

Practical-1

1. Take array as input from user:

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
void main()
{
    int a[100],i,size;
    system("color f0");
    printf("enter size:");
    scanf("%d",&size);
    for(i=0;i<size;i++)
    {
        printf("\n enter value for element a[%d]:",i+1);
        scanf("%d",&a[i]);
    }
    getch();
}
```

Output:

```
enter size:5
enter value for element a[1]:10
enter value for element a[2]:100
enter value for element a[3]:1000
enter value for element a[4]:10000
enter value for element a[5]:20
```

2. Print array elements:

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
void main()
{
    int a[100],i,size;
    system("color f0");
    printf("enter size:");
    scanf("%d",&size);
    for(i=0;i<size;i++)
    {
        printf("\n enter value for element a[%d]:",i+1);
        scanf("%d",&a[i]);
    }
    printf("Array Contains:\n");
    for(i=0;i<size;i++)
        printf("A[%d]=%d\n",i+1,a[i]);
    printf("Thank You %c",2);
    getch();
}
```

Output:

```
enter size:5
enter value for element a[1]:10
enter value for element a[2]:20
enter value for element a[3]:30
enter value for element a[4]:40
enter value for element a[5]:50
Array Contains:
A[1]=10
A[2]=20
A[3]=30
A[4]=40
A[5]=50
Thank You  
```

3. Insert element into array:

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
void main()
{
    int a[100],i,pos,n,val;
    system("color f0");
    printf("Enter Size of Array :");
    scanf("%d",&n);
    printf("Enter %d elements:",n);
    for(i=0;i<n;i++)
    {
        printf("A[%d]=",i+1);
        scanf("%d",&a[i]);
    }
    printf("Enter position where you want to insert number:");
    scanf("%d",&pos);
    printf("Enter Value to be inserted:");
    scanf("%d",&val);

    for(i=n-1;i>=pos-1;i--)
        a[i+1]=a[i];
    a[pos-1]=val;

    printf("\nAfter Inserting an element:\n");
    for (i=0;i<=n;i++)
        printf("A[%d]=%d\n",i+1,a[i]);
    getch();
}
```

Output:

```
Enter Size of Array :4
Enter 4 elements:A[1]=1
A[2]=2
A[3]=4
A[4]=5
Enter position where you want to insert number:3
Enter Value to be inserted:3

After Inserting an element:
A[1]=1
A[2]=2
A[3]=3
A[4]=4
A[5]=5
```

4. Delete element from array:

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#define p printf

void main()
{
    int n,i,a[50],po;
    system("color f0");
    p("Enter how many elements u want in ur array??");
    scanf("%d",&n);
    p("Enter Elements:\n");
    for(i=0;i<n;i++)
    {
        p("A[%d]=",i+1);
        scanf("%d",&a[i]);
    }
    p("Enter position to delete the element:");
    scanf("%d",&po);

    for(i=po-1;i<n;i++)
    {
        a[i]=a[i+1];
    }
    p("After Deletion Operation :\n");
    for(i=0;i<n-1;i++)
    {
        printf("A[%d]=%d\n",i+1,a[i]);
    }
    getch();
}
```

Output:

```
Enter how many elements u want in ur array??5
Enter Elements:
A[1]=11
A[2]=12
A[3]=19
A[4]=13
A[5]=14
Enter position to delete the element:3
After Deletion Operation :
A[1]=11
A[2]=12
A[3]=13
A[4]=14
```

5. Search element from an array(sequential search):

```
#include<stdio.h>
#define p printf
void main()
{
    int a[50],i,n,c=0,v;
    system("color f0");
    p("Enter how many elemnts u want in your array?");
    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        p("Enter a[%d]=",i+1);
        scanf("%d",&a[i]);
    }
    p("Enter value which u want to search:\n");
    scanf("%d",&v);
    for(i=0;i<n;i++)
    {
        if(a[i]==v)
        {
            c++;
            p("Yess!! We find ur value at position no %d",i+1);
            p("\n");
            break;
        }
    }
    if(c==0)
        p("Sorry!!We couldn't find ur value\n");
}
```

Output:

```
Enter how many elemnts u want in your array?5
Enter a[1]=12
Enter a[2]=14
Enter a[3]=56
Enter a[4]=78
Enter a[5]=90
Enter value which u want to search:
56
Yess!! We find ur value at position no 3
```

