

18CSL01

Problem solving with programming using C

Lab Manual



BAPATLA ENGINEERING COLLEGE :: BAPATLA (AUTONOMOUS)

I Year B-Tech Students
For the Academic Year 2018 - '19

Prepared by

Mr. SK.Mabasha, Asst. Professor

Department of Information Technology



Bapatla Engineering College :: Bapatla
(Autonomous)
Department of Information Technology
CP LAB MANUAL

Class: **Iyr.B.Tech**

(Semester-I)

Sub& Code: **CP LAB (18CSL01)**

Faculty Name: **SK.MABASHA**

List of Experiments

Ex. No.	Name of the Experiment	Page No.
1	A program for electricity bill taking different categories of users, different slabs in each category. (Using nested if else statement).	1
2	Write a C program to evaluate the following (using loops): (a) $1 + x^2/2! + x^4/4! + \dots$ up to ten terms (b) $x + x^3/3! + x^5/5! + \dots$ up to 7 digit accuracy	3
3	Write a C program to check whether the given number is (a) Prime or not. (b) Perfect or Abundant or Deficient.	5
4	Write a C program to display statistical parameters (using one – dimensional array). (a) Mean (b) Mode (c) Median (d) Variance.	7
5	Write a C program to read a list of numbers and perform the following operations (a) Print the list. (b) Delete duplicates from the list. (c) Reverse the list.	9
6	Write a C program to read a list of numbers and search for a given number using Binary search algorithm and if found display its index otherwise display the message “Element not found in the List”.	11
7	Write a C program to read two matrices and compute their sum and product.	13
8	A menu driven program with options (using array of character pointers). (a) To insert a student name (b) To delete a student name (c) To print the names of students	17
9	Write a C program to read list of student names and perform the following operations (a) To print the list of names. (b) To sort them in ascending order. (c) To print the list after sorting.	20
10	Write a C program that consists of recursive functions to (a) Find factorial of a given number (b) Solve towers of Hanoi with three towers (A, B & C) and three disks initially on tower A.	22

REFERENCES:

1. Ashok N.Kamthane, “Programming in C”, PEARSON 2nd Edition.

Experiment: 1

Aim: A program for electricity bill taking different categories of users, different slabs in each category. (Using nested if else statement).

Domestic Customer:		
Consumption Units	Rate of Charges(Rs.)	
0 – 200	0.50 per unit	
201 – 400	100 plus	0.65 per unit
401 – 600	230 plus	0.80 per unit
601 and above	390 plus	1.00 per unit
Commercial Customer:		
Consumption Units	Rate of Charges(Rs.)	
0 – 50	0.50 per unit	
100 – 200	50 plus	0.6 per unit
201 – 300	100 plus	0.70 per unit
301 and above	200 plus	1.00 per unit

Source code:

```
#include<stdio.h>
void main()
{
    chat type;
    int units
    float bill;
    printf("d-Domestic customer\nc - Commercial customer");
    printf("\nEnter customer type :");
    scanf("%c",&type);
    printf("\nEnter no.of units  :");
    scanf("%d",&units);
    if(type=='d')
    {
        if(total > 0 && total <= 200)
        {
            bill = total * 0.50;
        }
        else if(total > 200 && total <= 400)
        {
            bill =100 + total * 0.65;
        }
        else if(total > 400 && total <= 600)
        {
            bill =230 + total * 0.80;
        }
        else
        {
            bill =390 + total * 1.00;
        }
        printf("\n\nyour bill is : Rs. %f",bill);
    }
    else if(type=='c')
    {
        if(total > 0 && total <= 100)
        {
            bill = total * 0.50;
        }
        else if(total > 100 && total <= 200)
        {
            bill =50 + total * 0.60;
```

```

    }
    else if(total > 200 && total <= 300)
    {
        bill =100 + total * 0.70;
    }
    else
    {
        bill =200 + total * 1.00;
    }
    printf("\n\nyour bill is : Rs. %f",bill);
}
else
{
    printf("\nInvalid Option\nPlease try again ...");
}
}

