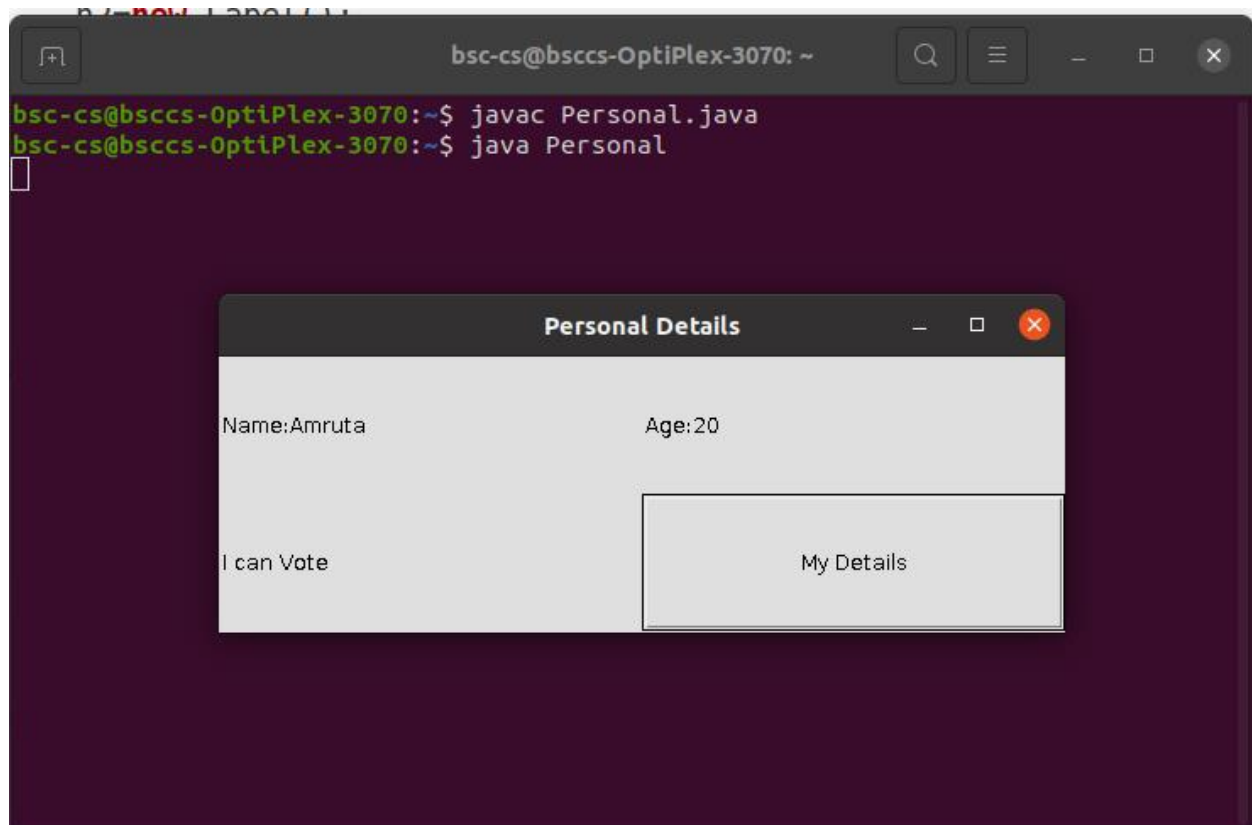
/*7. Create a frame which displays your personal details with respect to a button click*/

```
import java.awt.*;
import java.awt.event.*;
public class Personal extends Frame implements ActionListener
{
    Label n1,n2,n3;
    Button b;
    Personal()
    {
        setLayout(new GridLayout(2,3));
        setSize(500,200);
        setTitle("Personal Details");
        setVisible(true);
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });
        n1=new Label();
        n2=new Label();
        n3=new Label();
        b=new Button("My Details");
        add(n1);
        add(n2);
        add(n3);
        add(b);
        b.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource().equals(b))
        {
            n1.setText("Name:Amruta");
            n2.setText("Age:20");
            n3.setText("I can Vote");
        }
    }
    public static void main(String as[])
```

Java Lab Manual

```
{  
    Personal p=new Personal();  
}
```

output :



/*8. Program to create a window with TextFields and Buttons. The "ADD" button adds the two integers and display the result. The "CLEAR" button shall clear all the text fields.*/

```
import java.awt.*;
import java.awt.event.*;
public class CalSum extends Frame implements ActionListener
{
    Label n1,n2,n3;
    TextField t1,t2,t3;
    Button b1,b2;
    CalSum()
    {
        GridLayout gl=new GridLayout(4,2);
        setLayout(gl);
        setSize(200,300);
        setTitle("Calculate Sum");
        setVisible(true);
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });
        n1=new Label("Enter n1");
        n2=new Label("Enter n2");
        n3=new Label("Result");
        t1=new TextField();
        t2=new TextField();
        t3=new TextField();
        b1=new Button("ADD");
        b2=new Button("CLEAR");
        add(n1);
        add(t1);
        add(n2);
        add(t2);
        add(n3);
        add(t3);
        add(b1);
        add(b2);
        b1.addActionListener(this);
        b2.addActionListener(this);
    }
}
```

Java Lab Manual

```
public void actionPerformed(ActionEvent e)
{
    int a=Integer.parseInt(t1.getText());
    int b=Integer.parseInt(t2.getText());
    int c=0;

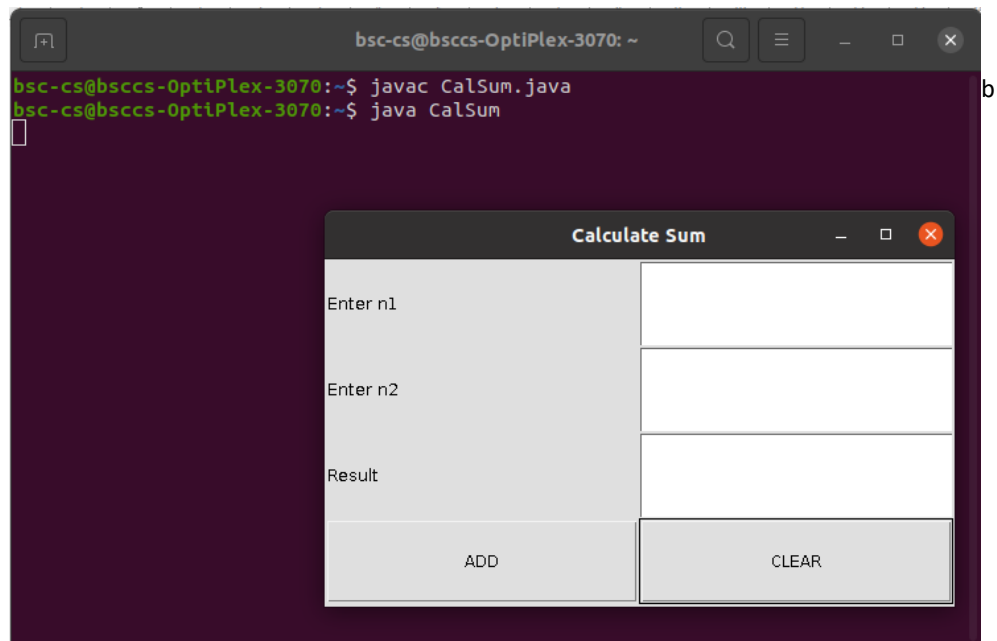
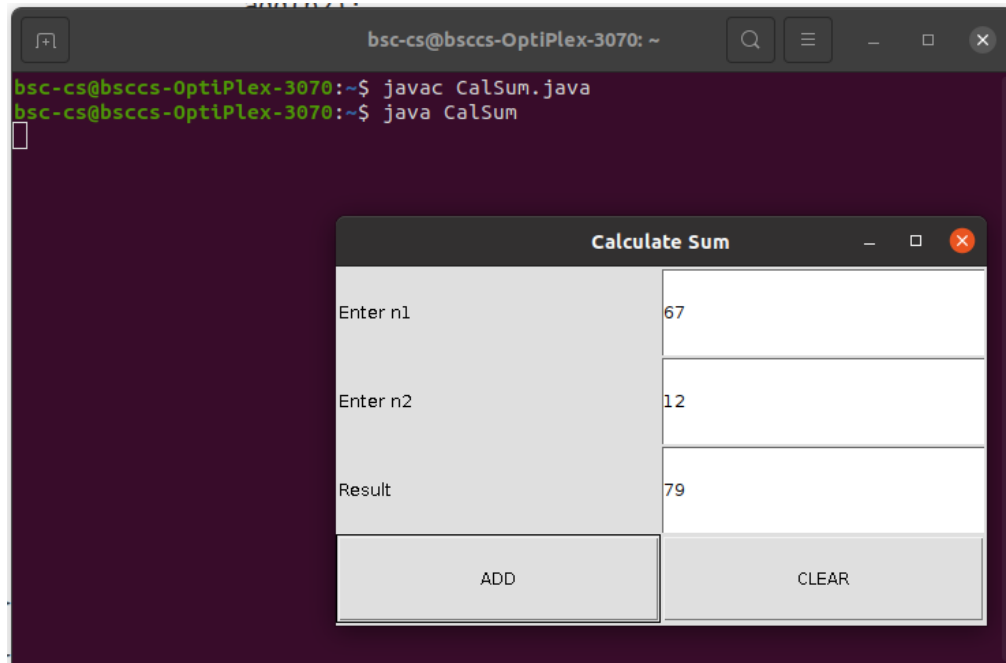
    c=a+b;
    t3.setText(String.valueOf(c));

    if(e.getSource().equals(b2))
    {
        t1.setText("");
        t2.setText("");
        t3.setText("");
    }
}

public static void main(String as[])
{
    CalSum f=new CalSum();
}
}
```

Java Lab Manual

output :



Java Lab Manual

9. Program to create a window, when we press M or m, the window displays “goodmorning”, A or a, the window displays “ Good Afternoon”, E or e, the window displays “good morning”,N or n, the window displays“good morning”

```
import java.awt.*;
import java.awt.event.*;
public class KeyWish extends Frame implements KeyListener
{
    String msg="";
    KeyWish()
    {
        setLayout(null);
        setSize(300,250);
        setTitle("Wish you a very");
        setVisible(true);
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });
        addKeyListener(this);
    }
    public void keyPressed(KeyEvent ke)
    {
        int key=ke.getKeyCode();
        switch(key)
        {
            case KeyEvent.VK_M:msg=msg+"Good Morning"; break;
            case KeyEvent.VK_A:msg=msg+"Good Afternoon"; break;
            case KeyEvent.VK_E:msg=msg+"Good Evening"; break;
            case KeyEvent.VK_N:msg=msg+"Good Night"; break;
            default: msg=msg+"Key Does not Match"; break;
        }
        repaint();
    }
    public void keyTyped(KeyEvent ke)
    {
        repaint();
    }
    public void keyReleased(KeyEvent ke)
    {
        repaint();
    }
    public void paint(Graphics g)
    {

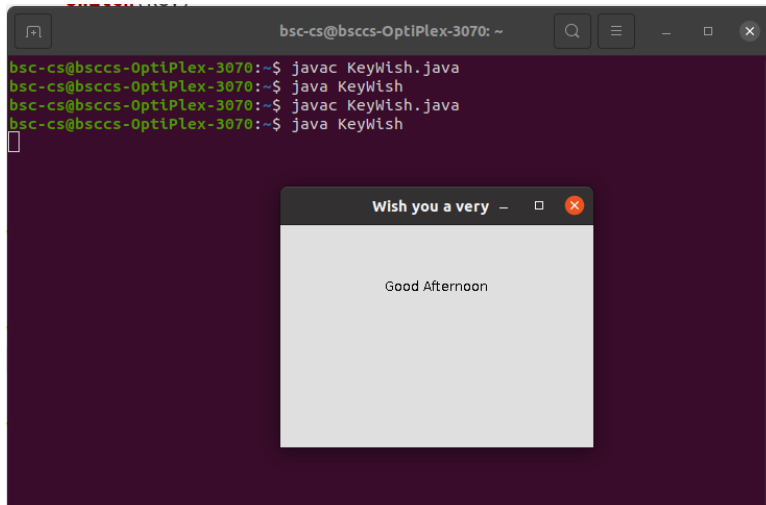
```

Java Lab Manual

```
        g.drawString(msg,100,100);
    }
    public static void main(String as[])
    {
        KeyWish m=new KeyWish();
    }
}
```

Java Lab Manual

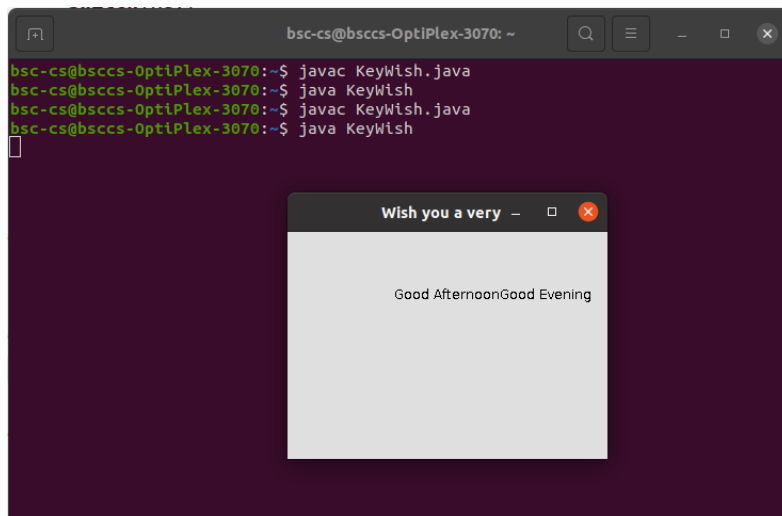
output :



```
bsc-cs@bsccs-OptiPlex-3070: ~  
bsc-cs@bsccs-OptiPlex-3070:~$ javac KeyWish.java  
bsc-cs@bsccs-OptiPlex-3070:~$ java KeyWish  
bsc-cs@bsccs-OptiPlex-3070:~$ javac KeyWish.java  
bsc-cs@bsccs-OptiPlex-3070:~$ java KeyWish
```

Wish you a very

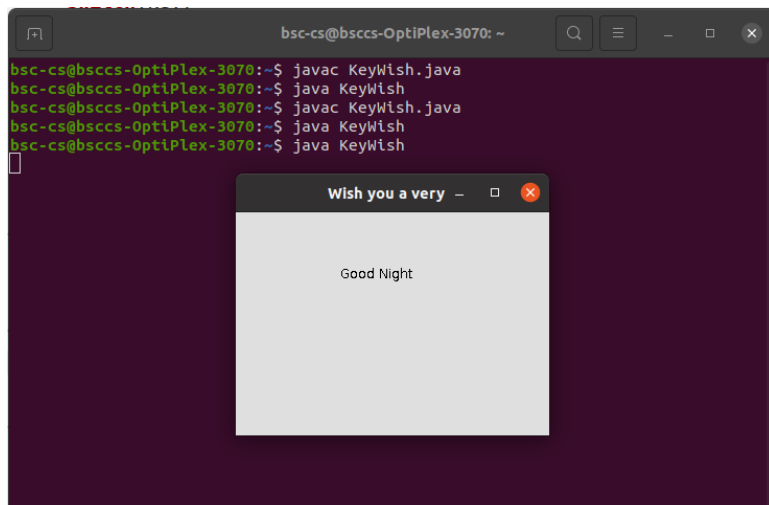
Good Afternoon



```
bsc-cs@bsccs-OptiPlex-3070: ~  
bsc-cs@bsccs-OptiPlex-3070:~$ javac KeyWish.java  
bsc-cs@bsccs-OptiPlex-3070:~$ java KeyWish  
bsc-cs@bsccs-OptiPlex-3070:~$ javac KeyWish.java  
bsc-cs@bsccs-OptiPlex-3070:~$ java KeyWish
```