6. Search element from an array(binary search):

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#define p printf

void main()
{
    int a[50],i,n,v,l=0,h,j,t,f=0,m;
    system("color f0");
    p("Enter how many elements u want in your array?");
    scanf("%d",&n);
    h=n-1;
    p("Enter Values:\n");
    for(i=0;i<n;i++)
    {
        p("Enter a[%d]=",i+1);
        scanf("%d",&a[i]);
    }
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(a[i]>a[j])
            {
                t=a[i];
                a[i]=a[j];
                a[j]=t;
            }
        }
    }
    p("Enter value which u want to search:\n");
    scanf("%d",&v);
    while(l<=h)
    {
        m=(l+h)/2;
        if(v<a[m])
            h=m-1;
```

```
        else if(v>a[m])
            l=m+1;
        else
        {
            printf("Congrats.... Your value is found.\n Thank You %c",2);
            f++;
            break;
        }
    }

    if(f==0)
        p("Searching is unsuccseeful");
        getch();
}
```

Output:

```
Enter how many elements u want in your array?5
Enter Values:
Enter a[1]=12
Enter a[2]=67
Enter a[3]=4
Enter a[4]=34
Enter a[5]=90
Enter value which u want to search:
4
Congrats.... Your value is found.
Thank You 🍀
```

7. Sort the elements of an array: (ascending order)

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

#define p printf

void main()
{
    int a[100],i,j,n,t;
    system("color f0");
    p("Enter how many elements u want in ur array?");
    scanf("%d",&n);
    p("Enter elements:\n");
    for(i=0;i<n;i++)
    {
        printf("Enter A[%d]",i+1);
        scanf("%d",&a[i]);
    }
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(a[j]<a[i])
            {
                t=a[j];
                a[j]=a[i];
                a[i]=t;
            }
        }
    }
    p("After Ascending:\n");
    for(i=0;i<n;i++)
        p("A[%d]=%d\n",i+1,a[i]);
}
```

Output:

```
Enter how many elements u want in ur array?6
Enter elements:
Enter A[1]12
Enter A[2]67
Enter A[3]98
Enter A[4]65
Enter A[5]35
Enter A[6]22
After Ascending:
A[1]=12
A[2]=22
A[3]=35
A[4]=65
A[5]=67
A[6]=98
```


8. Create 2D array and print the elements of array:

```

#include<stdio.h>
#include<conio.h>
#include<windows.h>
#define p printf
void main()
{
    int a[100][100],i,m,j,n;
    system("color f0");
    printf("How many rows U want in ur array??");
    scanf("%d",&m);
    printf("How many columns U want in ur array??");
    scanf("%d",&n);
    printf("Enter Elements:");

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("\nA[%d][%d]=",i+1,j+1);
            scanf("%d",&a[i][j]);
        }
    }

    printf("Your 2-D Array :\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }

    getch();
}

```

Output:

```

How many rows U want in ur array??2
How many columns U want in ur array??3
Enter Elements:
A[1][1]=12
A[1][2]=56
A[1][3]=78
A[2][1]=90
A[2][2]=23
A[2][3]=45
Your 2-D Array :
12      56      78
90      23      45

```

Practical-2

1. Find length of given string.

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#define n printf("\n");

void print()
{
    int i;
    for(i=0;i<40;i++)
        printf("==");
}

void len(char *);
void main()
{
    system("color fc");
    char a[50];
    printf("Enter ur name:");
    gets(a);
    len(a);
    getch();
}

void len(char *x)
{
    int i=0;
    while(x[i]!='\0')
        i++;
    print();
    n n
    printf("Ur name has %d characters.", i);
    printf("\n Thank You %c",1);
    n n
    print();
}

}
```

Output:

```
Enter ur name:Himani Trivedi
=====
Ur name has 14 characters.
Thank You ☺
=====
```

2. Convert UPPER to lower case and vice-versa.

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#define n printf("\n");
#define p printf

void lwr(char *x)
{
    int i=0;
    for(;*x!=0;x++)
    {
        if(*x>='A'&&*x<='Z')
            *x+=32;
        else if(*x>='a'&&*x<='z')
            *x-=32;
    }
}

void main()
{
    char a[50];
    system("color f0");
    p("Enter Ur name:");
    gets(a);
    lwr(a);
    p("Your Converted String is:");
    p("%s",a);
    getch();
}
```