```

Output 1:

```

1.Domestic customer
2.Commercial customer
Enter your choice :1

Enter old units :400

Enter new units :543

Total number of units consumed : 143 units

your bill is : Rs. 71.50

```

Output 2:

```

1.Domestic customer
2.Commercial customer
Enter your choice :2

Enter old units :430

Enter new units :123

sorry! you entered wrong units..._

```

Experiment: 2(a)

Aim: Write a C program to evaluate the following (using loops):

a) $1 + x^2/2! + x^4 / 4! + \dots$ upto ten terms

Source code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,j,k,x,pwr;
    double sum=0.0, fact;
    clrscr();
    printf("enter value of X :");
    scanf("%d",&x);
    for(i=0;i<=10;i=i+2)
    {
        fact=1;
        pwr=1;
        for(k=1;k<=i;k++)
        {
            pwr = pwr * x;
            fact = fact * k;
        }
        sum = sum + (double)pwr/fact;
        printf("pwr = %d\tfact = %.3lf\tsum = %.3lf\n",pwr,fact,sum);

    }
    printf("\nResult :%.3lf\n",sum);
    getch();
}
```

Output:

```
enter value of X :2
pwr = 1 fact = 1.000    sum = 1.000
pwr = 4 fact = 2.000    sum = 3.000
pwr = 16 fact = 24.000   sum = 3.667
pwr = 64 fact = 720.000  sum = 3.756
pwr = 256 fact = 40320.000 sum = 3.762
pwr = 1024 fact = 3628800.000 sum = 3.762

Result :3.762
```

Experiment: 2(b)

Aim: Write a C program to evaluate the following (using loops):

b) $x + x^3/3! + x^5/5! + \dots$ upto 7 digit accuracy

Source code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,j,k,x,pwr;
    double sum=0.0, fact;
    clrscr();
    printf("enter value of X :");
    scanf("%d",&x);
    for(i=1;i<=10;i=i+2)
    {
        fact=1;
        pwr=1;
        for(k=1;k<=i;k++)
        {
            pwr = pwr * x;
            fact = fact * k;
        }
        sum = sum + (double)pwr/fact;
        printf("pwr = %d\tfact = %.3lf\tsum = %.3lf\n",pwr,fact,sum);

    }
    printf("\nResult :%.3lf\n",sum);
    getch();
}
```

Output:

```
enter value of X :3
pwr = 3 fact = 1.000    sum = 3.000
pwr = 27    fact = 6.000    sum = 7.500
pwr = 243    fact = 120.000    sum = 9.525
pwr = 2187    fact = 5040.000    sum = 9.959
pwr = 19683    fact = 362880.000    sum = 10.013

Result :10.013
```

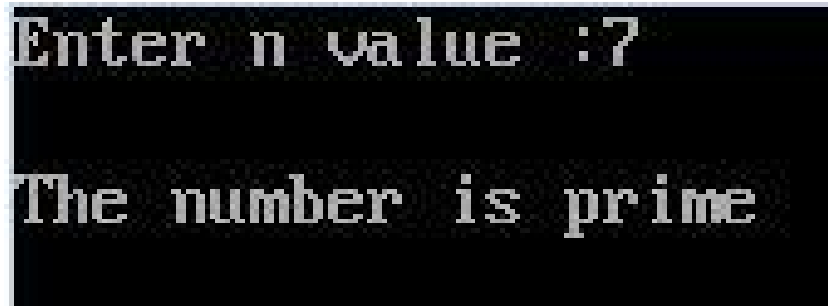
Experiment: 3(a)

Aim: Write a C program to check whether the given number is **Prime or not**.