Wish you a very

Good AfternoonGood Evening



```
bsc-cs@bsccs-OptiPlex-3070: ~  
bsc-cs@bsccs-OptiPlex-3070:~$ javac KeyWish.java  
bsc-cs@bsccs-OptiPlex-3070:~$ java KeyWish  
bsc-cs@bsccs-OptiPlex-3070:~$ javac KeyWish.java  
bsc-cs@bsccs-OptiPlex-3070:~$ java KeyWish  
bsc-cs@bsccs-OptiPlex-3070:~$ java KeyWish
```

Wish you a very

Good Night

Java Lab Manual

//10. Demonstrate the various mouse handling events using suitable example

```
import java.awt.*;

import java.awt.event.*;

public class Mousing extends Frame implements MouseListener
{
    String msg="";
    Mousing()
    {
        setLayout(null); setSize(300,250);
        setTitle("MouseEvents");
        setVisible(true);

        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });

        addMouseListener(this);
    }

    public void mouseClicked(MouseEvent me)
    {
        msg=msg+"Mouse Clicked["+me.getX()+","+me.getY()+"]";repaint();
    }

    public void mouseEntered(MouseEvent me)
    {
        msg=msg+"Mouse Entered["+me.getX()+","+me.getY()+"]";repaint();
    }

    public void mouseExited(MouseEvent me)
    {
        msg=msg+"Mouse Exited["+me.getX()+","+me.getY()+"]";repaint();
    }
}
```

Java Lab Manual

```
public void mousePressed(MouseEvent me)
{
    msg=msg+"Mouse Pressed["+me.getX()+","+me.getY()+"]";repaint();
}
public void mouseReleased(MouseEvent me)
{
    msg=msg+"Mouse Released["+me.getX()+","+me.getY()+"]";repaint();
}

public void paint(Graphics g)
{
    g.drawString(msg,50,50);
}

public static void main(String as[])
{
    Mousing m=new Mousing();
}
}
```

B.L.D.E Association's

S.B. Arts and K.C.P. Science College, Vijayapur

Department of Computer Science

LABORATORY MANUAL

Class : BSc IV Semester

Subject : Computer Science (Database Management System)

Course Code: 21BSC4C2CS2P

Practicals:

CO:Student would be able to create tables, execute queries and PL/SQL programs.

- 1. Execute a single line query and group functions.**
- 2. Execute DDL Commands.**
- 3. Execute DML Commands**
- 4. Execute DCL and TCL Commands.**
- 5. Implement the Nested Queries.**
- 6. Implement Join operations in SQL**
- 7. Create views for a particular table**
- 8. Implement Locks for a particular table**
- 9. Write PL/SQL procedure for an application using exception handling.**
- 10. Write PL/SQL procedure for an application using cursors.**
- 11. Write a PL/SQL procedure for an application using functions**
- 12. Write a PL/SQL procedure for an application using package**

Experiment 1 : Execute a single line query and group functions:

Single Row Function and Group Function

Single Row functions - Single row functions are the one who work on single row and return one output per row. For example, length and case conversion functions are single row functions.

Multiple Row functions - Multiple row functions work upon group of rows and return one result for the complete set of rows. They are also known as Group Functions. The group functions are used to calculate aggregate values like total or average, which return just one total or one average value after processing a group of rows.

Types of single row functions are:

- 1) Numeric Functions:** These are functions that accept numeric input and return numeric values.
- 2) Character or Text Functions:** These are functions that accept character input and can return both character and number values.
- 3) Date Functions:** These are functions that take values that are of data type DATE as input and return values of Data type DATE, except for the MONTHS_BETWEEN function, which returns a number.

1) Numeric Functions:

Numeric functions are used to perform operations on numbers. They accept numeric values as input and return numeric values as output. Few of the Numeric functions are:

Function Name	Return Value
ABS (x)	Absolute value of the number 'x'
CEIL (x)	Integer value that is Greater than or equal to the number 'x'
FLOOR (x)	Integer value that is Less than or equal to the number 'x'
TRUNC (x, y)	Truncates value of number 'x' up to 'y' decimal places
ROUND (x, y)	Rounded off value of the number 'x' up to the number 'y' decimal places

The following examples explain the usage of the above numeric functions

Function Name	Examples	Return Value
ABS (x)	ABS (1)	1
	ABS (-1)	-1
CEIL (x)	CEIL (2.83)	3

	CEIL (2.49) CEIL (-1.6)	3 -1
FLOOR (x)	FLOOR (2.83) FLOOR (2.49) FLOOR (-1.6)	2 2 -2
TRUNC (x, y)	ROUND (125.456, 1) ROUND (125.456, 0) ROUND (124.456, -1)	125.4 125 120
ROUND (x, y)	TRUNC (140.234, 2) TRUNC (-54, 1) TRUNC (5.7) TRUNC (142, -1)	140.23 54 5 140

These functions can be used on database columns.

For Example: Let's consider the product table used in sql joins. We can use ROUND to round off the unit_price to the nearest integer, if any product has prices in fraction.

SELECT ROUND (unit_price) FROM product;

2) Character or Text Functions:

Character or text functions are used to manipulate text strings. They accept strings or characters as input and can return both character and number values as output.

Few of the character or text functions are as given below:

Function Name	Return Value
LOWER (string_value)	All the letters in ' <i>string_value</i> ' is converted to lowercase.
UPPER (string_value)	All the letters in ' <i>string_value</i> ' is converted to uppercase.
INITCAP (string_value)	All the letters in ' <i>string_value</i> ' is converted to mixed case.
LTRIM (string_value, trim_text)	All occurrences of ' <i>trim_text</i> ' is removed from the left of ' <i>string_value</i> '.
RTRIM (string_value, trim_text)	All occurrences of ' <i>trim_text</i> ' is removed from the right of ' <i>string_value</i> ' .
TRIM (trim_text FROM string_value)	All occurrences of ' <i>trim_text</i> ' from the left and right of ' <i>string_value</i> ', ' <i>trim_text</i> ' can also be only one character long .
SUBSTR (string_value, m, n)	Returns ' <i>n</i> ' number of characters from ' <i>string_value</i> ' starting from the ' <i>m</i> ' position.
LENGTH (string_value)	Number of characters in ' <i>string_value</i> ' in returned.
LPAD (string_value, n, pad_value)	Returns ' <i>string_value</i> ' left-padded with ' <i>pad_value</i> ' . The length of

pad_value)	the whole string will be of 'n' characters.
RPAD (string_value, n, pad_value)	Returns 'string_value' right-padded with 'pad_value'. The length of the whole string will be of 'n' characters.

For Example, we can use the above UPPER() text function with the column value as follows.

```
SELECT UPPER (product_name) FROM product;
```

The following examples explain the usage of the above character or text functions

Function Name	Examples	Return Value
LOWER(string_value)	LOWER('Good Morning')	good morning
UPPER(string_value)	UPPER('Good Morning')	GOOD MORNING
INITCAP(string_value)	INITCAP('GOOD MORNING')	Good Morning
LTRIM(string_value, trim_text)	LTRIM ('Good Morning', 'Good')	Morning
RTRIM (string_value, trim_text)	RTRIM ('Good Morning', ' Morning')	Good
TRIM (trim_text FROM string_value)	TRIM ('o' FROM 'Good Morning')	Gd Mrning
SUBSTR (string_value, m, n)	SUBSTR ('Good Morning', 6, 7)	Morning
LENGTH (string_value)	LENGTH ('Good Morning')	12
LPAD (string_value, n, pad_value)	LPAD ('Good', 6, '*')	**Good
RPAD (string_value, n, pad_value)	RPAD ('Good', 6, '*')	Good**

3) Date Functions:

These are functions that take values that are of datatype DATE as input and return values of datatypes DATE, except for the MONTHS_BETWEEN function, which returns a number as output.

Few date functions are as given below.

Function Name	Return Value
ADD_MONTHS (date, n)	Returns a date value after adding 'n' months to the date 'x'.
MONTHS_BETWEEN (x1, x2)	Returns the number of months between dates x1 and x2.
ROUND (x, date_format)	Returns the date 'x' rounded off to the nearest century, year, month, date, hour, minute, or second as specified by the 'date_format'.
TRUNC (x, date_format)	Returns the date 'x' lesser than or equal to the nearest century, year, month, date, hour, minute, or second as specified by the 'date_format'.
NEXT_DAY (x, week_day)	Returns the next date of the 'week_day' on or after the date 'x' occurs.
LAST_DAY (x)	It is used to determine the number of days remaining in a month from the date 'x' specified.

SYSDATE	Returns the systems current date and time.
NEW_TIME (x, zone1, zone2)	Returns the date and time in zone2 if date 'x' represents the time in zone1.

The below table provides the examples for the above functions

Function Name	Examples	Return Value
ADD_MONTHS ()	ADD_MONTHS ('16-Sep-81', 3)	16-Dec-81
MONTHS_BETWEEN()	MONTHS_BETWEEN ('16-Sep-81', '16-Dec-81')	3
NEXT_DAY()	NEXT_DAY ('01-Jun-08', 'Wednesday')	04-JUN-08
LAST_DAY()	LAST_DAY ('01-Jun-08')	30-Jun-08
NEW_TIME()	NEW_TIME ('01-Jun-08', 'IST', 'EST')	31-May-08

SQL GROUP Functions

Group functions are built-in SQL functions that operate on groups of rows and return one value for the entire group. These functions are: **COUNT, MAX, MIN, AVG, SUM, DISTINCT**

Example : **Employee Table**

empID	ename	dept	salary
1020	Shashank	Civil	50000
1021	Mahesh	Civil	45000
1022	Jagadeesh	Electronics	75000
1023	Roopa	Electronics	70000
1024	Roopa	Marketing	120000

SQL COUNT (): This function returns the number of rows in the table that satisfies the condition specified in the WHERE condition. If the WHERE condition is not specified, then the query returns the total number of rows in the table.

a. If you want the number of employees in a particular department, the query would be:

```
SQL> SELECT COUNT (*) FROM employee
WHERE dept = 'Electronics';
```

b. If you want the total number of employees in all the department, the query would take the form:

```
SQL>SELECT COUNT (*) FROM employee;
```

SQL DISTINCT(): This function is used to select the distinct rows.

a. If you want to select all distinct department names from employee table, the query would be:

```
SQL>SELECT DISTINCT dept FROM employee;
```

b. To get the count of employees with unique name, the query would be:

```
SQL>SELECT COUNT (DISTINCT name) FROM employee;
```

SQL MAX(): This function is used to get the maximum value from a column.

To get the maximum salary drawn by an employee, the query would be:

```
SQL>SELECT MAX (salary) FROM employee;
```

SQL MIN(): This function is used to get the minimum value from a column.

To get the minimum salary drawn by an employee, the query would be:

```
SQL>SELECT MIN (salary) FROM employee;
```

SQL AVG(): This function is used to get the average value of a numeric column.

To get the average salary, the query would be

```
SQL>SELECT AVG (salary) FROM employee;
```

SQL SUM(): This function is used to get the sum of a numeric column

To get the total salary given out to the employees,

```
SQL>SELECT SUM (salary) FROM employee;
```

Experiment 2 : Execute DDL Commands

The DDL Commands in Structured Query Language are used to create and modify the schema of the database and its objects. The syntax of DDL commands is predefined for describing the data. The commands of Data Definition Language deal with how the data should exist in the database.

Following are the five DDL commands in SQL:

1. CREATE Command
2. DROP Command
3. ALTER Command
4. TRUNCATE Command
5. RENAME Command

CREATE Command

CREATE is a DDL command used to create databases, tables, triggers and other database objects.

SOLUTION:

1. Creating a Database
`CREATE DATABASE DATABASE_NAME;`
Ex. `CREATE DATABASE ABC;`
2. Viewing all databases
`SHOW DATABASES;`
3. Change database
`SQL>USE DATABASE_NAME`
Ex. `USE DATABASE ABC;`
4. Viewing all Tables in a Database,
`SQL>SHOW TABLES;`
5. Creating Tables (With and Without Constraints)

a. Creating table without using constraints

```
SQL> create table customer
2  (custno number(5),
3   cname varchar(20),
4   city varchar(20));
```

b. Creating table with using constraints

```
CREATE TABLE employee  
(empID int UNIQUE,  
ename varchar(20) NOT NULL,  
dname varchar(20),  
Salary FLOAT(10,3) CHECK (Salary>5000));
```

ALTER Command

ALTER is a DDL command which changes or modifies the existing structure of the database, and it also changes the schema of database objects.

ALTER: (a)ALTER TABLE ...ADD...: This is used to add some extra fields into existing relation.

Syntax: ALTER TABLE relation_name ADD (new field_1 data_type(size), new field_2 data_type(size),...);

Example:

```
SQL>ALTER TABLE customer ADD (Address CHAR(10));
```

```
SQL> desc customer;
```

(b)ALTER TABLE...MODIFY...: This is used to change the width as well as data type of fields of existing relations.

Syntax: ALTER TABLE relation_name MODIFY (field_1 newdata_type(Size), field_2 newdata_type(Size),...field_newdata_type(Size));

Example: SQL>ALTER TABLE student MODIFY(custno VARCHAR(10),cname VARCHAR(25));

```
SQL> desc customer;
```

c) ALTER TABLE..DROP.... This is used to remove any field of existing relations.

Syntax: ALTER TABLE relation_name DROP COLUMN (field_name);

Example: SQL>ALTER TABLE Employee DROP column dname;

```
SQL> desc customer;
```

d)ALTER TABLE..RENAME...: This is used to change the name of fields in existing relations.

Syntax: ALTER TABLE relation_name RENAME COLUMN (OLD field_name) to (NEW field_name);

Example: SQL>ALTER TABLE employee RENAME COLUMN ename to emp_name;

```
SQL> desc customer;
```

DROP Command

DROP is a DDL command used to delete/remove the database objects from the SQL database. We can easily remove the entire table, view, or index from the database using this DDL command.

Syntax to remove a database:

DROP DATABASE Database_Name;

Example : DROP DATABASE BSCCS001;

Syntax to remove a table:

DROP TABLE Table_Name;

SQL> DROP TABLE Employee;

RENAME Command

RENAME is a DDL command which is used to change the name of the database table.

Syntax of RENAME command

RENAME **TABLE** Old_Table_Name **TO** New_Table_Name;

SQL> rename table customer to customer_kar;

TRUNCATE Command

TRUNCATE is another DDL command which deletes or removes all the records from the table.

This command also removes the space allocated for storing the table records.

Syntax of TRUNCATE command

TRUNCATE TABLE Table_Name;

Example :

SQL> TRUNCATE TABLE Employee;

Experiment 3 : Execute DML Commands

Data Manipulation Language (DML) allows you to modify the database instance by inserting, modifying, and deleting its data. It is responsible for performing all types of data modification in a database.

There are three basic constructs which allow database program and user to enter data and information are:

DML commands in SQL are:

- INSERT
- UPDATE
- DELETE

INSERT:

This is a statement is a SQL query. This command is used to insert data into the row of a table.

Syntax:

```
INSERT INTO TABLE_NAME (col1, col2, col3,... col N)
VALUES (value1, value2, value3, .... valueN);
```

Or

```
INSERT INTO TABLE_NAME
VALUES (value1, value2, value3, .... valueN);
```

UPDATE:

This command is used to update or modify the value of a column in the table.

Syntax:

```
UPDATE table_name SET [column_name1= value1,...column_nameN = valueN]
[WHERE CONDITION]
```

DELETE:

This command is used to remove one or more rows from a table.