Output:

```
Enter Ur name:HIMANI trivedi
Your Converted String is:himani TRIVEDI
Process returned 13 (0xD)   execution tir
Press any key to continue.
```

3. Concatenate two strings.

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

```
#include<conio.h>
#include<windows.h>
#define n printf("\n");
#define p printf

void cat(char *a,char *c, char *b)
{
    int i;
    for(i=0;*a!='\0';a++,b++)
        *b=*a;
        *b++=' ';
    for(i=0;*c!='\0';c++,b++)
        *b=*c;
}

void main()
{
    char a[100],b[200],c[100];
    system("color f0");
    p("Enter Ur First Name:");
    gets(a);
    p("Enter Ur Surname:");
    gets(c);
    cat(a,c,b);
    n n
    p("Your Name is: ");
    p("%s",b);
}
```

Output:

```
Enter Ur First Name:Himani
Enter Ur Surname:Trivedi

Your Name is: Himani Trivedi
Process returned 14 (0xE)   ex
Press any key to continue.
```

4.

4. Reverse string.

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#define n printf("\n");
#define p printf

void rev(char *,char *);
void main()
{
    char a[50],b[50];
    int i;
    system("color f0");
    print(); n n
    p("Enter Ur name:");
    gets(a);
    rev(a,b);
    p("Reverse Of Ur Name:");
    puts(b);
    n n
    print();
}
void rev(char *x, char *y)
{
    int len=strlen(x);
    for(len-=1;len>=0;len--)
    {
        *y=x[len];
        y++;
    }
    y='\0';
}
```

Output:

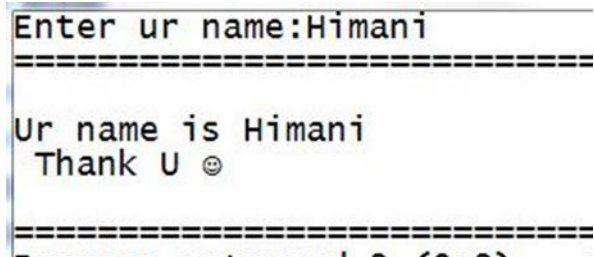
```
=====
Enter Ur name:Himani Trivedi
Reverse Of Ur Name:ideviriT inamiH
=====
```

5. Copy string.

```
=====
Process returned 2 (0x2)    execution time :
=====
```

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#define n printf("\n");
#define p printf

void copy(char *,char *);
void main()
{
    system("color f0");
    char a[50],b[50];
    p("Enter ur name:");
    gets(a);
    copy(b,a);
}
void copy(char *x,char *y)
{
    int i=0;
    while(y[i]!='\0')
    {
        x[i]=y[i];
        i++;
    }
    x[i]='\0';
    print();
    n n
    p("Ur name is %s",x);
    p("\n Thank U %c",1);
    n n
    print();
}
```

Output:

```
Enter ur name:Himani
=====
Ur name is Himani
Thank U 😊
=====
```

6. Compare two strings.

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#define n printf("\n");
#define p printf
void cmp(char *x,char *y);
void main()
{
    system("color f0");
    char a[50],b[50];
    print();n n
    p("Enter ur name:");
    gets(a);
    p("Confirm ur name:");
    gets(b);
    cmp(a,b);
    getch();
}

void cmp(char *x,char *y)
{
    int i,len1=0,len2=0,j=0;
    for(i=0;x[i]!='\0';i++)
        len1++;

    for(i=0;y[i]!='\0';i++)
        len2++;

    if(len1!=len2)
    {
        p("Pls reenter ur name"); n n
        system("color 0f");
        return;
    }
    else
    {
```

```
    for(i=0;i<len1;i++)
    {
        if(x[i]!=y[i])
            break;
        else
            j++;
    }

    if(j==len1)
    {
        p("\n Thank U %c",2);
        n n
    }
    else
    {
        p("Sorry....Pls reenter ur name");
        n n
        system("color 0f");
    }
    print();
}
}
```

Output:

```
=====
Enter ur name:Himani Trivedi
Confirm ur name:Himani Trivedi
Thank U 🍷
=====
```


7. Find substring.

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#define n printf("\n");
#define p printf

int len(char *x)
{
    int i=0;
    while(x[i]!='\0')
        i++;
    return i;
}

int cmp(char *x,char *y)
{
    int i,len1=0,len2=0,j=0;
    for(i=0;x[i]!='\0';i++)
        len1++;

    for(i=0;y[i]!='\0';i++)
        len2++;

    if(len1!=len2)
    {
        system("color 0f");
        return 0;
    }
    else
    {
        for(i=0;i<len1;i++)
        {
            if(x[i]!=y[i])
                break;
            else
                j++;
        }
    }
}
```

```
    }

    if(j==len1)
    {
        system("color f0");
        p("\n Congrats Ur String is Found ...\nThank U %c",2);
        printf("\n");
        return 2;
    }
    else
    {
        system("color 0f");
        return 0;
    }

}
}
void main()
{
    char a[1000],x[100],y[100];
    int i,q,j=0,o=0,po,re;
    system("color f9");
    p("Enter Ur String:");
    gets(a);
    p("\n");
    p("Enter String which U want to Find:");
    gets(x);
    q=len(x);
    for(i=0;a[i]!='\0';i++)
    {
        if(x[j]==a[i])
        {
            po=i;
            for(;q!=0;q--)
            {
                y[o]=a[i];
                i++;
                o++;
            }
            y[o]='\0';
        }
    }
}
```

```
        re=cmp(x,y);
        if(re==2)
            goto x;
        else
        {
            q=len(x);
            o=0;
        }
    }
    x:
    p("\n%s is at position no %d\n",y,po+1);
    print();
    getch();
}
```

Output:

```
Enter Ur String:My name is Himani Trivedi Akshaykumar.
Enter String which U want to Find:Himani
Congrats Ur String is Found ...
Thank U ☺
Himani is at position no 12
=====
```

8. Insert substring.

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#define n printf("\n");
#define p printf
```

```
int len(char *x)
{
    int i=0;
    while(x[i]!='\0')
        i++;
    return i;
}

void add(char *a,char *x,int po)
{
    int i=0;
    char str[50];
    while(i<po-1)
    {
        str[i]=*a;
        a++;
        i++;
    }
    while(*x!='\0')
    {
        str[i]=*x;
        x++;
        i++;
    }
    while(*a!=NULL)
    {
        str[i]=*a;
        a++;
        i++;
    }
    str[i]='\0';
    puts(str);
}

void main()
{
    char a[100],x[100];
    int po;
    system("color f0");
    p("Enter Ur String:");
    gets(a);
```

```
p("Length of String is %d\n",len(a));
p("\nEnter at which position u want to add Ur string: ");
scanf("%d",&po);
fflush(stdin);
p("\nNow Please enter Ur String which u want to enter: ");
gets(x);
p("Length of String is %d\n\n",len(x)) ;
add(a,x,po);
getch();
}
```

Output:

```
Enter Ur String:Himani Akshaykumar.
Length of String is  19

Enter at which position u want to add Ur string: 8

Now Please enter Ur String which u want to enter: Trivedi
Length of String is  8

Himani Trivedi Akshaykumar.
```

9. Delete substring.

```
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#define n printf("\n");
#define p printf

void del(char *a,char *x,int po)
{
    int l,i=0;
    char y[100];
    p("How long u want to delete the characters??");
```

```
scanf("%d",&l);
po-=1;
while(po--)
{
    y[i]=*a;
    i++;
    a++;
}
while(l--)
    a++;
while(*a!=NULL)
{
    y[i]=*a;
    a++;
    i++;
}
y[i]='\0';
puts(y);
}
void main()
{
    char a[100],x[100];
    int po;
    system("color f0");
    p("Enter Ur String:");
    gets(a);
    p("\n Enter From which position u want to delete the string");
    scanf("%d",&po);
    del(a,x,po);
}
```

Output:

```
Enter Ur String:Himani Abt Trivedi.
Enter From which position u want to delete the string8
How long u want to delete the characters??3
Himani Trivedi.
```

Sub name: Data structure

Enrollment no: 196170307149

3. Stack & Queue

18. Push, Pop, Peep and Update Operations.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void push();
void update();
void pop();
void peep();
#define p printf
#define n printf("\n");
int top=-1,a[20];

void main()
{
    int ch;
    do
    {
        p("\nEnter Ur choice for below Operation on stack"); n
        p("1.Push"); n
        p("2.Pop"); n
        p("3.Update");n
        p("4.Peep"); n
        p("5.Exit"); n
        p("Enter Ur Choice:");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:
                push();
                break;
            case 2:
                pop();
                break;
            case 3:
                update();
                break;
            case 4:
```



```
                peep();
                break;
            case 5: exit(0);
            default:
                p("Please Enter Valid Choice.\n");
        }
    }while(ch!=5);
}

void push()
{
    int x;
    if(top>=20)
        p("\nStack Overflow\n");
    else
    {
        p("Enter Value which u want to Push");
        scanf("%d",&x);
        top++;
        a[top]=x;
    }
}

void peep()
{
    if(top==-1)
        printf("Stack is empty");
    else
        p("\n Ur top of the stack is  %d and the value of it is
%d",top,a[top]); n n
}

void update()
{
    int x,po;
    if(top!=-1)
    {
        p("Enter at which Position U want to update the value:
");
        scanf("%d",&po);
```

```
        p("Enter Value which u want to update:");
        scanf("%d",&x);
        if(a[po-1]=x)
            p(" Ur value is successfully updated.. "); n n
    }
    else
        p("Stack is empty");

}

void pop()
{
    int x;
    if(top<0)
        p("\nStack Underflow ");
    else
    {
        x=a[top];
        top--;
        p("\n U have Poped out %d from the stack",x); n
    }
}
```

Output:-

```
Enter Ur choice for below Operation on stack
1.Push
2.Pop
3.Update
4.Peep
5.Exit
Enter Ur Choice:4
Stack is empty

Enter Ur choice for below Operation on stack
1.Push
2.Pop
3.Update
4.Peep
5.Exit
Enter Ur Choice:1
Enter Value which u want to Push12

Enter Ur choice for below Operation on stack
1.Push
2.Pop
3.Update
4.Peep
5.Exit
Enter Ur Choice:4

Ur top of the stack is 0 and the value of it is 12
```

```
Enter Ur choice for below Operation on stack
1.Push
2.Pop
3.Update
4.Peep
5.Exit
Enter Ur Choice:3
Enter at which Position U want to update the value: 2
Enter Value which u want to update:12
Ur value is successfully updated..

Enter Ur choice for below Operation on stack
1.Push
2.Pop
3.Update
4.Peep
5.Exit
Enter Ur Choice:4

Ur top of the stack is 1 and the value of it is 12

Enter Ur choice for below Operation on stack
1.Push
2.Pop
3.Update
4.Peep
5.Exit
Enter Ur Choice:5
```

```
Enter Ur choice for below Operation on stack
1.Push
2.Pop
3.Update
4.Peep
5.Exit
Enter Ur Choice:1
Enter Value which u want to Push:11

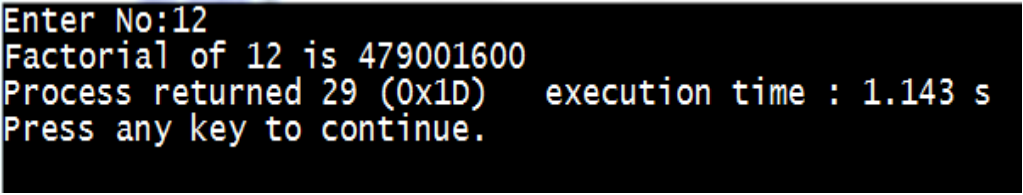
Enter Ur choice for below Operation on stack
1.Push
2.Pop
3.Update
4.Peep
5.Exit
Enter Ur Choice:4

Ur top of the stack is 1 and the value of it is 11
```

19. Write a Program to find Factorial using recursion

```
#include<stdio.h>
#include<conio.h>
int long fact(int long );
void main()
{
    int long a;
    printf("Enter No:");
    scanf("%ld",&a);
    printf("Factorial of %ld is %ld ",a,fact(a));
    return 0;
}

int long fact(int long a)
{
    if(a==0 || a==1)
        return 1;
    else
    {
        return(a*fact(a-1));
    }
}
```