Source code:

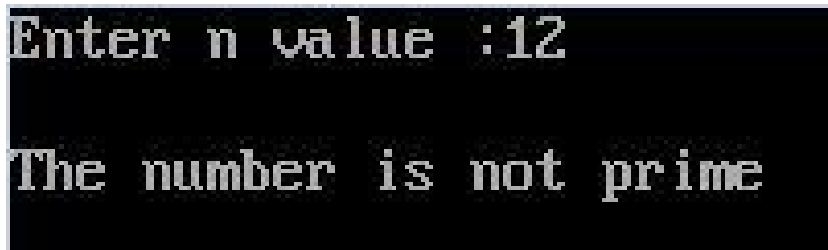
```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,count = 0;
    clrscr();
    printf("Enter n value :");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        if(n % i == 0)
        {
            count++;
        }
    }
    if(count == 2)
    {
        printf("\nThe number is prime\n");
    }
    else
    {
        printf("\nThe number is not prime\n");
    }
    getch();
}
```

Output 1:

A screenshot of a terminal window with a black background and white text. The first line shows the prompt "Enter n value :7". The second line shows the output "The number is prime".

```
Enter n value :7
The number is prime
```

Output 2:

A screenshot of a terminal window with a black background and white text. The first line shows the prompt "Enter n value :12". The second line shows the output "The number is not prime".

```
Enter n value :12
The number is not prime
```

Experiment: 3(b)

Aim: Write a C program to check whether the given number is **Perfect or Abundant or Deficient**.

Source code:

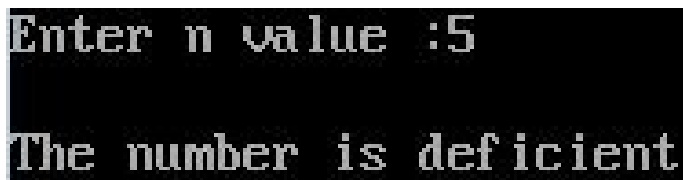
```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,sum = 0;
    clrscr();
    printf("Enter n value :");
    scanf("%d",&n);
    for(i=1;i<n;i++)
    {
        if(n % i == 0)
        {
            sum = sum + i;
        }
    }
    if(sum == n)
    {
        printf("\nThe number is perfect\n");
    }
    else if(sum > n)
    {
        printf("\nThe number is abundant\n");
    }
    else
    {
        printf("\nThe number is deficient\n");
    }
    getch();
}
```

Output 1:



Enter n value :12
The number is abundant

Output 2:



Enter n value :5
The number is deficient

Output 3:



Enter n value :6
The number is perfect

Experiment: 4

Aim: Write a C program to display statistical parameters (using one – dimensional array).

- a) Mean
- b) Mode
- c) Median
- d) Variance.

Source code:

```
#include<stdio.h>
#include<math.h>
void main()
{
    int x[10], i, j, n, temp, count;
    float mean, median, variance, mode, sum=0.0;
    clrscr();
    printf("Enter no.of elements :");
    scanf("%d" , &n);

    /* Read elements into Array */
    printf("Enter %d elements \n", n);
    for(i=0;i<=n-1;i++)
    {
        scanf("%d",&x[i]);
        sum=sum+x[i];
    }

    /* Mean value */
    mean = sum / n;

    /*variance value */
    sum=0;
    for(i=0;i<=n-1;i++)
    {
        sum = sum + (x[i]-mean)*(x[i]-mean);
    }
    variance = sum / n;

    /* sorting */
    for(j=0;j<=n-1;j++)
    {
        for(i=j+1;i<=n-1;i++)
        {
            if(x[j]>x[i])
            {
                temp=x[j];
                x[j]=x[i];
                x[i]=temp;
            }
        }
    }

    /* Travesing Array */
    printf("Soted elements \n");
```

```

for(i=0;i<=n-1;i++)
{
    printf("%d \t" , x[i]);
}

/* Median */
if(n % 2 == 0)
{
    median = (float)(x[n/2] + x[n/2-1]) / 2;
}
else
{
    median = x[n/2];
}
/* Mode */
temp = 0;
for(i=0; i<=n-1; i++)
{
    count = 0;
    for(j=i+1; j<=n-1; j++)
    {
        if(x[i] == x[j])
        {
            count++;
        }
    }
    if(count>temp)
    {
        temp = count;
        mode = x[i];
    }
}
printf("\nMean is : %f",mean);
printf("\nMode is : %f",mode);
printf("\nMedian is :%f",median);
printf("\nvariance is :%f",variance);
getch();
}