Syntax:

```
DELETE FROM table_name [WHERE condition];
```

Solution :

```
SQL> desc customer;
```

Insert the data into the customer table

```
SQL>INSERT INTO customer(custno, cname, city) VALUES ('601', 'Krishna', 'Bangalore');
```

```
SQL> INSERT INTO customer(custno, cname, city) VALUES ('602', 'Shital', 'Hubli');
```

```
SQL> INSERT INTO customer(custno, cname, city) VALUES ('603', 'Anupama', 'Hubli');
```

```
SQL> INSERT INTO customer(custno, cname, city) VALUES ('604', 'Geeta', 'Bangalore');
```

```
SQL> INSERT INTO customer(custno, cname, city) VALUES ('605', 'Raghavendra', 'Bagalkot');
```

Update the customer name to Prakash, whose city is 'bagalkot'

```
SQL>  UPDATE customer  
      SET cname = 'Prakash'  
      WHERE custno = 605;
```

Delete the customer information who belongs to bangalore

```
SQL>  DELETE FROM customer  
      WHERE city = 'Bangalore';
```

Delete all the records from customer table

```
SQL>  DELETE FROM customer ;
```

Experiment 4 : Execute DCL and TCL Commands.

DCL (Data Control Language) includes commands like GRANT and REVOKE, which are useful to give "rights & permissions." Other permission controls parameters of the database system.

Commands that come under DCL:

- Grant
- Revoke

Grant:

This command is used to grant permission on Database objects from one user to another user. i.e., give user access privileges to a database.

Syntax:

Grant <permissionlist> on <tablename> to <userlist> ;

```
GRANT SELECT, UPDATE ON Tablename TO SOME_USER, ANOTHER_USER;
```

Revoke:

It is useful to back permissions from the user. This command is use to cancel given permissions

Syntax:

```
REVOKE privilege_name ON object_name FROM {user_name |PUBLIC |role_name}
```

REVOKE <permissionlist> on <tablename> from <userlist>

Solution :

Grant the access permission of customer table to the user bsc-cs

```
SQL>GRANT SELECT, UPDATE ON customers TO 'bsc-cs@'localhost;
```

Revoke the access permissions of customer table given to multiple users

```
SQL>REVOKE SELECT, UPDATE ON customers TO 'bsc-cs@'localhost;
```

TCL commands:

Transaction control language or TCL commands deal with the transaction within the database.

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.

Here are some commands that come under TCL:

- COMMIT
- ROLLBACK
- SAVEPOINT

Commit

This command is used to save all the transactions to the database.

Syntax:

```
Commit;
```

For example:

```
DELETE FROM Employee  
WHERE empID =2;  
  
COMMIT;
```

Rollback

Rollback command allows you to undo transactions that have not already been saved to the database.

Syntax:

```
ROLLBACK;
```

```
DELETE FROM Employee  
WHERE empID =5;  
  
ROLLBACK;;
```

Experiment 5 : Implement the Nested Queries.

A nested query is a query that has another query embedded within it. The embedded query is called a subquery.

A subquery typically appears within the WHERE clause of a query. It can sometimes appear in the FROM clause or HAVING clause.

The SELECT query inside the brackets () is the inner query, and the SELECT query outside the brackets is the outer query. The outer query uses the result of the inner query.

The general syntax of nested queries will be:

```
SELECT column_name [, column_name ]  
FROM table1 [, table2 ]  
WHERE column_name OPERATOR  
(  
    SELECT column_name [, column_name ]  
    FROM table1 [, table2 ]  
    [WHERE]  
)
```

Solution :

SQL : CREATE TABLE employee

```
( employee_id INT PRIMARY KEY,  
  name VARCHAR2(100) NOT NULL,  
  salary FLOAT(15,2) CHECK (salary>5000) ,  
  role VARCHAR2(100) NOT NULL  
);
```

SQL : CREATE TABLE awards

```
( id int PRIMARY KEY,  
  employee_id INT NOT NULL,  
  award_date DATE NOT NULL  
);
```

SQL > INSERT INTO employees VALUES (1, 'Amrut Patel', 10000, 'Developer');

SQL > INSERT INTO employees VALUES (2, 'Sneha Shah', 10000, 'Manager');

SQL > INSERT INTO employees VALUES (3, 'Stefen', 30000, 'Developer');

SQL > INSERT INTO employees VALUES (4, 'Mahesh', 40000, 'Manager');

SQL > INSERT INTO employees VALUES (5, 'Sayed', 50000, 'Developer');

SQL > /

Employee

employee_id	name	salary	role
1	Amrut Patel	10000	Developer
2	Sneha Shah	10000	Manager
3	Stefen	30000	Developer
4	Mahesh	40000	Manager
5	Sayed	50000	Developer

SQL >INSERT INTO awards VALUES(1, 1, '2022-04-01');

SQL >INSERT INTO awards VALUES(2, 3, '2022-05-01');

Awards

id	employee_id	award_date
1	1	2022-04-01
2	3	2022-05-01

Independent Nested Queries

a. Select all employees who won an award.

SQL > SELECT employee_id, name FROM employee
WHERE employee_id IN (SELECT employee_id FROM awards);

b. Select all employees who never won an award.

SQL > SELECT employee_id, name FROM employee
WHERE employee_id NOT IN (SELECT employee_id FROM awards);

Co-related Nested Queries

a. Select all employees whose salary is above the average salary of employees in their role.

SQL >SELECT * FROM employee1 emp1
WHERE salary > (SELECT AVG(salary) FROM employee emp2
WHERE emp1.role = emp2.role);

Experiment 6:

The SQL `JOIN` joins two tables based on a common column, and selects records that have matching values in these columns.



```
SQL > CREATE TABLE CUSTOMERS
      (CUSTOMER_ID INT PRIMARY KEY,
      FIRST_NAME VARCHAR(20));
```

```
SQL > CREATE TABLE ORDERS
      (ORDER_ID INT PRIMARY KEY,
      AMOUNT INT,
      CUSTOMER_ID INT);
```

```
SQL > SELECT CUSTOMERS.CUSTOMER_ID, ORDERS.FIRST_NAME, ORDERS.AMOUNT FROM ORDERS
      INNER JOIN CUSTOMERS
      ON ORDERS.CUSTOMER_ID=CUSTOMERS.CUSTOMER_ID;
```

Experiment 7 : Create views for a particular table

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

Syntax

```
CREATE VIEW view_name AS  
SELECT column1, column2,.... columnN  
FROM table_name  
WHERE condition;
```

Create the following Student Table:

STU_ID	NAME	ADDRESS	MARKS	AGE
1	Amita	Vijayapur	97	19
2	Viren	Athani	86	20
3	Shraddha	Jatt	95	19
4	Kartik	Vijayapur	90	18

```
SQL> CREATE TABLE STUDENT
```

```
(STU_ID INT PRIMARY KEY,
```

```
NAME VARCHAR(20),
```

```
ADDRESS VARCHAR(20),
```

```
MARKS INT,
```

```
AGE INT);
```

Create view

```
SQL> CREATE VIEW STD_INFO
```

```
AS STU_ID, NAME, MARKS
```

```
FROM STUDENT
```

```
WHERE AGE>18;
```

SQL>/

STU_ID	NAME	MARKS
1	Amita	97
2	Viren	86
3	Shraddha	95

Experiment 8. Implement Locks for a particular table

```
SQL>CREATE TABLE sales (  
order_id int,  
salesman_name character varying(50),  
product_id character varying(50),  
sales_region character varying(50),  
sales_date date  
);
```

```
SQL>INSERT INTO public.sales(  
order_id, salesman_name, product_id, sales_region, sales_date)  
VALUES (1,'Mohit K','Book11','New Delhi','2020-05-01');
```

```
SQL>SELECT * FROM sales;
```

Explain

Notifications

Messages

Data Output

	order_id integer	salesman_name character varying (50)	product_id character varying (50)	sales_region character varying (50)	sales_date date
1	1	Mohit K	Book11	New Delhi	2020-05-01
2	2	Rey Holt	Book11	Mumbai	2020-05-02
3	3	Swati Singh	Book24	New Delhi	2020-05-03
4	4	Indrani K	Book24	Mumbai	2020-05-01
5	5	Dave Prakash	Book11	Mumbai	2020-05-02
6	6	Joshua S	Book24	New Delhi	2020-05-03
7	7	Mrinali Pal	Book11	New Delhi	2020-05-04
8	8	Mohit K	Book24	New Delhi	2020-05-02
9	9	Rey Holt	Book24	Mumbai	2020-05-05
10	10	Indrani K	Book11	Mumbai	2020-05-04
11	11	Joshua S	Book24	New Delhi	2020-05-05
12	12	Mohit K	Book11	New Delhi	2020-05-04

```
SQL>CREATE TABLE sales_count(  
sales_region VARCHAR,  
tot_orders INT  
);
```

```
BEGIN WORK;  
LOCK TABLE sales IN SHARE MODE;  
INSERT INTO sales_count(sales_region,tot_orders)  
SELECT sales_region, count(order_id) as tot_orders  
FROM sales  
GROUP BY sales_region;  
COMMIT WORK;
```

```
SQL>SELECT * FROM sales_count;
```

	sales_region character varying	tot_orders integer
1	Mumbai	5
2	New Delhi	7

Experiment 9. Write a PL/SQL program to demonstrate Exceptions.

```
DECLARE Eid emp.empno%type:= 69000;
name emp.ename%type;
BEGIN SELECT empno,ename INTO eid,name FROM emp WHERE empno = eid;
DBMS_OUTPUT.PUT_LINE ('empno: ' || eid);
DBMS_OUTPUT.PUT_LINE ('Name: ' || name);
EXCEPTION WHEN no_data_found THEN dbms_output.put_line('No such employee!');
WHEN others THEN dbms_output.put_line('Error!');
END;
/
```

Output

empno: 69000

Name: JULIUS

PL/SQL procedure successfully completed.

Experiment 10. Write a PL/SQL program to demonstrate Cursors.

```
SQL>declare cursor c11 is select * from emp order by sal,,job;
```

```
cval emp%rowtype;
```

```
begin dbms_output.put_line('empno '||'ename');
```

```
open c11;
```

```
loop fetch c11 into cval;
```

```
exit when c11%rowcount=10;
```

```
dbms_output.put_line(cval.empno||' '||cval.ename);
```

```
end loop;
```

```
close c11;
```

```
end;
```

```
/
```

Output

Experiment 11: Write a PL/SQL procedure for an application using functions

```
SQL> create or replace function fname(a in number,b in out number) return number is begin b:=a;  
return b;  
end;  
SQL> /
```

Function created.

Program to invoke a function

```
SQL> declare x number;  
begin x:=fname(23,x);  
dbms_output.put_line(x);  
end;
```

Output

```
SQL> /
```

23 PL/SQL procedure successfully completed.

Experiment 12: Write PL/SQL queries to create Procedures.

SQL> create or replace procedure emp_proc1 IS

BEGIN update emp

set sal=sal+sal*0.10

where comm<>sal*0.09;

END;

SQL> /

Procedure created. Executing Procedure:

SQL> execute emp_proc1;

Output

PL/SQL procedure successfully completed.

Experiment 13. Write a PL/SQL procedure for an application using package

Step1: Creating package specification CREATE or replace PACKAGE emppackage AS procedure emp_proc2;
function fname2(a in number,b in out number) return number;

end;
Step2: Creating package definition or body CREATE or replace PACKAGE body emppackage AS procedure
emp_proc2 is

```
begin
update emp
set sal=sal+sal*0.10
where comm<>sal*0.09;
  DBMS_OUTPUT.PUT_LINE ('I am a procedure ');
end emp_proc2;
function fname2(a in number, b in out number) return number is begin b:=a;
return b;
end fname2;
END;
```

Step3: Calling function / procedure of a package in a program declare a number:=10;

```
b number:=10;
begin a:=emppackage.fname2(20,b);
  dbms_output.put_line(a || b);
emppackage.emp_proc2;
end;
/
```

Output

2020 I am a procedure PL/SQL procedure successfully completed.

**BLDEA's
S.B.ARTS AND K.C.P. SCIENCE COLLEGE, VIJAYAPUR
DEPARTMENT OF COMPUTER SCIENCE**

Lab Manual

Python Programming

Course Code:DSC5-Lab

BSC-V Semester

2023-24

PROGRAM EDUCATIONAL OBJECTIVES (PEOS):

A graduate of the Computer Science and Engineering Program should:

PEO1	Program Educational Objective1: (PEO1) The Graduates will provide solutions to difficult and challenging issues in their profession by applying computer science and engineering theory and principles.
PEO2	Program Educational Objective2 :(PEO2) The Graduates have successful careers in computer science fields or will be able to successfully pursue advanced degrees.
PEO3	Program Educational Objective3: (PEO3) The Graduates will communicate effectively, work collaboratively and exhibit high levels of Professionalism, moral and ethical responsibility.
PEO4	Program Educational Objective4 :(PEO4) The Graduates will develop the ability to understand and analyse Engineering issues in a broader perspective with ethical responsibility towards sustainable development.

PROGRAM SPECIFIC OUTCOMES(PSOS):

PSO1	Problem Solving Skills – Graduate will be able to apply computational techniques and software principles to solve complex engineering problems pertaining to software engineering.
PSO2	Professional Skills – Graduate will be able to think critically, communicate effectively, and collaborate in teams through participation in co and extra-curricular activities.
PSO3	Successful Career – Graduates will possess a solid foundation in computer science that will enable them to grow in their profession and pursue lifelong learning through professional development.

Course Outcomes (CO's)

Upon completion of this course, the student will be able to:

CO1: Write, test, and debug simple Python programs.

CO2: Implement Python programs with conditionals and loops.

CO3: Develop Python programs step-wise by defining functions and calling them.

CO4: Use Python lists, tuples, dictionaries for representing compound data, Read and write data from/to files in Python

CO5: Develop Programs using GUI Widgets

CO6: Develop Python programs to establish connection with MySql

List of Programs :

S.No.	Topic	Page number
1	Write a Python function to calculate the factorial of a number	
2	Write a Python to generate Fibonacci Sequence	
3	Write a Python program to get the sum of digits of a non-negative integer	
4	Write a Python program to create a module Calculation.py that contains functions to perform basic arithmetic operations.	
5	Write a python program to reverse a string without using built-in functions	
6	Write a python program to generate random numbers.	
7	Write a python program to display Multiplication Tables	
8	Demonstrate importing the math module and perform any five math functions.	
9	Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.	
10	Write a Python class named Rectangle constructed by a length and width and a method which will compute the area and perimeter of rectangle.	

