PRACTICAL NO- 01

The binary search procedures to search on element in a given list.

```
# include < stdio.h>
# Include < conio.h>
  void main()
 {
   clr scr();
   inta[5] = {1, 2, 3, 4, 5};
   int Beg = 1 . End=5 , temp =0 mid. n;
   print f(" enter any number\n");
   Scanf ("% d", &n);
  while ( temp = = 0 && End > Beg)
  {
   mid =( End + Beg) /2;
   if (n>a [mid])
   {
    Beg = mid;
   }
    if (n<a [mid])
    {
      End = mid;
     }
     if (n==a[mid])
{
temp= 1;
}
if (temp = =1)
{
printf("\n%.d is at location%. d\n", n, mid);
 }
  getch ();
 }
```

PRACTICAL NO- 02

The selection sort technique.

```
# Include < Stdio.h>
# Include< conio.h>
  void main()
  {
  clr ser ();
  int i, j. temp. a [5] = {5, 2, 1, 4, 3};
  for (i=0. i<=4; i++)
  {
  for
  (i=1 ; j<=4; j++)
  {
 if (a [i] < a [j+1])
 {
  temp = a [1];
  a[ i] = a[i+j];
  a [ j+ 1] = temp;
 }
 {
  for(i=0; i<>4; i++)
 {
  for(i=1; j<>4; j++)
  {
```

```
if (a [&) <a [j+i] )
{
 temp = a[j];
  a[j] = a[j+1);
  a [j+1] = temp;
  }
  }
  }
for (i=0; i<=4; i++)
{
 printf ("%d\n" a [i]);
 }
  getch ();
 }
```

PRACTICAL NO- 03

The factorial of given number with reacusion.

```
  # include< stdio.h>
  # includec conio.h>
     void main()
    {
    clrscr ();
    unsigned int n;
   printf ("Enter any number\n");
   scanf ("%d ,&n);
   printf ("The factorial of %d is%li",
          n, fact(n));
    getch ();
   }
   ( long fact unsigned int x)
   {
   if (x==0)
   {
   return 1;
   }
   else
   {
  return fact (x-1)*x;
 }
```

PRACTICAL NO- 04

The fabonaci series with recursion;

```
# include< stdio.h>
#   include< conio.h>
   void main ()
 {
   clrscr ();
   int n ;
  void ~~f~~fibo (int);
  printf ("Enter any number \n");
 Scanf ("%d", &n);
 Printf (" The fibonacci series:");
  Fibo (n);
  getch ();
  }
   void fibo (int x)
  {
  int a=0, b= 1,; i;
 print f ("%d\n %d\n", a,b);
 for(i=0; i<1 ; ++)
 {
  c= a+b;
  printf ("%d\n", c);
   a=b;
   b= c;
 }
 }
```

PRACTICAL NO- 05

Addition of two matrix using function.

```
 # include< stdio.h>
 # include< conio.h>
  int rows, columns;
 void matrix addition (int mat1 [] [10], int mat2
  [] [10] int mat3 [] [10]
  {
   int i, j;
  for (i=0; I <rows ; i++)
  {
  for (i=0; j< columns ; j++)
 mat 3 [i] [] = mat 1 [i] [j] + mat 2 [i] [j] ;
 }
 return;
 }
 int main ()
 {
int matrix 1 [10] [10], matrix 2 [10] [10];
int matrix 3 [10] [10], i , j ;
printf (" Enter the no of rows and (columns (<=
     10) ; ");
 scanf ("%d %d", & rows, & column);
for (i=0 ; < 1 J++)
}
```

PRACTICAL NO- 06

multiplication of two matrix.

```
# include< stdio.h>
# include< conio.h>
 int main()
 {
  clrscr()
  int a[10][10], b[10][10], c[b][10];
  int r1, c1, r2, c2;
  int i, j;
printf("enter the row and column of second
                       matrix:);
scanf("%d", &r1, &c1);
printf("Enter the row and column of second matrix:")
scanf("%d", &r2, &c2);
if (c1 != r2);
{
printf("\n matrix are not multipication\n");
}
printf("\n enter element of matrix A:\n");
 for (i=0; i<r1; j++)
 scanf("%d\t", &a[i][j]);
}
}
```

```
print f ("\n Enter element of matrix b:\n");
  for (i=0; <+ j++)
 {
 Scanf ("%d\n", & b [i] [j]);
 }
 }
for (i=0; i<=r, ; j++)
{
for (i=0; j< columns; J++)
{
printf ("%sd", matrix 3 [i] [J];
 }
print f ("\n");
}
return 0;
}
```

PRACTICAL NO- 07

Push and pop operation in stack.

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

# define max5 // maximum number of element
   that can be stored.
    int top =1, stack [max];
     void  PUSH();
     void   POP ();
      void  display ();
      void  main ();
   {
    int ch;
  while(1) // infinite loop. will end when choice
  will be 4{
 printf("n*** stack menu*** ");
 printf ("\n\n 1. PUSH\n2 POP\n3. Display\n 4
     exit");
printf ("\n\n enter your choice (1-4):,");
 scanf ("%d, &ch);
 switch(ch)
 {
 stack (top) = val;
  }
 }
void pop ()
{
 if Top (= = -1){
print f ("\n stack is empty %d", stack [TOP])
   top = top - 1;

 }

 }


 void display ()
{
 int i;
if (top = = - 1 ){
print f ("\n Stack is empty !"); }
 case 1 : PUSH ( );
 ~~cas~~ break;
 case 2 : Pop ();
  break :
  case 3 : display ();
  break
  case 4 Exit (0);
  default : Print f ("\n wrong choice !");
  }
  }
  }
```

```
void  PUSH()
{
 int val;
 if (top = = max-1)
 printf ("n Stack is full !!");
}

printf ("\n Enter element of  PUSH :");
 Scanf (" %d" & );
  top  =  top + 1;
```