```

Output:

```

Enter no.of elements:7
Enter 7 elements
4
7
3
78
2
3
1
Sorted elements
1      2      3      3      4      7      78
Mean is : 14.000000
Mode is : 3.000000
Median is :3.000000
variance is :685.714294

```

Experiment: 5

Aim: Write a C program to read a list of numbers and perform the following operations

- Print the list.
- Delete duplicates from the list.

c) Reverse the list.

Source code:

```
#include<stdio.h>
#include<conio.h>
int DeletDupl(int n,int x[10])
{
    int i,j,k;
    for(j=0;j<n;j++)
    {
        for(i=j+1;i<n;i++)
        {
            if(x[j] == x[i])
            {
                for(k=i; k<=n-1; k++)
                {
                    x[k] = x[k+1];
                }
                n--;
            }
        }
    }
    return n;
}
void reverse(int n,int x[10])
{
    int i;
    for(i=n-1;i>=0;i--)
    {
        printf("%d\t", x[i]);
    }
}
void read(int n,int x[10])
{
    int i;
    printf("Enter %d elements :",n);
    for(i=0;i<n;i++)
    {
        scanf(" %d", &x[i]);
    }
}
void display(int n, int x[10])
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("%d\t",x[i]);
    }
}
void main()
{
    int x[10];
    int n,i,ele,ind;
    clrscr();
```

```

printf("Enter Number of elements :");
scanf("%d", &n);
read(n,x);
printf("\n the elements in List\n");
display(n,x);

printf("\nAfter Deleting Duplicates\n");
n=DeletDupl(n,x);
display(n,x);

printf("\nThe Reverse order List\n");
reverse(n,x);
getch();
}

```

Output:

```

Enter Number of elements :6
Enter 6 elements :4
6
2
4
8
6

the elements in List
4      6      2      4      8      6
After Deleting Duplicates
4      6      2      8
The Reverse order List
8      2      6      4      _

```

Experiment: 6

Aim: Write a C program to read a list of numbers and search for a given number using Binary search algorithm and if found display its index otherwise display the message “Element not found in the List”.

Source code:

```
#include<stdio.h>
#include<conio.h>
void Sorting(int n,int x[10])
{
    int i,j, temp;
    for(j=0;j<n;j++)
    {
        for(i=j+1;i<n;i++)
        {
            if(x[j]>x[i])
            {
                temp = x[j];
                x[j] = x[i];
                x[i] = temp;
            }
        }
    }
}
void BinarySearch(int f, int l, int ele, int x[10])
{
    int mid;
    int temp = -1;
    while(f<=l)
    {
        mid=(f+l)/2;
        if(x[mid] == ele)
        {
            printf("Element found %d index",mid);
            temp = 1;
            break;
        }
        else if(ele < x[mid])
        {
            l = mid -1;
        }
        else
        {
            f = mid + 1;
        }
    }
    if(temp == -1)
    {
        printf("Element not found");
    }
}
void read(int n,int x[10])
{
    int i;
    printf("Enter %d elements :",n);
    for(i=0;i<n;i++)
    {
        scanf(" %d", &x[i]);
    }
}
```

```

void display(int n,int x[10])
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("%d\t",x[i]);
    }
}
void main()
{
    int x[10];
    int n,i,ele,ind;
    clrscr();
    printf("Enter Number of elements :");
    scanf("%d", &n);
    read(n,x);
    Sorting(n,x);
    printf("After sorting\n");
    display(n,x);
    printf("\nEnter element to be search :");
    scanf("%d", &ele);
    BinarySearch(0,n-1,ele,x);

    getch();
}

```

Output 1:

```

Enter Number of elements :9
Enter 9 elements :34
12
6
8
4
5
3
9
2
After sorting
2      3      4      5      6      8      9      12      34
Enter element to be search :12
Element found 7 index

```

Output 2:

```

Enter Number of elements :5
Enter 5 elements :3
6
8
1
2
After sorting
1      2      3      6      8
Enter element to be search :9
Element not found

```

Experiment: 7

Aim: Write a C program to read two matrices and compute their sum and product.