Practice Programs:

11	Demonstrate usage of basic regular expression with match (), search (), findall (), sub () and split ().	
12	Find the largest and smallest element in the list	
13	Demonstrate use of Dictionaries to store and retrieve contact information.	
14	Create SQLite Database and Write a Python program to demonstrate modification of an existing table data from SQLite Database	
15	Write a python program that prompts the user for a number and handles a "ValueError"	
16	Inherit a class Box that contains additional method volume. Override the perimeter method to compute perimeter of a Box.	
17	Write a Python program to read a file line by line store it into an array.	
18	Write a python program to create a class representing a basic bank account class with deposit and withdrawal methods.	
19	Design Student Registration form using any 5 widgets using Tkinter Module.	
20	Write a python program to create a GUI interface for temperature converter using Tkinter	

1. Python function to calculate the factorial of a number

```
def factorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n-1)  
n=int(input("Input a number to compute the factorial : "))  
print(factorial(n))
```

OUTPUT :

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ python3 factorial.py
```

```
enter the number for factorial 5
```

```
factorial of 5 is 120
```

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ python3 factorial.py
```

```
enter the number for factorial 6
```

```
factorial of 6 is 720
```

2. Python to generate Fibonacci Sequence

```
def fibonacci(n):  
    if n<=1:  
        return n  
    else:  
        return(fibonacci(n-1)+fibonacci(n-2))  
num=int(input("How many terms you want to display:"))  
for i in range(num):  
    print(fibonacci(i)," ", end=" ")
```

OUTPUT:

```
bsc-cs@bsccs-OptiPlex-3070:~$ python3 fib.py
```

```
How many terms you want to display:6
```

```
0   1   1   2   3   5
```

3. Python program to print sum of digits of non-negative integer number

```
num=int(input("Enter a positive integer number : "))
result=0
if num<0:
    print("Entered negative number")
else:
    while num>0:
        digit=int(num%10)
        result=result+digit
        num=int(num/10)
    print("sum is :",result)
```

OUTPUT :

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ gedit sum_digitExp2.py
```

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ python3 sum_digitExp2.py
```

```
Enter a positive integer number 1234
```

```
sum is : 10
```

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ python3 sum_digitExp2.py
```

```
Enter a positive integer number : -458
```

```
Entered negative number
```

4. Python program to create a module **Calculation.py** that contains functions to perform basic arithmetic operations. Demonstrate importing the module.

File name: Calculation.py

```
def add(a, b):  
    return a + b
```

```
def sub(a, b):  
    return a - b
```

```
def mul(a, b):  
    return a * b
```

```
def div(a, b):  
    return a / b
```

File name: Arithmetic.py

```
import Calculation  
num1=int(input('Enter first number: '))  
num2=int(input('Enter second number: '))  
  
print("Addition=",Calculation.add(num1, num2))  
print("Subtraction=",Calculation.sub(num1, num2))  
print("Multiplication=",Calculation.mul(num1, num2))  
print("Division=",Calculation.div(num1, num2))
```

OUTPUT:

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ python3 Arithmetic.py
```

```
Enter first number: 56
```

```
Enter second number: 12
```

```
Addition= 68
```

```
Subtraction= 44
```

```
Multiplication= 672
```

```
Division= 4.666666666666667
```

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ python3 Arithmetic.py
```

```
Enter first number: 23
```

```
Enter second number: 5
```

```
Addition= 28
```

```
Subtraction= 18
```

```
Multiplication= 115
```

```
Division= 4.6
```

5. Python program to reverse a string without using built-in functions

```
my_string=input("Enter the string to reverse")  
  
print("Original String is :",my_string)  
  
rev=""  
  
for i in my_string:  
    rev=i+rev  
  
print("Reversed string:",rev)
```

Output :

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ python3 reverse123.py
```

Enter the string to reverse welcome

welcome

Original String is : welcome

Reversed string: emoclew

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ python3 reverse123.py
```

Enter the string to reverse computer science

computer science

Original String is : computer science

Reversed string: ecneics retupmoc

6. Python program to generate random numbers.

```
import random

randomlist = []

for a in range(0,4):

    q=random.randint(1,30)

    randomlist.append(q)

print("List of random numbers is :\n")

print(randomlist)
```

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ python3 rand123.py
```

List of random numbers is :

```
[8, 28, 24, 26]
```

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ python3 rand123.py
```

List of random numbers is :

```
[16, 29, 12, 2]
```

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ python3 rand123.py
```

List of random numbers is :

```
[5, 15, 18, 19]
```

7. Python program to display Multiplication Tables

```
number=int(input("Enter the number"))  
print("Multiplication Table for ",number)  
for i in range(1,11):  
    print(number,"X",i,"=",number*i)
```

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ python3 MultiplicationT.py
```

```
Enter the number 6
```

```
Multiplication Table for 6
```

```
6 X 1 = 6
```

```
6 X 2 = 12
```

```
6 X 3 = 18
```

```
6 X 4 = 24
```

```
6 X 5 = 30
```

```
6 X 6 = 36
```

```
6 X 7 = 42
```

```
6 X 8 = 48
```

```
6 X 9 = 54
```

```
6 X 10 = 60
```

```
bsc-cs@bsccs-OptiPlex-3070:~/RRH$ python3 MultiplicationT.py
```

```
Enter the number23
```

```
multiplication Table for 23
```

```
23 X 1 = 23
```

```
23 X 2 = 46
```

```
23 X 3 = 69
```

```
23 X 4 = 92
```

```
23 X 5 = 115
```

```
23 X 6 = 138
```

```
23 X 7 = 161
```

```
23 X 8 = 184
```

```
23 X 9 = 207
```

```
23 X 10 = 230
```

8. Demonstrate importing the math module and perform any five math functions.

```
import math
```

```
a = math.sin(90)
```

```
print(r"sin of 45 =",a)
```

```
b = math.factorial(5)
```

```
print("Factorial of 5 is :",b)
```

```
c = math.floor(12.6567)
```

```
print("math.floor(12.6567) =",c)
```

```
d = math.ceil(12.4567)
```

```
print("math.ceil(12.4567) =",d)
```

```
e = math.exp(5)
```

```
print("Exponent of 5 = ",e)
```

```
f = math.log(5)
```

```
print("log(5) = ",f)
```

```
g = math.log10(5)
```

```
print("log10(5) = ",g)
```

```
h = math.pow(2,6)
```

```
print("math.pow(2,6) =",h)
```

```
i = math.sqrt(625)
```

```
print("square root of 625 =",i)
```

output :

bsc-cs@bsccs-OptiPlex-3070:~/RRH\$ python3 mathModule.py

sin of 45 = 0.8939966636005579

Factorial of 5 is : 120

math.floor(12.6567) = 12

math.ceil(12.4567) = 13

Exponent of 5 = 148.4131591025766

log(5) = 1.6094379124341003

log10(5) = 0.6989700043360189

math.pow(2,6) = 64.0

square root of 625 = 25.0

9. Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.

```
class Circle:

    def Area(self,r):
        self.radius=r
        r=3.142*self.radius*self.radius
        print("Area of Circle=",r)
    def Perimeter(self,r):
        self.radius=r
        p=2*3.142*self.radius
        print("Perimeter of Circle=",p)

obj=Circle()
r=int(input("Enter radius for circle"))
obj.Area(r)
obj.Perimeter(r)
```

OUTPUT :

```
bsc-cs@bsccs-OptiPlex-3070:~$ python3 AreaCircle.py
Enter radius for circle4
Area of Circle= 50.272
Perimeter of Circle= 25.136
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ python3 AreaCircle.py
Enter radius for circle 7
Area of Circle= 153.958
Perimeter of Circle= 43.988
```

10. Write a Python class named Rectangle constructed by a length and width and a method which will compute the area and perimeter of rectangle

```
import math

class rectangle():
    def __init__(self,breadth,length):
        self.breadth=breadth
        self.length=length
    def area(self):
        return self.breadth*self.length
    def perimeter(self):
        return 2*(self.breadth+self.length)
a=int(input("Enter length of rectangle: "))
b=int(input("Enter breadth of rectangle: "))
obj=rectangle(a,b)
print("Area of rectangle:",obj.area())
print("Area of rectangle:",obj.perimeter())

print()
```

Output

```
bsc-cs@bsccs-OptiPlex-3070:~$ python3 rectangle.py
```

Enter length of rectangle: 5

Enter breadth of rectangle: 6

Area of rectangle: 30

Area of rectangle: 22

Part – B

- 1. Demonstrate usage of basic regular expression with match (), search (), findall (), sub() and split().**

```
import re
line="Python and Java supports OOPS concept and both are easy to learn"
#Regular Expression for Search()
r1=re.match(r'Java',line)
if r1:
    print("Match is found")
else:
    print("Match not found")
#Regular Expression for Search()
r3=re.search(r'OOPS',line)
if r3:
    print("Match is found")
else:
    print("Match not found")
r4=re.sub(r'OOPS','Procedural',line)
print(r4)
r4=re.findall(r'DS|CPP',line)
print(r4)

r5=re.split(r'a',line)
print(r5)
```

OUTPUT:

```
bsc-cs@bsccs-OptiPlex-3070:~$ python3 match1.py
```

```
Match not found
```

```
Match is found
```

```
Python and Java supports Procedural concept and both are easy to learn
```

```
[]
```

```
['Python ', 'nd J', 'v', ' supports OOPS concept ', 'nd both ', 're e', 'sy to le', 'rn']
```

2. Find the largest and smallest element in the list

```
L=[]
n=int(input("Enter the number of elements to be inserted into list: "))
for x in range(n):
    element=int(input(f"enter the {x+1} element to be inserted into list:" ))
    L.append(element)
smallest=L[0]
largest=L[0]
for i in range(n):
    if smallest>L[i]:
        smallest=L[i]
    if largest<L[i]:
        largest=L[i]
print("The smallest number in the list is : ", smallest)
print("The largest number in the list is : ", largest)
```

Output :**bsc-cs@bsccs-OptiPlex-3070:~\$ python3 SmallLarge.py**

Enter the number of elements to be inserted into list: 5

Enter the 1 element to be inserted into list:12

Enter the 2 element to be inserted into list:56

Enter the 3 element to be inserted into list:23

Enter the 4 element to be inserted into list:46

Enter the 5 element to be inserted into list:45

The smallest number in the list is : 12

The largest number in the list is : 56

3. Demonstrate use of Dictionaries to store and retrieve contact information.

```
n = int(input("Enter how many names you want to enter: "))
# initialize empty dictionary
names={}
for i in range(n):
    name=input("Enter name of friend: ")
    number=input("Enter phone number: ")
    #add name number to dictionary
    names[name]=number
print(names)

#add new item
names["Arun"]="9877666234"
print("Modified dictionary ",names)

#delete an item
del names["abc"]

#modify first key value
for name in names:
    names[name] = "9456356344"
    break

#Dictionary after modifying first key value
print("Dictionary after modifying first key value\n")
print(names)
```

Output :

bsc-cs@bsccs-OptiPlex-3070:~\$ python3 contactinfo.py

Enter how many names you want to enter: 2

Enter name of friend: abc

Enter phone number: 698678607

Enter name of friend: xyz

Enter phone number: 678656958

{'abc': '698678607', 'xyz': '678656958'}

Modified dictionary {'abc': '698678607', 'xyz': '678656958', 'Arun': '9877666234'}

Dictionary after modifying first key value

{'xyz': '9456356344', 'Arun': '9877666234'}

4. Write a python program that prompts the user for a number and handles a “ValueError”

A ValueError in Python is raised when a function receives an argument of the correct #type but an inappropriate value. To handle it, you can use a try-except block to catch the #error and handle it appropriately.

```
def square_root(n):  
    if n < 0:  
        raise ValueError('number must be a non-negative number')  
    return n ** 0.5  
  
try:  
    x=int(input("Enter a number"))  
    y=square_root(x)  
    print(f"Square root of {x} is {y}")  
  
except ValueError as e:  
    print(e)
```

Output :

bsc-cs@bsccs-OptiPlex-3070:~\$ python3 ValueErrorEx1.py

Enter a number -6

-6

number must be a non-negative number

bsc-cs@bsccs-OptiPlex-3070:~\$ python3 ValueErrorEx1.py

Enter a number 8

8

Square root of 8 is 2.8284271247461903

5. Inherit a class Box that contains additional method volume. Override the perimeter method to compute perimeter of a Box.

```
class Rectangle:
    # define constructor with attributes: length and width
    def __init__(self, length , width):
        self.length = length
        self.width = width

    # Create Perimeter method
    def Perimeter(self):
        return 2*(self.length + self.width)

    # Create area method
    def Area(self):
        return self.length*self.width

    # create display method
    def display(self):
        print("The length of rectangle is: ", self.length)
        print("The width of rectangle is: ", self.width)
        print("The perimeter of rectangle is: ", self.Perimeter())
        print("The area of rectangle is: ", self.Area())

class Box(Rectangle):
    def __init__(self, length, width , height):
        Rectangle.__init__(self, length, width)
        self.height = height

    # define Volume method
    def volume(self):
        return self.length*self.width*self.height
    def Perimeter(self):
        return 4*(self.length + self.width+self.height)

myRectangle = Rectangle(7 , 5)
myRectangle.display()
print("-----")
B1 = Box(7 , 5 , 3)
print("the Perimeter of Box is: " , B1.Perimeter())
print("the volume of Box is: " , B1.volume())
```

Output :

bsc-cs@bsccs-OptiPlex-3070:~\$ python3 Rectangle1.py

The length of rectangle is: 7

The width of rectangle is: 5

The perimeter of rectangle is: 24

The area of rectangle is: 35

the Perimeter of Box is: 60

the volume of Box is: 105

7. **Demonstrate importing the module. Write a Python program to read a file line by line store it into an array.**

File 1 : test.txt

Welcome to Python Lab

Python is very user friendly programming Language

Python with compact code

This is an example to extract lines from file and store them in as array

File 2: ReadFile.py

```
content_array=[]
```

```
with open("test.txt","r") as file:
```

```
    lines=file.readlines()
```

```
for i in lines:
```

```
    content_array=i.strip() # this strip fn removes new line character from each line
```

```
    print(content_array)
```

OUTPUT :

```
bsc-cs@bsccs-OptiPlex-3070:~$ python3 ReadFile.py
```

Welcome to Python Lab

Python is very user friendly programming Language

Python with compact code

This is an example to extract lines from file and store them in as array

- 8. Write a python program to create a class representing a basic bank account class with deposit and withdrawal methods.**

```
class Bank_Account:
    def __init__(self):
        self.balance=0
        print("Hello!!! Welcome to the Deposit & Withdrawal Machine")

    def deposit(self):
        amount=float(input("Enter amount to be Deposited: "))
        self.balance += amount
        print("\n Amount Deposited:",amount)

    def withdraw(self):
        amount = float(input("Enter amount to be Withdrawn: "))
        if self.balance>=amount:
            self.balance-=amount
            print("\n You Withdrew:", amount)
        else:
            print("\n Insufficient balance ")

    def display(self):
        print("\n Net Available Balance=",self.balance)

# Driver code

# creating an object of class
s = Bank_Account()

# Calling functions with that class object
s.deposit()
s.withdraw()
s.display()
```

9. Write a Python GUI program to design Student Registration Form using any 5 widgets.

Some of the common **widgets** used in Tkinter are :

- **Frame** : serves as a holding area for other widgets and serves as a container.
- **Text** : It enables us to display and alter text in a variety of styles and offers a prepared text display.
- **Label** : Used to display text and images, but we are unable to interact with it.
- **Button** : Often used add buttons and we may add functions and methods to it.
- **Entry** : One-line string text can be entered into this widget.
- **Labelframe** : For intricate window layouts, this widget serves as a separator or container.
- **Listbox** : It just has text elements, all of which are the same colour and font.
- **Scrollbar** : This gives a sliding controller.
- **Canvas** : Custom widgets can be implemented using the canvas widget.
- **Scale** : This widget offers graphical slider items that let us choose different scale values.
- **Radiobutton** : Use a radio button to carry out one of several choices.
- **Checkbox** : Use a checkbox to implement on-off choices.
- **Listbox** : It just has text elements, all of which are the same colour and font.

How to start making a simple registration form using Tkinter :

- **Step 1** : The first step is to import the tkinter module (using either `tkinter import *` or just `import tkinter`).
- **Step 2** : The primary window of the GUI programme was created.
- **Step 3** : Include one or more widgets in the GUI programme (controls such as buttons, labels, and text boxes, etc.).
- **Step 4** : Enter the primary events to react to each event that the user has triggered.