Source code (matrix addition):

```

#include<stdio.h>
#include<conio.h>
void read(int A[2][2], int r, int c)
{
    int i,j;
    printf("Enter %d * %d Matrix\n",r,c);
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            scanf("%d", &A[i][j]);
        }
    }
}
void display(int A[2][2], int r, int c)
{
    int i,j;
    printf("The Matrix is\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf("%d\t", A[i][j]);
        }
        printf("\n");
    }
}
void sum(int A[2][2],int B[2][2],int C[2][2], int r1, int c1)
{
    int i,j;
    for(i=0;i<r1;i++)
    {
        for(j=0;j<c1;j++)
        {
            C[i][j] = A[i][j] + B[i][j];
        }
    }
}
void main()
{
    int A[2][2], B[2][2], C[2][2];
    int r1, c1, r2,c2;
    clrscr();
    printf("Enter A matrix row and column size :");
    scanf("%d %d", &r1, &c1);
    printf("Enter B matrix row and column size :");
    scanf("%d %d", &r2, &c2);
    if(r1==r2 && c1==c2)
    {
        read(A,r1,c1);
        read(B,r1,c1);
        display(A,r1,c1);
        display(B,r2,c2);
        printf("Sum of A and B Matris is:");
        sum(A,B,C,r1,c1);
    }
}

```

```

        display(C,r1,c1);
    }
    else
    {
        printf("\nSorry Addition is not possible");
    }
    getch();
}

```

Output 1:

```

Enter A matrix row and column size :2
2
Enter B matrix row and column size :2
2
Enter 2 * 2 Matrix
1
2
3
4
Enter 2 * 2 Matrix
5
6
7
8
The Matrix A is
1      2
3      4
The Matrix B is
5      6
7      8
Sum of A and B Matris is:
6      8
10     12

```

Output 2:

```

Enter A matrix row and column size :2
3
Enter B matrix row and column size :4
5

Sorry Addition is not posible_

```

Source code (matrix product):

```

#include<stdio.h>
#include<conio.h>
void read(int A[2][2], int r, int c)

```



```

{
    int i,j;
    printf("Enter %d * %d Matrix\n",r,c);
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            scanf("%d", &A[i][j]);
        }
    }
}

void display(int A[2][2], int r, int c)
{
    int i,j;
    printf("The Matrix is\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf("%d\t", A[i][j]);
        }
        printf("\n");
    }
}

void Multi(int A[2][2],int B[2][2],int C[2][2], int r1, int c1, int c2)
{
    int i,j,k;
    for(i=0;i<r1;i++)
    {
        for(j=0;j<c2;j++)
        {
            C[i][j]=0;
            for(k=0;k<c1;k++)
            {
                C[i][j] = C[i][j] + A[i][k] * B[k][j];
            }
        }
    }
}

void main()
{
    int A[2][2], B[2][2], C[2][2];
    int r1, c1, r2,c2;
    clrscr();
    printf("Enter A matrix row and column size :");
    scanf("%d %d", &r1, &c1);
    printf("Enter B matrix row and column size :");
    scanf("%d %d", &r2, &c2);
    if( c1==r2 )
    {
        read(A,r1,c1);
        read(B,r2,c2);
        display(A,r1,c1);
        display(B,r2,c2);
        printf("Multiplication of A and B Matrix is:");
        Multi(A,B,C,r1,c1,c2);
    }
}

```

```

        display(C,r1,c2);
    }
    else
    {
        printf("\n Sorry Multiplication is not possible");
    }
    getch();
}

```

Output 1:

```

Enter A matrix row and column size :2
2
Enter B matrix row and column size :2
2
Enter 2 * 2 Matrix
1
2
3
4
Enter 2 * 2 Matrix
5
6
7
8
The Matrix A is
1      2
3      4
The Matrix B is
5      6
7      8
Multiplication of A and B Matrix is:
19      22
43      50

```

Output 2:

```

Enter A matrix row and column size :2
3
Enter B matrix row and column size :2
2

    Sorry Multiplication is not possible

```

Experiment: 8

Aim: A menu driven program with options (using array of character pointers).