```
import tkinter as tk
import tkinter.messagebox as box

#creating window
base=tk.Tk()
base.geometry('500x500')
base.title("Registration Form")

#Label for Form Heading
label0 =tk.Label(base, text="Registration form",width=20,font=("bold", 20))
label0.place(x=90,y=53)

# Label and Entry for student name
label1 =tk.Label(base, text="FullName",width=20,font=("bold", 10))
label1.place(x=80,y=130)

entry_1 =tk.Entry(base)
entry_1.place(x=240,y=130)

#Label and Entry for EmailID
labl_2 =tk.Label(base, text="Email",width=20,font=("bold", 10))
labl_2.place(x=68,y=180)

entry_02 =tk.Entry(base)
entry_02.place(x=240,y=180)

#Label and Radiobutton for Gender
labl_3 =tk.Label(base, text="Gender",width=20,font=("bold", 10))
labl_3.place(x=70,y=230)

radio_var =tk.IntVar()
radio1=tk.Radiobutton(base, text="Male",padx = 3, variable=radio_var, value=1)
radio2=tk.Radiobutton(base, text="Female",padx = 15, variable=radio_var,
value=2)
radio1.place(x=235,y=230)
radio2.place(x=300,y=230)

#Label and Entry for Age
labl_4 =tk.Label(base, text="Age:",width=20,font=("bold", 10))
labl_4.place(x=70,y=280)

entry_02 =tk.Entry(base)
entry_02.place(x=240,y=280)
```

```
# combobox for Selection of class
class_label=tk.Label(base, text="Select the Class")
class_label.place(x=310,y=320)

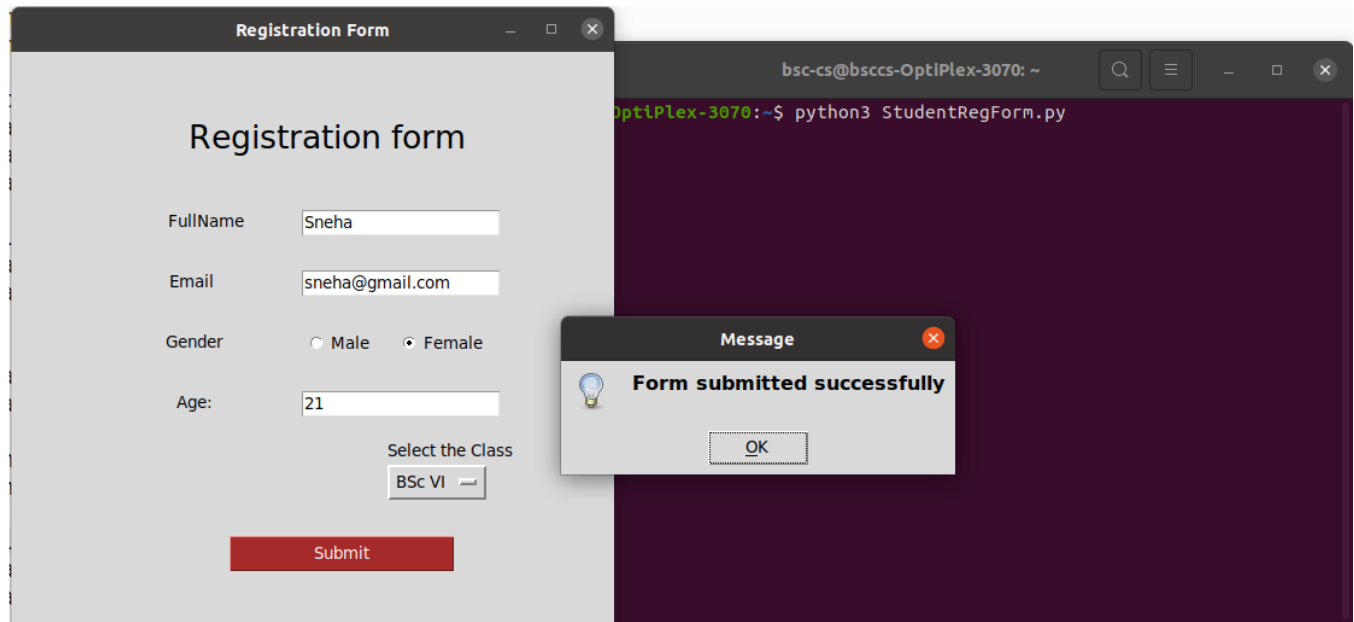
combo1=tk.StringVar()
combo1.set("Select")
choice1=["BSc VI", "BSc IV","BSc II"]
class_dropdown=tk.OptionMenu(base, combo1,*choice1)
class_dropdown.place(x=310, y=340)

def exbutton():
    box.showinfo("Message","Form submitted successfully")

bt=tk.Button(base,
text='Submit',width=20,bg='brown',fg='white',command=exbutton)
bt.place(x=180,y=400)

# it will be used for displaying the registration form onto the window
base.mainloop()
print("Registration form is created seccussfully...")
```

OUTPUT :



**BLDEA's
S.B.ARTS AND K.C.P. SCIENCE COLLEGE, VIJAYAPUR
DEPARTMENT OF COMPUTER SCIENCE**

Lab Manual

R Programming

Course Code: DSC 8-Lab

BSC-VI Semester

2023-24

PART A

1. Write a R program to find Area and Circumference of Circle
2. Write an R program Illustrate with if-else statement and how does it operate on vectors of variable length.
3. Write an R program Illustrate with for loop and stop on condition, to print the error message.
4. Write an R Program to find Factorial of given number.
5. Write an R Program to append a value to given empty vector .
6. Implementation of Vectors data objects operations.
7. Implementation of matrix, arrays and factors objects operations.
8. Write an R Program to Find Mean, Mode & Median.

PART B

1. Write an R Program to implement T-Test.
2. Write an R Program Compute mean values for vector aggregates defined by factors tapply and sapply.
3. Write an R Program to find Unique element of a given string and unique value from vector.
4. Write a R program to demonstrate Binomial Distribution.
5. Write a R program to demonstrate Normal Distribution.
6. Write an R Program Illustrate Reading & Writing Files.
7. Write a R program for simple bar plot for 5 subject marks
8. Implementation of Data visualization using ggplot.

Part A

Experiment 1:

#Write a R Program to find Area and Circumference of Circle

```
cat("Enter the radius")  
  
r=as.numeric(readLines("stdin",n=1))  
  
area=3.142*r*r  
  
circumference=2*3.142*r  
  
cat("Area of Circle=",area,"\n")  
  
cat("Circumference of Circle=",circumference,"\n")
```

Output :

```
bsc-cs@bsc-cs:~$ Rscript Ex1Area.r
```

```
Enter the radius 2.4
```

```
Area of Circle= 18.09792
```

```
Circumference of Circle= 15.0816
```

Experiment 2:

#Write an R program Illustrate with if-else statement and how does it operate on vectors of variable length

```
a = c(5,7,2,9)                # create a vector
res=ifelse(a %% 2 == 0,"even","odd") # check if each element in a is even or odd
print(res)
```

Using ifelse() statement with multiple conditions

```
a=50
b=55
val = ifelse(a >= 40 & b <= 60, 'TRUE', 'FALSE')
print(val)
```

With same vector length

```
x <- c(6,5,4,3,2)
y <- c(2,3,4,5,6)
res <- ifelse(x > y,x,y)
print(res)
```

With different vector length

```
x <- c(4)
y <- c(2,3,4,5,6)
res <- ifelse(x >= y,x,y)
print(res)
```

Output :

```
bsc-cs@bsc-cs:~$ Rscript Ex2Vector.r
```

```
[1] "odd" "odd" "even" "odd"
```

```
[1] "TRUE"
```

```
[1] 6 5 4 5 6
```

```
[1] 4 4 4 5 6
```

Experiment 3:

#Write an R Program to illustrate with for loop and stop on condition, to print the error message.

```
x = c(1, 2, 3, 4, 5, 6, 7)

for(i in x)                                # for loop with break statement
{

    if(i == 4)                             # if condition with break
    {
        print("Loop has terminated")
        break
    }

    print(i)
}
```

Output :

```
bsc-cs@bsc-cs:~$ Rscript Ex3Break.r
[1] 1
[1] 2
[1] 3
[1] "Loop has terminated"
```

Experiment 4:

#Write an R program to find Factorial of given number

```
cat("Enter a number: ")
```

```
num = as.integer(readLines("stdin",n=1))
```

```
factorial = 1
```

```
if(num < 0)                                # check is the number is negative or positive
```

```
{
```

```
  print("Not possible for negative numbers")
```

```
}
```

```
else if(num == 0)
```

```
{
```

```
  print("The factorial of 0 is 1")
```

```
}
```

```
else
```

```
{
```

```
  for(i in 1:num)
```

```
{
```

```
    factorial = factorial * i
```

```
}
```

```
  print(paste("The factorial of", num ,"is",factorial))
```

```
}
```

Output :

```
bsc-cs@bsc-cs:~$ Rscript Ex4Factorial.r
```

```
Enter a number: 5
```

```
[1] "The factorial of 5 is 120"
```

```
bsc-cs@bsc-cs:~$ Rscript Ex4Factorial.r
```

```
Enter a number: 12
```

```
[1] "The factorial of 12 is 479001600"
```

```
bsc-cs@bsc-cs:~$ Rscript Ex4Factorial.r
```

```
Enter a number: 20
```

```
[1] "The factorial of 20 is 2432902008176640000"
```

Experiment 5:**#Implementation of Vectors data objects operations.**

```
cat("Vector Operations")
vector_x <- c(1,2,21,56,82)
cat("\n accessing elements from vector\n")
vector_x[3]
vector_x[1:4]
cat("Operations on Vector\n")
cat("length of the vector",length(vector_x),"\n")
cat("Concatenation of vectors:\n")
combined_vector <- c(vector_x, 6:8)
cat(combined_vector)

cat("Element-wise operations:\n")
cat("Multiply each element in vector by 2:",vector_x * 2)
cat("\nAdd each element in vector with 10:",vector_x + 10)

cat("\nSum of all elements in a numeric vector:",sum(vector_x))
cat("\nMean of all elements in a numeric vector:",mean(vector_x))
cat("\nSorted vector:\n",sort(vector_x))
cat("\nAdding elements to vector:",c(vector_x, 6))
cat("\nReversing a vector: ",rev(vector_x))
cat("\nRemoving elements from a vector:",vector_x[-2])
y<- c(1, 2, 3,4,5)
cat("\nVectorized Addition=",result <- vector_x + y,"\n")
```


Experiment 6:**#Write an R Program to append a value to given empty vector****# Create an empty vector**

```
empty_vector=c( )
```

```
# Add a value to the empty vector
```

```
empty_vector<-append(empty_vector,42)
```

```
print(empty_vector)
```

Output :

```
bsc-cs@bsc-cs:~$ Rscript Ex6EmptyVector.r  
[1] 42
```

Experiment 7:**#Implementation of matrix, arrays and factors objects operations.****#a. Implementation of Matrix**

```
# Define a matrix
mat <- matrix(1:12, nrow = 3, ncol = 4)
print("Original Matrix:")
print(mat)

# Access elements of the matrix
print("Element at row 2, column 3:")
print(mat[2, 3])

# Modify an element of the matrix
mat[3, 2] <- 99
print("Modified Matrix:")
print(mat)

# Calculate the dimensions of the matrix
dim_mat <- dim(mat)
print("Dimensions of the matrix:")
print(dim_mat)

# Matrix multiplication
mat2 <- matrix(1:8, nrow = 4, ncol = 2)
product_mat <- mat %*% mat2
print("Matrix Multiplication:")
print(product_mat)

# Sum of all elements
total_sum <- sum(mat)
print("Sum of all elements in the matrix:")
print(total_sum)
```

Output :

```
bsc-cs@bsc-cs:~$ Rscript Ex7aMatrix.r
```

```
[1] "Original Matrix:"
```

```
  [,1] [,2] [,3] [,4]  
[1,]   1   4   7  10  
[2,]   2   5   8  11  
[3,]   3   6   9  12
```

```
[1] "Element at row 2, column 3:"
```

```
[1] 8
```

```
[1] "Modified Matrix:"
```

```
  [,1] [,2] [,3] [,4]  
[1,]   1   4   7  10  
[2,]   2   5   8  11  
[3,]   3  99   9  12
```

```
[1] "Dimensions of the matrix:"
```

```
[1] 3 4
```

```
[1] "Matrix Multiplication:"
```

```
  [,1] [,2]  
[1,]  70 158  
[2,]  80 184  
[3,] 276 768
```

```
[1] "Sum of all elements in the matrix:"
```

```
[1] 171
```

#B. Implementation of Arrays

```
# Create a 2x3 array
arr <- array(1:6, dim = c(2, 3))

# Print the array
print("Original Array:")
print(arr)

# Access elements of the array
print("Element at row 1, column 2:")
print(arr[1, 2]) # Output: 3

# Modify an element of the array
arr[2, 3] <- 99
print("Modified Array:")
print(arr)

# Calculate the dimensions of the array
dim_arr <- dim(arr)
print("Dimensions of the array:")
print(dim_arr) # Output: [1] 2 3

# Sum of all elements
total_sum <- sum(arr)
print("Sum of all elements in the array:")
print(total_sum)
```

Output :

```
bsc-cs@bsc-cs:~$ Rscript Ex7bArray.r
```

```
[1] "Original Array:"
```

```
  [,1] [,2] [,3]
```

```
[1,]   1   3   5
```

```
[2,]   2   4   6
```

```
[1] "Element at row 1, column 2:"
```

```
[1] 3
```

```
[1] "Modified Array:"
```

```
  [,1] [,2] [,3]
```

```
[1,]   1   3   5
```

```
[2,]   2   4  99
```

```
[1] "Dimensions of the array:"
```

```
[1] 2 3
```

```
[1] "Sum of all elements in the array:"
```

```
[1] 114
```

#C. Implementation of Factors

Creating a Vector:

- We create a vector `categories` containing categorical data.
- **Creating a Factor:**
- We convert the vector `categories` into a factor using the `factor()` function. This function automatically identifies unique values (levels) in the vector and assigns them integer codes.
- **Accessing Factor Levels:**
- We use `levels(factor_categories)` to print the unique levels present in the `factor_categories` factor.
- **Printing the Factor:**
- We print the factor itself using `print(factor_categories)`.
- **Frequency of Factor Levels:**
- We use `table(factor_categories)` to get the frequency count of each level in the factor.
- **Checking the Class:**
- We use `class(factor_categories)` to confirm that `factor_categories` is indeed a factor.

```
# Create a vector of categorical data
categories <- c("A", "B", "B", "C", "A", "C", "B")
```

```
# Create a factor from the vector
factor_categories <- factor(categories)
```

```
# Print the levels of the factor
print("Factor Levels:")
print(levels(factor_categories))
```

```
# Print the factor itself
print("Factor:")
print(factor_categories)
```

```
# Get the frequency of each level
print("Frequency of each level:")
print(table(factor_categories))
```

```
# Check the class of the factor
print("Class of the factor:")
print(class(factor_categories))
```

Output :

```
bsc-cs@bsc-cs:~$ Rscript Ex7Factor.r
Fatal error: cannot open file 'Ex7Factor.r': No such file or directory
```

```
bsc-cs@bsc-cs:~$ Rscript Ex7cFactor.r
[1] "Factor Levels:"
[1] "A" "B" "C"
[1] "Factor:"
[1] A B B C A C B
```

Levels: A B C

```
[1] "Frequency of each level:"
factor_categories
A B C
2 3 2
[1] "Class of the factor:"
[1] "factor"
```

#Experiment 8:**#Write an R Program to Find Mean, Mode & Median.**

```
# Create a numeric vector
data <- c(10, 15, 7, 20, 15, 12, 15, 18)

# Calculate Mean
mean_value <- mean(data)
cat("Mean:", mean_value, "\n")

# Calculate Mode

# Function to find mode
mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

mode_value <- mode(data)
cat("Mode:", mode_value, "\n")

# Calculate Median
median_value <- median(data)
cat("Median:", median_value, "\n")
```

Output :

```
bsc-cs@bsc-cs:~$ Rscript Ex8Mean.r  
Mean: 14  
Mode: 15  
Median: 15
```

PART – B

Experiment 1:**#Write an R Program to implement T-Test.**

```
# Example data
group1 <- c(22, 25, 28, 30, 32)
group2 <- c(18, 20, 25, 27, 29)

# Perform t-test
t_test_result <- t.test(group1, group2)

# Print the results
cat("T-Test Results:\n")
print(t_test_result)
```

Output :

```
bsc-cs@bsc-cs:~$ Rscript Ex1PBTTest.r
```

T-Test Results:

Welch Two Sample t-test

data: group1 and group2

t = 1.3145, df = 7.8068, p-value = 0.226

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-2.742567 9.942567

sample estimates:

mean of x mean of y

27.4 23.8

Experiment 2:

#Write an R Program Compute mean values for vector aggregates defined by factors tapply and sapply.

sapply() Function in R

The sapply() function in R is used to apply a function to each element of a vector, list, or data frame, and then attempts to simplify the result into a vector, matrix, or array if possible. Here's a breakdown of its syntax, parameters, and an example with its explanation.

Syntax

sapply(X, FUN, ...)

Parameters

- X: A vector, list, or data frame containing the data to be analyzed.
- FUN: The function to be applied to each element of X.

tapply() Function in R

The tapply() function in R is used to apply a function to subsets of a vector or array, grouped by one or more factors. It's especially useful for calculating summary statistics for different groups within your data.

Syntax

tapply(X, INDEX, FUN, ...)

Here, FUN is the function to be applied to each subset, X is the input vector or array, INDEX is a factor or list of factors denoting the subsets, and... are optional parameters that can be supplied to the function FUN.