- a) To insert a student name
- b) To delete a student name

c) To print the names of students

Source code:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int i,j,ch,p=0;
char str[20][20],*ptr[20],del[20];
void insert();
void display();
void delet();
void main()
{
    int ch;
    clrscr();
    do
    {
        printf("\nWelcome to student operations\n");
        printf("\n1. Insert Student Name\n2. Delete Student Name\n3. Display Student Name\n4. Exit");
        printf("\n\nEnter your choice(1 to 4) :");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:insert();
                    break;
            case 2:delet();
                    break;
            case 3:display();
                    break;
            case 4:exit(0);
            default:printf("Sorry ! you entered rong choice\n");
        }
    }while(ch > 0);
    getch();
}
void insert()
{
    printf("Enter string to insert :");
    scanf("%s",&str[p]);
    ptr[p]=&str[p][0];
    printf("\nCongrats Student name inserted successfully\n");
    p++;
}
void delet()
{
    int count=0;
    printf("Enter string to delete :");
    scanf("%s",&del);
    for(i=0;i<p;i++)
    {
        if( strcmp(del, *(ptr+i)) == 0 )
        {
            for(j=i;j<p;j++)
            {
                *(ptr+j) = *(ptr+j+1);
            }
        }
    }
}
```

```

        printf("\n \'%s' String deleted successfully\n",del);
        count++;
        p--;
        break;
    }
}
if(count==0)
    printf("\n Sorry!! \'%s' String not fount\n",del);
}
void display()
{
    printf("\nStudent names are\n");
    for(i=0;i<p;i++)
    {
        printf("\n%s",*(ptr+i));
    }
}

```

Output:

```

Welcome to student operations

1. Insert Student Name
2. Delete Student Name
3. Display Student Name
4. Exit

Enter your choice(1 to 4) : 1
Enter string to insert : krishna

Congrats Student name inserted successfully

Welcome to student operations

1. Insert Student Name
2. Delete Student Name
3. Display Student Name
4. Exit

Enter your choice(1 to 4) :1
Enter string to insert : mahesh

Congrats Student name inserted successfully

Welcome to student operations

1. Insert Student Name
2. Delete Student Name
3. Display Student Name
4. Exit

Enter your choice(1 to 4) :3

Student names are

krishna
mahesh
Welcome to student operations

1. Insert Student Name
2. Delete Student Name
3. Display Student Name
4. Exit

```

```
Enter your choice(1 to 4) :2
Enter string to delete :lucky

Sorry!! 'lucky' String not found

Welcome to student operations

1. Insert Student Name
2. Delete Student Name
3. Display Student Name
4. Exit

Enter your choice(1 to 4) :2
Enter string to delete :mahesh

'mahesh' String deleted successfully

Welcome to student operations

1. Insert Student Name
2. Delete Student Name
3. Display Student Name
4. Exit

Enter your choice(1 to 4) :3

Student names are

krishna
Welcome to student operations

1. Insert Student Name
2. Delete Student Name
3. Display Student Name
4. Exit

Enter your choice(1 to 4) :4_
```

Experiment: 9

Aim: Write a C program to read list of student names and perform the following operations

- a) To print the list of names.
- b) To sort them in ascending order.
- c) To print the list after sorting.

Source code:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void read(char s[][20],int n, char *ptr[]);
void display(char *ptr[],int n);
void sort(char *ptr[],int n);
void main()
{
    int i,j,n;
    char s[20][20],*ptr[20];
    clrscr();
    printf("Enter how many strings you want :");
    scanf("%d",&n);
    read(s,n,ptr);
    printf("\nEnter strings are...\n");
    display(ptr,n);
    sort(ptr,n);
    printf("\nsorted strings are...\n");
    display(ptr,n);
    getch();
}
void read(char s[][20],int n, char *ptr[])
{
    int i;
    printf("Enter %d strings\n",n);
    for(i=0;i<n;i++)
    {
        scanf("%s", &s[i]);
        ptr[i] = &s[i][0];
    }
}
void display(char *ptr[],int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("%s\n",*(ptr+i));
    }
}
void sort(char *ptr[],int n)
{
    int i,j;
    char temp[10];
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
```

```

{
    if(strcmp(*(ptr+i),*(ptr+j))>0)
    {
        strcpy(temp,*(ptr+i));
        strcpy(*(ptr+i),*(ptr+j));
        strcpy(*(ptr+j),temp);
    }
}
}
}

```

Output:

```

Enter how many strings you want :5
Enter 5 strings
apple
cat
bat
banana
orange

Entered strings are...
apple
cat
bat
banana
orange

sorted strings are...
apple
banana
bat
cat
orange

```

Experiment: 10(a)

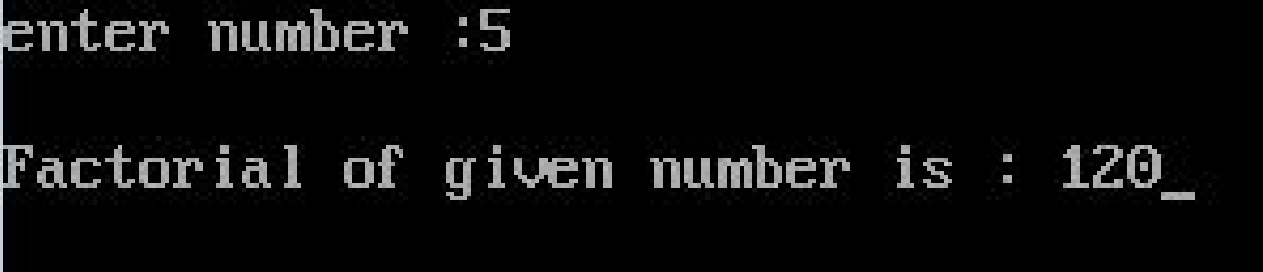
Aim: Write a C program that consists of recursive functions to

a) Find factorial of a given number

Source code:

```
#include<stdio.h>
#include<conio.h>
int fact(int);
void main()
{
    int fact,n;
    clrscr();
    printf("enter number :");
    scanf("%d",&n);
    fact = factorial(n);
    printf("\nFactorial of given number is : %d", fact);
    getch();
}
int factorial(int f)
{
    if(f == 1)
    {
        return 1;
    }
    return( f * factorial(f-1) );
}
```

Output:

A screenshot of a terminal window with a black background and white text. The first line shows the prompt 'enter number :5'. The second line shows the output 'Factorial of given number is : 120_'.

```
enter number :5
Factorial of given number is : 120_
```


Experiment: 10(b)

Aim: Write a C program that consists of recursive functions to

b) Solve towers of Hanoi with three towers (A, B & C) and three disks initially on tower A.

Source code:

```
#include <stdio.h>

// C recursive function to solve tower of hanoi puzzle
void towerOfHanoi(int n, char S, char T, char D)
{
    if (n > 0)
    {
        towerOfHanoi(n-1, S, D, T);
        printf("\n Move disk %d from  %c to  %c", n, S, D);
        towerOfHanoi(n-1, T, S, D);
    }
}

void main()
{
    int n = 3;
    clrscr();
    towerOfHanoi(n, 'A', 'B', 'C');
    getch();
}
```

Output:-



```
Move disk 1 from  A to  C
Move disk 2 from  A to  B
Move disk 1 from  C to  B
Move disk 3 from  A to  C
Move disk 1 from  B to  A
Move disk 2 from  B to  C
Move disk 1 from  A to  C
```