```
scores_vector<-c(85, 90, 78, 92, 70, 88)
groups<-factor(c("FC","DF","FC","DF","FC","DF"))
cat("\n sapply function\n")
cat("\nThe given vectors are :\n")
print(scores_vector)
print(groups)
```

```
mean_values <- sapply(split(scores_vector, groups), mean)
cat("\nAverage scores by students in Distinction and First class using sapply
function:\n")
print(mean_values)
```

```
cat("\n tapply function\n")
cat("\nMean for scores of students according to their results using tapply function:\n
n ")
result <- tapply(scores_vector, groups, mean)
# Print the result
result
```

Output:

Rscript ExPB_Apply.r

sapply function

The given vectors are :

[1] 85 90 78 92 70 88

[1] FC DF FC DF FC DF

Levels: DF FC

Average scores by students in Distinction and First class using sapply function:

DF	FC
90.00000	77.66667

tapply function

Mean for scores of students according to their results using tapply function:

DF	FC
90.00000	77.66667

Experiment 3:

#Write an R Program to find Unique element of a given string and unique value from vector.

```
# Unique element for String Vector
str1 = "R programming is very simple"
cat("\nOriginal vector(string)\n")
print(str1)

# Convert string to character vector
char_vector <- strsplit(str1, "")[[1]]

# Find unique elements
unique_elements <- unique(char_vector)
cat("\nUnique elements of the given String vector:\n")
print(unique(unique_elements))

nums = c(11, 21, 21, 32, 24, 24, 5, 6)
cat("\n\nOriginal vector(number)\n")
print(nums)
cat("\nUnique elements of the given vector of numbers:\n")
print(unique(nums))
```

Output :

```
bsc-cs@bsc-cs:~$ Rscript ExPBUnique.r
```

```
Original vector(string)
```

```
[1] "R programming is very simple"
```

```
Unique elements of the given String vector:
```

```
[1] "R" " " "p" "r" "o" "g" "a" "m" "i" "n" "s" "v" "e" "y" "l"
```

```
Original vector(number)
```

```
[1] 11 21 21 32 24 24 5 6
```

```
Unique elements of the given vector of numbers:
```

```
[1] 11 21 32 24 5 6
```

Experiment 4:**#Write a R program to demonstrate Binomial Distribution.**

The binomial distribution, on the other hand, describes the probability of obtaining a certain number of successes in a fixed number of independent Bernoulli trials. It is a discrete probability distribution that has two parameters – the number of trials, denoted by n , and the probability of success on each trial, denoted by p .

Binomial distribution

In R programming, you can use the built-in `dbinom()`, `pbinom()`, `qbinom()`, and `rbinom()` functions to work with the binomial distribution.

Here's a brief explanation of each function:

1. **Parameters:** n is the number of trials (e.g., flipping a coin 10 times), and p is the probability of success on each trial (e.g., probability of getting heads on each flip).
2. **Generating Random Numbers:** `rbinom(1000, n, p)` generates 1000 random numbers following a binomial distribution with parameters n and p .
3. **Histogram Plot:** `hist()` function is used to plot a histogram of the generated random numbers to visualize the distribution.
4. **Theoretical Probabilities:** `dbinom(x, n, p)` calculates the probability mass function of the binomial distribution for each possible number of successes x from 0 to n .
5. **Output:** The program prints the theoretical probabilities for each number of successes from 0 to n .

```
# Parameters
n <- 10                # Number of trials
p <- 0.5               # Probability of success

# Generate random numbers from binomial distribution
random_numbers <- rbinom(1000, n, p)

# Plot histogram of the generated random numbers
hist(random_numbers, breaks = seq(0, n+1, by = 1),
      main = "Histogram of Binomial Distribution",
      xlab = "Number of Successes",
      ylab = "Frequency")

# Calculate and print the theoretical probabilities
x <- 0:n # Possible number of successes
probabilities <- dbinom(x, n, p)
cat("Theoretical Probabilities:\n")
for (i in 1:length(x)) {
  cat("P(X =", x[i], ") =", probabilities[i], "\n")
}
```

Output :

```
bsc-cs@bsc-cs:~$ Rscript Ex_PBBinomial.r
```

Theoretical Probabilities:

$P(X = 0) = 0.0009765625$

$P(X = 1) = 0.009765625$

$P(X = 2) = 0.04394531$

$P(X = 3) = 0.1171875$

$P(X = 4) = 0.2050781$

$P(X = 5) = 0.2460938$

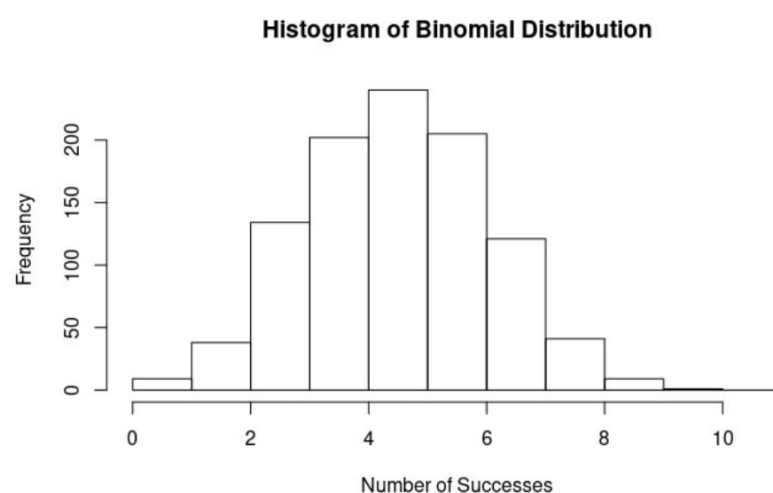
$P(X = 6) = 0.2050781$

$P(X = 7) = 0.1171875$

$P(X = 8) = 0.04394531$

$P(X = 9) = 0.009765625$

$P(X = 10) = 0.0009765625$



Experiment 5 :**#Write a R program to demonstrate Normal Distribution.****The normal distribution**

The normal distribution, also known as the Gaussian distribution, is a continuous probability distribution characterized by its bell-shaped curve. In R, you can generate random numbers from a normal distribution and visualize its density using the `dnorm()` function for the probability density function (PDF) and `rnorm()` function for generating random numbers.

Explanation:

1. Parameters: mean and sd represent the mean and standard deviation of the normal distribution, respectively.
2. Generating Random Numbers: `rnorm(1000, mean, sd)` generates 1000 random numbers following a normal distribution with parameters mean and sd.
3. Histogram Plot: `hist()` function is used to plot a histogram of the generated random numbers with `freq = FALSE` to display the density instead of frequencies.
4. Theoretical Density Curve: `curve(dnorm(x, mean, sd), add = TRUE, col = "blue", lwd = 2)` overlays the theoretical density curve of the normal distribution on top of the histogram.
5. Legend: `legend()` function is used to add a legend to the plot to differentiate between the histogram and the density curve.

```
mean <- 0 # Mean of the distribution
sd <- 1 # Standard deviation of the distribution

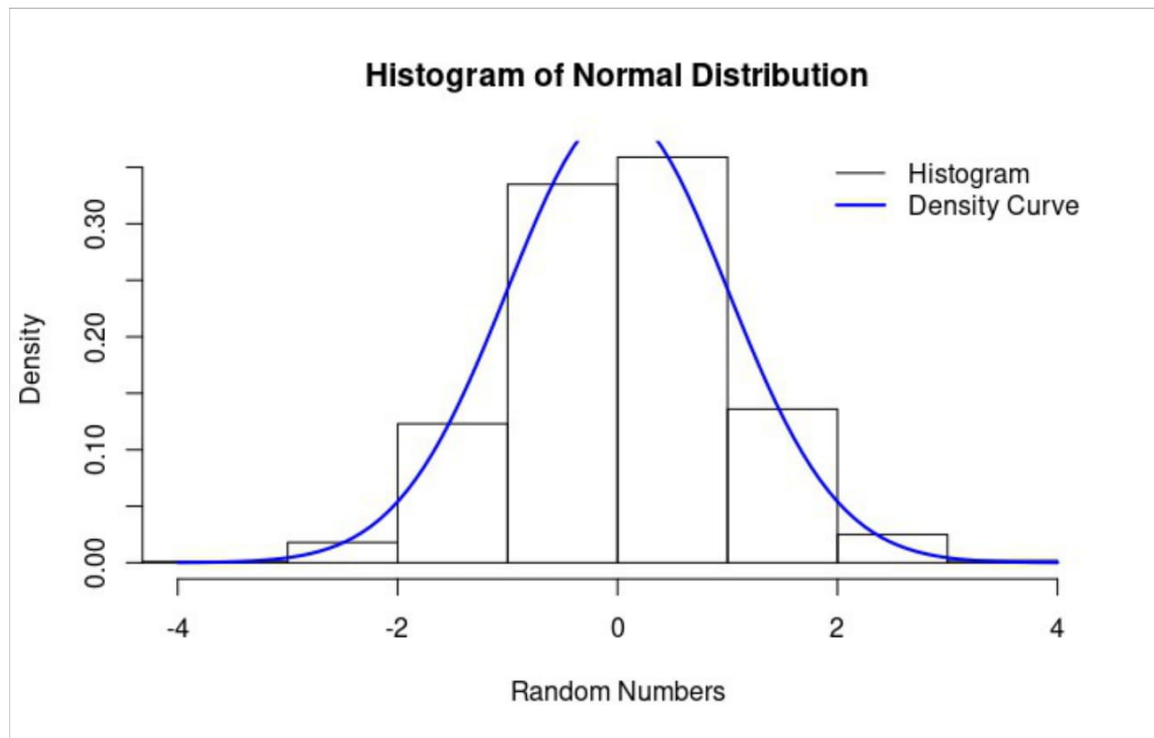
# Generate random numbers from normal distribution
random_numbers <- rnorm(1000, mean, sd)

# Plot histogram of the generated random numbers
hist(random_numbers, freq = FALSE,
      main = "Histogram of Normal Distribution",
      xlab = "Random Numbers",
      ylab = "Density",
      xlim = c(-4, 4)) # Adjust xlim to zoom into the range

# Overlay the theoretical density curve
curve(dnorm(x, mean, sd), add = TRUE, col = "blue", lwd = 2)

# Add a legend
legend("topright", legend = c("Histogram", "Density Curve"),
      col = c("black", "blue"), lwd = c(1, 2), bty = "n")
```

Output :



Experiment 6:**Write an R Program Illustrate Reading & Writing Files.**

Reading and writing files are fundamental operations in data analysis and processing. In R, you can handle various types of files, such as text files, CSV files, Excel files, and more. Here's a basic example to illustrate how to read from and write to a text file in R:

Reading a Text File

Create a text file named `sample.txt` and save it in your working directory with the following content:

Hello, This is a text file.

This file is used to demonstrate extracting the lines.

```
# Assign the file path
file_path <- "sample.txt"

# Read the entire content of the file into a character vector
file_content <- readLines(file_path)

# Print the content of the file
cat("\n\nExtracted Contents of an existing text file:sample.txt", "\n")
cat(file_content, sep = "\n")

# Content to write to the file
content_to_write <- c("In R, you can handle various types of files",
                      "such as text files, CSV files, Excel files, and more",
                      "this program illustrate how to read from and write to a text file in R.")

# File path for the new file to be created
new_file <- "sampleNew.txt";

# Write content to the file
writeLines(content_to_write, new_file)

# Print a message indicating success
cat("\n\nContent has been written to", new_file, "\n")

# Optionally, read and print the content of the newly created file
cat("Contents of the new file:", "\n")
new_file_content <- readLines(new_file)
cat(new_file_content, sep = "\n")
```

Output :

```
bsc-cs@bsc-cs:~$ Rscript Ex_PBFileRW.r
```

Extracted Contents of an existing text file:sample.txt

Hello, This is a text file.

illustrate how to read from and write to a text file in R

Content has been written to sampleNew.txt

Contents of the new file:

In R, you can handle various types of files

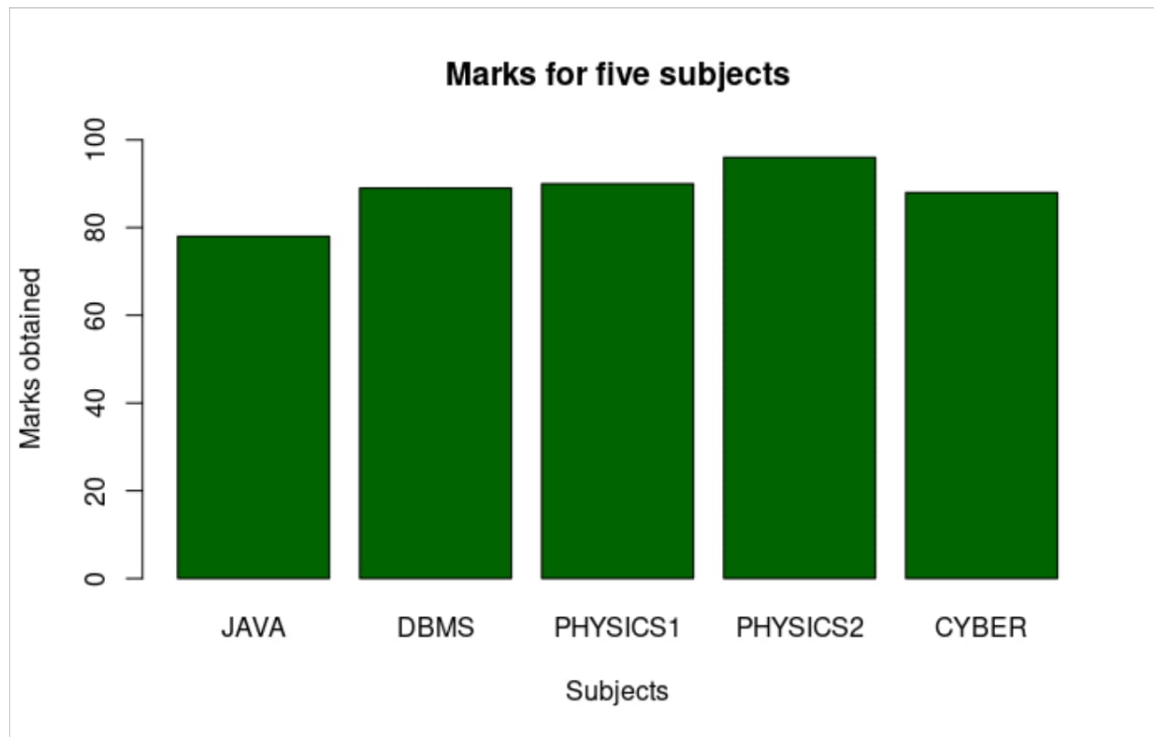
such as text files, CSV files, Excel files, and more

this program illustrate how to read from and write to a text file in R.

Experiment 7:**Write a R program for simple bar plot for 5 subject marks**

```
marks=c(78,89,90,96,88)
subjects=c("JAVA","DBMS","PHYSICS1","PHYSICS2","CYBER")
barplot(marks,
        names.arg=subjects,          # Names of the subjects on x-axis
        xlab="Subjects",             # Label for x-axis
        ylab="Marks obtained",       # Label for y-axis
        main="Marks for five subjects", # Title of the plot
        col="darkgreen",             # Color of the bars
        border="black",              # Border color of the bars
        horiz=FALSE,
        ylim = c(0, 100)             # Set y-axis limits
    )
```

Output :



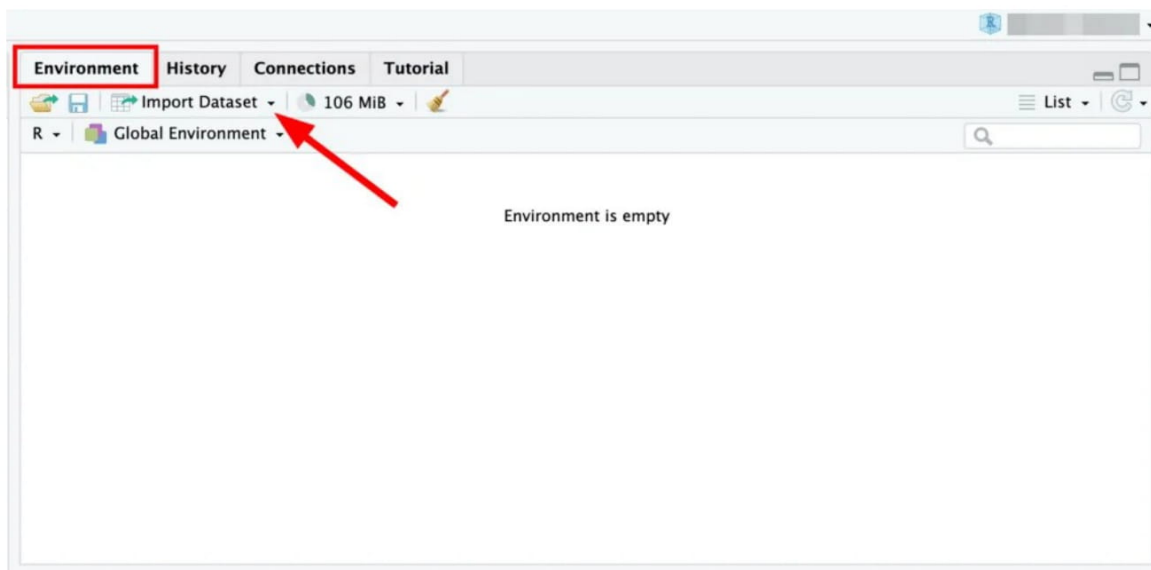
Experiment 8 : Implementation of Data visualization using ggplot.

1. Create the following Crop Production dataset that you can use to create a visualization using ggplot2 in R. This dataset contains fictional data about the production of different crops (in tons) in different regions:

Crop_Production Table

Region	Crop	Acreage	Production
North	Wheat	1000	5000
North	Maize	1500	6000
North	Rice	800	4000
South	Wheat	1200	4800
South	Maize	1000	5000
South	Rice	600	3000
East	Wheat	800	4000
East	Maize	1300	5200
East	Rice	900	4500
West	Wheat	1100	4400
West	Maize	900	3600
West	Rice	700	2800

2. Create this fill using Excel and Save this file .CSV extension. Save file with name "Crop_Production.CSV"
3. Then import this dataset in Rstudio using Go to the Environment pane in R Studio, typically on the upper right-hand side. You will see an **Import Dataset** dropdown menu – as shown in the image below. Click on it.



4. In the dropdown menu, select the **From Excel...** option. This will open a file explorer.



5. On the *Import Excel Data* window, click the **Browse** button to navigate to your Excel file in the file explorer, select it, and click **Open**.

6. Then click on import button.

```
library(ggplot2)

# Import dataset
crop_data <- read.csv("~/Documents/Crop_Production.csv")

# Check structure of the dataset
str(crop_data)

# Summary statistics
summary(crop_data)

# Display first few rows
head(crop_data)

# Bar plot of crop production by region
ggplot(crop_data, aes(x = Region, y = Production, fill = Crop)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Crop Production by Region",
       x = "Region", y = "Production (tons)",
       fill = "Crop") +
```

```
theme_minimal()
```

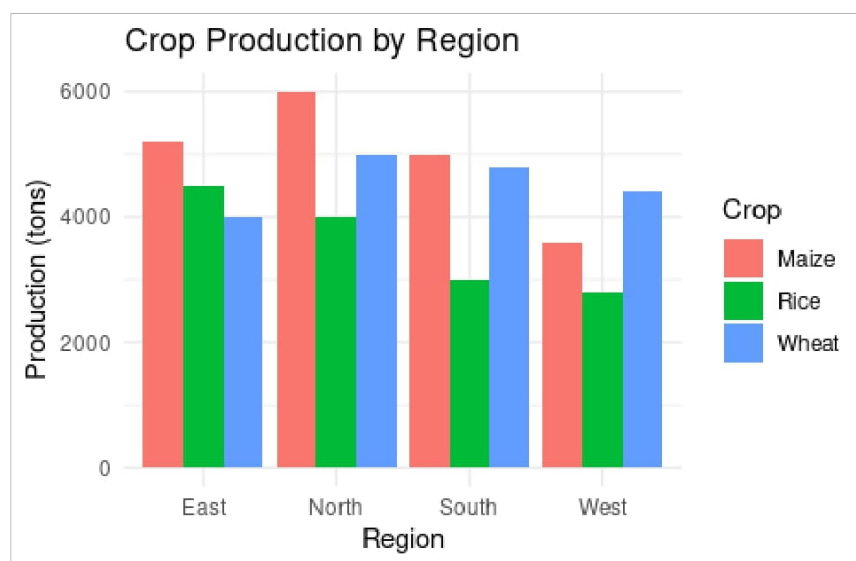
```
# Save the plot as a PNG file
```

```
ggsave("crop_production_plot.png", plot = last_plot(), width = 10, height = 6)
```

Output :

Region Crop Acreage Production

1	North	Wheat	1000	5000
2	North	Maize	1500	6000
3	North	Rice	800	4000
4	South	Wheat	1200	4800
5	South	Maize	1000	5000
6	South	Rice	600	3000



**BLDEA's
S.B.ARTS AND K.C.P. SCIENCE COLLEGE, VIJAYAPUR
DEPARTMENT OF COMPUTER SCIENCE**

Lab Manual

Web Programming

Course Code:21BSC6CSMJ1P

BSC-VI Semester

2023-24

Part - A

1. Design web pages for your college containing college name and Logo, departments list using href, list tags.

```
<html>
<head>
<title>S.B.Arts and K.C.P.Science College</title>

</head>

<body>


<h6><center> B.L.D.E. Association's</center></h6>
<h3><center>S.B.Arts and K.C.P.Science College, Vijayapur</center></h3>

<h4><b>Departments</b></h4>
<ol name="Departments" align=center><b>
<li>Physics</li>
<li>Mathematics</li>
<li><a href="exp1_1.html">Computer Science</a></li>

<li>Botany</li>
<li>Zoology</li>
<li>Chemistry</li>
<li>BCA</li>
<li>MSc in Physics</li>
<li>MSc in Chemistry</li>
<li>MA in English</li>
</b>
</ol>
</body>
</html>
```



BSC CS V & VI SEM.pdf ×

S.B.Arts and K.C.P.Scienc ×



file:///E:/BSC-cs/CS_DeptFiles/LabManuals/WebPr ☆



B.L.D.E. Association's

S.B.Arts and K.C.P.Science College, Vijayapur

Departments

1. Physics
2. Mathematics
3. Computer Science
4. Botany
5. Zoology
6. Chemistry
7. BCA
8. MSc in Physics
9. MSc in Chemistry
10. MA in English

Exp2.Create a class timetable using table tag.

```
<!DOCTYPE html>
<html>

<head>
    <title>Time Table</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f8f9fa;
            margin: 0;
            padding: 0;
        }

        h1 {
            text-align: center;
            color: #343a40;
        }

        table {
            border-collapse: collapse;
            margin: 20px auto;
            background-color: #fff;
            border: 2px solid #dee2e6;
            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
        }

        th,
        td {
            border: 1px solid #dee2e6;
            padding: 10px;
            text-align: center;
        }

        th {
            background-color: #f2f2f2;
            color: #343a40;
        }

        .highlight {
            background-color: #f8f9fa;
        }

        .special {
            background-color: #f0f0f0;
        }
    </style>
</head>
```

<body>

<h1>TIME TABLE</h1>

<table>

<tr>

<th>Day/Period</th>

<th>I
9:30-10:20</th>

<th>II
10:20-11:10</th>

<th>III
11:10-12:00</th>

<th>12:00-12:40</th>

<th>IV
12:40-1:30</th>

<th>V
1:30-2:20</th>

<th>VI
2:20-3:10</th>

<th>VII
3:10-4:00</th>

</tr>

<tr>

<td class="highlight">Monday</td>

<td>Eng</td>

<td>Mat</td>

<td>Che</td>

<td rowspan="6" class="special">LUNCH</td>

<td colspan="3" class="special">LAB</td>

<td>Phy</td>

</tr>

<tr>

<td class="highlight">Tuesday</td>

<td colspan="3" class="special">LAB</td>

<td>Eng</td>

<td>Che</td>

<td>Mat</td>

<td class="special">SPORTS</td>

</tr>

<tr>

<td class="highlight">Wednesday</td>

<td>Mat</td>

<td>Phy</td>

<td>Eng</td>

<td>Che</td>

<td colspan="3">LIBRARY</td>

</tr>

<tr>

<td class="highlight">Thursday</td>

<td>Phy</td>

<td>Eng</td>

<td>Che</td>

<td colspan="3" class="special">LAB</td>

<td>Mat</td>

</tr>

<tr>

<td class="highlight">Friday</td>

```
        <td colspan="3" class="special">LAB</td>
        <td>Mat</td>
        <td>Che</td>
        <td>Eng</td>
        <td>Phy</td>
    </tr>
    <tr>
        <td class="highlight"><b>Saturday</b></td>
        <td>Eng</td>
        <td>Che</td>
        <td>Mat</td>
        <td colspan="3">SEMINAR</td>
        <td class="special">SPORTS</td>
    </tr>
</table>
</body>

</html>
```

TIME TABLE

Day/Period	I 9:30-10:20	II 10:20-11:10	III 11:10-12:00	12:00-12:40	IV 12:40-1:30	V 1:30-2:20	VI 2:20-3:10	VII 3:10-4:00	
Monday	Eng	Mat	Che	LUNCH	LAB			Phy	
Tuesday	LAB				Eng	Che	Mat	SPORTS	
Wednesday	Mat	Phy	Eng		Che	LIBRARY			
Thursday	Phy	Eng	Che		LAB			Mat	
Friday	LAB				Mat	Che	Eng	Phy	
Saturday	Eng	Che	Mat		SEMINAR			SPORTS	

Exp3. Write a HTML code to design Student registrations form for your college Admission

```
<html>
<head>
<title>Student Registration Form</title>
</head>

<body>
<h3>STUDENT REGISTRATION FORM</h3>

<table align="center" cellpadding = "10">

<tr>
<td>FIRST NAME</td>
<td><input type="text" name="First_Name" maxlength="30"/>
(max 30 characters a-z and A-Z)
</td>
</tr>

<tr>
<td>LAST NAME</td>
<td><input type="text" name="Last_Name" maxlength="30"/>
(max 30 characters a-z and A-Z)
</td>
</tr>

<tr>
<td>DATE OF BIRTH</td>
<td>
<select name="Birthday_day" id="Birthday_Day">
<option value="-1">Day:</option>
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>

<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>
<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>

<option value="13">13</option>
<option value="14">14</option>
<option value="15">15</option>
<option value="16">16</option>
<option value="17">17</option>
<option value="18">18</option>
<option value="19">19</option>
</td>
</tr>
</table>
</body>
</html>
```

```
<option value="20">20</option>
<option value="21">21</option>
```

```
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
```

```
<option value="31">31</option>
</select>
```

```
<select id="Birthday_Month" name="Birthday_Month">
<option value="-1">Month:</option>
<option value="January">Jan</option>
<option value="February">Feb</option>
<option value="March">Mar</option>
<option value="April">Apr</option>
<option value="May">May</option>
<option value="June">Jun</option>
<option value="July">Jul</option>
<option value="August">Aug</option>
<option value="September">Sep</option>
<option value="October">Oct</option>
<option value="November">Nov</option>
<option value="December">Dec</option>
</select>
```

```
<select name="Birthday_Year" id="Birthday_Year">
```

```
<option value="-1">Year:</option>
<option value="2009">2009</option>
<option value="2008">2008</option>
<option value="2007">2007</option>
<option value="2006">2006</option>
<option value="2005">2005</option>
```

```
</select>
</td>
</tr>
```

```
<tr>
<td>EMAIL ID</td>
<td><input type="text" name="Email_Id" maxlength="100" /></td>
</tr>
```

```
<tr>
<td>MOBILE NUMBER</td>
<td>
<input type="text" name="Mobile_Number" maxlength="10" />
```

(10 digit number)

</td>

</tr>

<tr>

<td>GENDER</td>

<td>

Male <input type="radio" name="Gender" value="Male" />

Female <input type="radio" name="Gender" value="Female" />

</td>

</tr>

<tr>

<td>ADDRESS

</td>

<td><textarea name="Address" rows="4" cols="30"></textarea></td>

</tr>

<tr>

<td>CITY</td>

<td><input type="text" name="City" maxlength="30" />

(max 30 characters a-z and A-Z)

</td>

</tr>

<tr>

<td>PIN CODE</td>

<td><input type="text" name="Pin_Code" maxlength="6" />

(6 digit number)

</td>

</tr>

<tr>

<td>STATE</td>

<td><input type="text" name="State" maxlength="30" />

(max 30 characters a-z and A-Z)

</td>

</tr>

<tr>

<td>COUNTRY</td>

<td><input type="text" name="Country" value="India" readonly="readonly" /></td>

</tr>

<tr>

<td>QUALIFICATION

</td>

<td>

<table>

<tr>

<td align="center">Sl.No.</td>

<td align="center">Examination</td>

<td align="center">Board</td>

<td align="center">Percentage</td>

```
<td align="center"><b>Year of Passing</b></td>
</tr>
```

```
<tr>
<td>1</td>
<td>Class X</td>
<td><input type="text" name="ClassX_Board" maxlength="30" /></td>
<td><input type="text" name="ClassX_Percentage" maxlength="30" /></td>
<td><input type="text" name="ClassX_YrOfPassing" maxlength="30" /></td>
</tr>
```

```
<tr>
<td>2</td>
<td>Class XII</td>
<td><input type="text" name="ClassXII_Board" maxlength="30" /></td>
<td><input type="text" name="ClassXII_Percentage" maxlength="30" /></td>
<td><input type="text" name="ClassXII_YrOfPassing" maxlength="30" /></td>
</tr>
```

```
<tr>
<td></td>
<td></td>
<td align="center">(10 char max)</td>
<td align="center">(upto 2 decimal)</td>
</tr>
</table>
```

```
</td>
</tr>
```

```
<tr>
<td>COURSES<br />APPLIED FOR</td>
<td>
BCA
<input type="radio" name="Course_BCA" value="BCA">
B.Com
<input type="radio" name="Course_BCom" value="B.Com">
B.Sc
<input type="radio" name="Course_BSc" value="B.Sc">
B.A
<input type="radio" name="Course_BA" value="B.A">
</td>
</tr>
```

```
<td colspan="2" align="center">
<input type="submit" value="Submit">
<input type="reset" value="Reset">
</td>
</tr>
</table> </form></body> </html>
```

STUDENT REGISTRATION FORM

FIRST NAME

Amruta

(max 30 characters a-z and A-Z)

LAST NAME

Patil

(max 30 characters a-z and A-Z)

DATE OF BIRTH

2

Mar

2007

EMAIL ID

amrutapatil@gmail.com

MOBILE NUMBER

9111114466

(10 digit number)

GENDER

Male

☐

Female

☒

ADDRESS

Adarsh Nagar,
Vijayapur

CITY

Vijayapur

(max 30 characters a-z and A-Z)

PIN CODE

586101

(6 digit number)

STATE

Karnataka

(max 30 characters a-z and A-Z)

COUNTRY

India

QUALIFICATION

SLNo.	Examination	Board	Percentage	Year of Passing
1	Class X	SSLC	90	2020
2	Class XII	PU Board	80	2022

(10 char max)

(upto 2 decimal)

COURSES APPLIED FOR

BCA

☐

B.Com

☒

B.Sc

☐

B.A

☐

Submit

Reset

Exp4. Design Web Pages with includes Multi-Media data (Image, Audio, Video, GIFs etc)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Multi-Media Web Page</title>
  <style>
    /* Add your CSS styles here */
    .container {
      max-width: 800px;
      margin: 0 auto;
      padding: 20px;
    }
    img {
      max-width: 100%;
      height: auto;
      margin-bottom: 20px;
    }
    video {
      max-width: 100%;
      height: auto;
      margin-bottom: 20px;
    }
    audio {
      margin-bottom: 20px;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Multi-Media Web Page</h1>

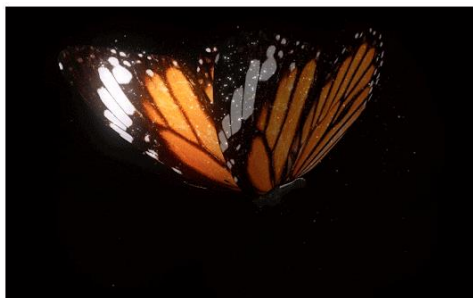
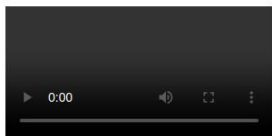
    <!-- Image -->
    

    <!-- Audio -->
    <audio controls>
      <source src="song1.mpeg" type="audio/mpeg">
      Your browser does not support the audio element.
    </audio>

    <!-- Video -->
    <video controls>
      <source src="video1.mp4" type="video/mp4">
      Your browser does not support the video element.
    </video>

    <!-- GIF -->
    

  </div> </body> </html>
```



Exp5. Create a web page using frame.

<!--File Name: Exp5.html-->

<!DOCTYPE html>

<html>

<frame>

<frameset cols="50%,50%">

<frame src="exp1.html">

<frame src="career.html">

</frameset>

</frame>

</html>

<!--File Name: career.html-->

<!DOCTYPE html>

<html>

<head>

<title>BSc Computer Science</title>

</head>

<body>

<h1 align="center"> Career Opportunities for BSc Students </h1>

<h2 align="center"> In IT Sectors </h2>

<h3>

<ol type="a">

System Integrator

Web Designer

Database Administrator

Computer Programmer

System Analyst

Games Developer

Application Analyst

Games Developer

IT Consultant

</h3>

<h2 align="center"> In Government Sectors </h2>

<h3>

<ul type="circle" >

Banking Sector

SAIL

BHEL

AAI

FCI

NTPC

HAL

RRB

ECIL

DRDO

 </h3> </body> </html>




B.L.D.E. Association's

S.B.Arts and K.C.P.Science College,
Vijayapur

Departments

1. Physics
2. Mathematics
3. Computer Science
4. Botany
5. Zoology
6. Chemistry
7. BCA
8. MSc in Physics
9. MSc in Chemistry
10. MA in English

Career Opportunities for BSc Students

In IT Sectors

- a. System Integrator
- b. Web Designer
- c. Database Administrator
- d. Computer Programmer
- e. System Analyst
- f. Games Developer
- g. Application Analyst
- h. Games Developer
- i. IT Consultant

In Government Sectors

- o Banking Sector
- o SAIL
- o BHEL
- o AAI
- o FCI
- o NTPC
- o HAL
- o RRB
- o ECIL
- o DRDO

Exp6. Write code in HTML to develop a webpage having two frames that divide the webpage into two equal rows and then divide the first row into equal columns fill each frame with a different background colour.

```
<!DOCTYPE HTML>
<html>
  <head>
  </head>
  <frameset rows="50%,50%">
    <frameset cols="50%,50%">
      <frame src="" style="background:red"></frame>
      <frame src="" style="background:green"></frame>
    </frameset>
    <frame src="" style="background:orange"></frame>
  </frameset>
</html>
```



Exp 8. Using HTML, CSS create display a text called –Hello India !|| on top of an image of India- Map using an overlay

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.container {
  position: relative;
  width: 50%;
}

.image {
  opacity: 1;
  display: block;
  width: 100%;
  height: auto;
  transition: .5s ease;
  backface-visibility: hidden;
}

.middle {
  transition: .5s ease;
  opacity: 0;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  -ms-transform: translate(-50%, -50%);
  text-align: center;
}

.container:hover .image {
  opacity: 0.3;
}

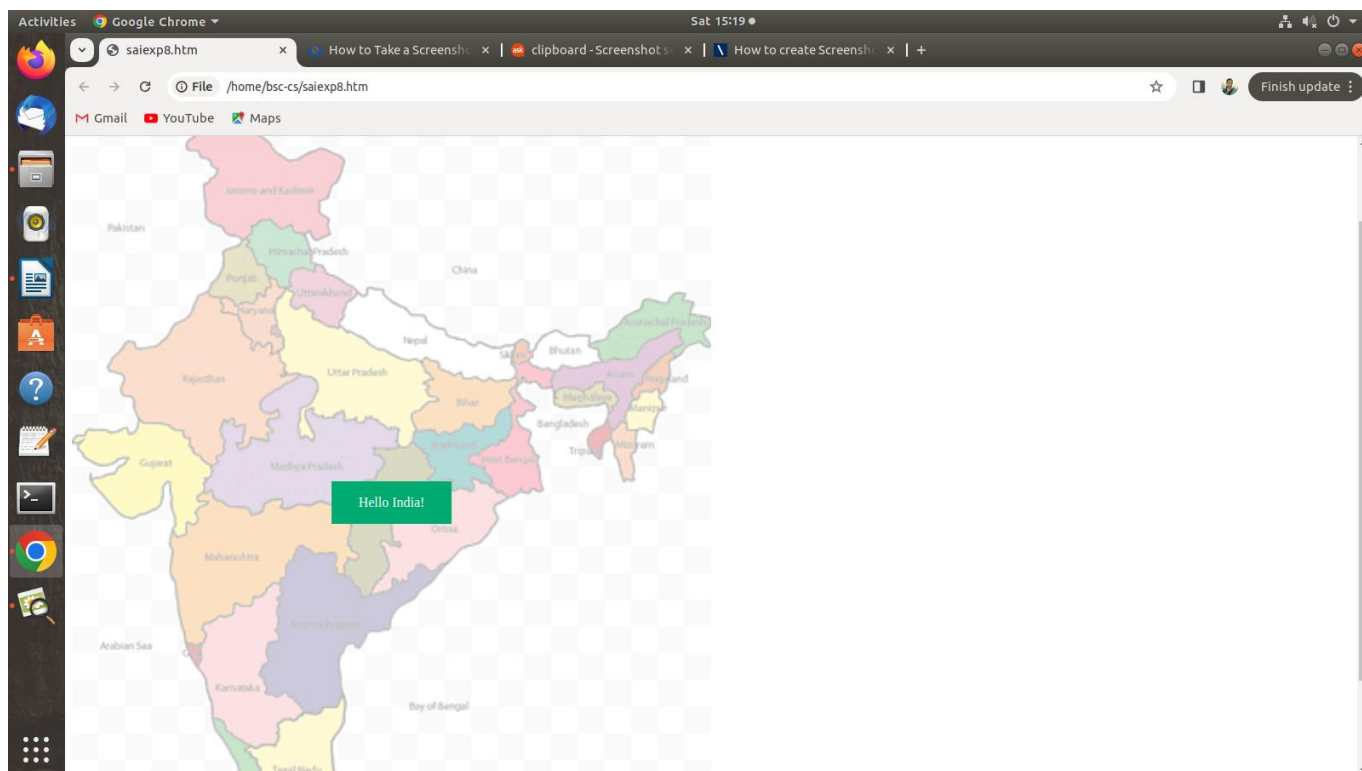
.container:hover .middle {
  opacity: 1;
}

.text {
  background-color: #04AA6D;
  color: white;
  font-size: 16px;
  padding: 16px 32px;
}
</style>
</head>
<body>

<h2>India Map</h2>
<p>Hover over the image to see the effect.</p>
```

```
<div class="container">
  
  <div class="middle">
    <div class="text">Hello India!</div>
  </div>
</div>

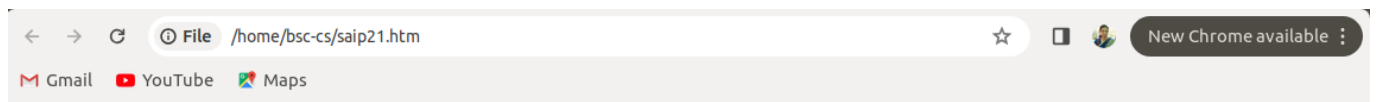
</body></html>
```



Part B

1. JavaScript Program to perform Basic Arithmetic operations.

```
<html>
<head>
<title>Arithmetic Operations</title>
</head>
<body>
<script type="text/javascript">
    var a = 12;
    var b = 34;
    var result; document.write("Value of a = " + a + " and b = "+ b); result = a + b;
    document.write("<br>Addition of a & b = " + result ); result = a - b;
    document.write("<br>Subtraction of a & b = " + result ); result = a * b;
    document.write("<br>Multiplication of a & b = " + result ); result = a / b;
    document.write("<br>Division of a & b = " + result );
</script>
</body>
</html>
```



Value of a = 12 and b = 34
Addition of a & b = 46
Subtraction of a & b = -22
Multiplication of a & b = 408
Division of a & b = 0.35294117647058826



2. JavaScript Program to implement all string operations.

```
// Define a sample string for demonstration
let sampleString = "Hello, World!";

// 1. Length of the string
console.log("Length of the string:", sampleString.length);

// 2. Convert to uppercase
console.log("Uppercase:", sampleString.toUpperCase());

// 3. Convert to lowercase
console.log("Lowercase:", sampleString.toLowerCase());

// 4. Extract substring from index 2 to 7 (exclusive)
console.log("Substring (2-7):", sampleString.substring(2, 7));

// 5. Extract substring from index 2 to the end
console.log("Substring (2 to end):", sampleString.substring(2));

// 6. Replace "Hello" with "Hi"
console.log("Replace 'Hello' with 'Hi':", sampleString.replace("Hello", "Hi"));

// 7. Concatenate two strings
let anotherString = " Have a nice day!";
console.log("Concatenated string:", sampleString.concat(anotherString));

// 8. Find the index of 'World'
console.log("Index of 'World':", sampleString.indexOf("World"));

// 9. Split the string by comma
console.log("Split by comma:", sampleString.split(","));

// 10. Trim leading and trailing whitespace
let stringWithWhitespace = " Hello, World! ";
console.log("Trimmed string:", stringWithWhitespace.trim());
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ node exp2_PB.js
```

Length of the string: 13

Uppercase: HELLO, WORLD!

Lowercase: hello, world!

Substring (2-7): llo,

Substring (2 to end): llo, World!

Replace 'Hello' with 'Hi': Hi, World!

Concatenated string: Hello, World! Have a nice day!

Index of 'World': 7

Split by comma: ['Hello', ' World!']

Trimmed string: Hello, World!

3. JavaScript Program to Check Prime Number

```
// Function to check if a number is prime
function isPrime(num) {
  if (num <= 1)
  {
    return false;
  }
  else
  {
    for (let i = 2; i <= Math.sqrt(num); i++)
    {
      if (num % i === 0) {
        return false;
      }
    }
    return true;
  }
}

// Read a number from the user
let number=prompt("Enter a number:");

// Convert the input string to an integer
number = parseInt(number);

// Check if the number is prime using the isPrime function
if (isPrime(number)) {
  console.log(number + " is a prime number.");
} else {
  console.log(number + " is not a prime number.");
}
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ node exp3_PB.js
```

```
Enter a number:12
```

```
12 is not a prime number.
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ node exp3_PB.js
```

```
Enter a number:73
```

```
73 is a prime number.
```

Exp 4. JavaScript Program to implement Java script Object Concept

```
// Define an object representing a car
let car = {
  brand: "Toyota",
  model: "Camry",
  year: 2022,
  color: "blue",
  // Method to display information about the car
  displayInfo: function() {
    console.log("Car Information:");
    console.log("Brand: " + this.brand);
    console.log("Model: " + this.model);
    console.log("Year: " + this.year);
    console.log("Color: " + this.color);
  }
};

// Accessing properties of the car object
console.log("Brand of the car:", car.brand);
console.log("Model of the car:", car.model);

// Accessing and modifying properties
car.year = 2023;
console.log("Updated year of the car:", car.year);

// Adding a new property
car.price = 25000;
console.log("Price of the car:", car.price);

// Calling the displayInfo method to display car information
car.displayInfo();
```

```
bsc-cs@bsccs-OptiPlex-3070:~$ node exp4_PB.js
```

Brand of the car: Toyota

Model of the car: Camry

Updated year of the car: 2023

Price of the car: 25000

Car Information:

Brand: Toyota

Model: Camry

Year: 2023

Color: blue

Exp5. JavaScript Program to Create Array and inserting Data into Array

```
// Create an empty array
let myArray = [ ];

// Read the number of elements in the array from the user
let numElements = parseInt(prompt("Enter the number of elements in the array:"));

// Read array elements from the user
for (let i = 0; i < numElements; i++) {
    let element = parseInt(prompt("Enter element " + (i + 1) + ":"));
    myArray.push(element);
}

// Display the array contents
console.log("Array contents:", myArray);
```

Enter the number of elements in the array:5

Enter element 1:45

Enter element 2:75

Enter element 3:62

Enter element 4:11

Enter element 5:22

Array contents: [45, 75, 62, 11, 22]

Exp 6. JavaScript Program to Validate an Email Address.

```
// Function to validate an email address
function validateEmail(email) {
    // Regular expression for validating email addresses
    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

    // Test the email against the regular expression
    return emailRegex.test(email);
}

// Read email address from the user
let userEmail = prompt("Enter your email address:");

// Validate the email address
if (validateEmail(userEmail)) {
    console.log(userEmail + " is a valid email address.");
} else {
    console.log(userEmail + " is not a valid email address.");
}
```

Enter your email address:rhrajeshwari@gmail.com
rhrajeshwari@gmail.com is a valid email address.

Enter your email address:neha123@
neha123@ is not a valid email address.