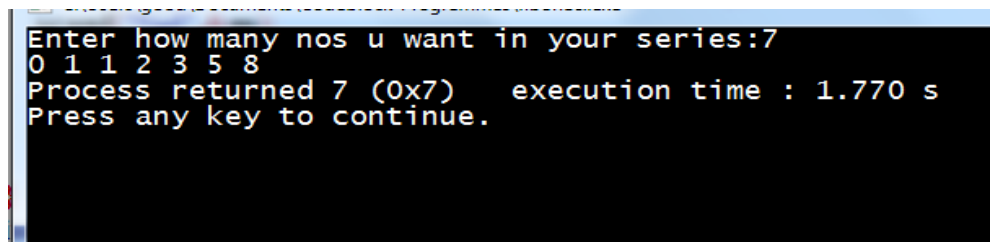
Output:-A screenshot of a terminal window with a black background and white text. The text shows the program's execution: it prompts for an input number (12), calculates its factorial (479001600), displays the result, shows the process ID (29) and execution time (1.143 s), and finally prompts the user to press any key to continue.

```
Enter No:12
Factorial of 12 is 479001600
Process returned 29 (0x1D)   execution time : 1.143 s
Press any key to continue.
```

20. Write a program to find Fibonacci series using recursion

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

int f(int);
void main()
{
    int i,m;
    printf("Enter nos u want in your series:");
    scanf("%d",&m);
    for(i=1;i<=m;i++)
    {
        printf("%d ",f(i));
    }
}
int f(int m)
{
    if(m==1)
        return 0;
    else if(m==2)
        return 1;
    else
        return (f(m-1)+ f(m-2));
}
```

Output:A screenshot of a terminal window showing the execution of a C program. The prompt 'Enter how many nos u want in your series:' is followed by the input '7'. The output displays the first 7 terms of the Fibonacci series: '0 1 1 2 3 5 8'. Below the output, it shows 'Process returned 7 (0x7) execution time : 1.770 s' and 'Press any key to continue.'.

```
Enter how many nos u want in your series:7
0 1 1 2 3 5 8
Process returned 7 (0x7) execution time : 1.770 s
Press any key to continue.
```

21.GCD using recursion.

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

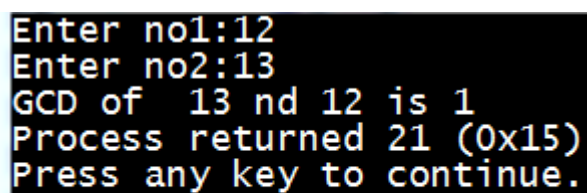
int gcd(int,int);
void main()
{
    int x,y,t;
    printf("Enter no1:");
    scanf("%d",&x);

    printf("Enter no2:");
    scanf("%d",&y);

    if(y>=x)
    {
        t=x;
        x=y;
        y=t;
    }

    printf("GCD of  %d nd %d is %d",x,y,gcd(x,y));
    return 0;
}

int gcd(int x,int y)
{
    if(x%y==0)
        return y;
    else
        return (gcd(y,x%y));
}
```

Output:

```
Enter no1:12
Enter no2:13
GCD of 13 nd 12 is 1
Process returned 21 (0x15)
Press any key to continue.
```

22.Insert Operation in queue.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int a[5],f=-1,r=-1;
void rear();
void display();

void main()
{
    int ch;
    do
    {
        printf("1.insertion\n");
        printf("2.Display\n");
        printf("3.Exit\n");
        printf("Enter ur choice:");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:rear();
                break;
            case 2:display();
                break;
            case 3:exit(0);
            default:printf("Please Enter valid Choice\n");
        }
    }while(ch!=3);
}

void rear()
{
    if(r>=4)
    {
        printf("\nQueue Overflow\n");
    }
    else
```

```

    {
        r++;
        if(r==0)
            f=0;
        printf("Enter value:");
        scanf("%d",&a[r]);
        // printf("\n Value of R is %d",r);
    }
}

void display()
{
    int i=0;
    if(f==-1)
        printf("Queue is Empty\n");
    else
        for(i=f;i<=r;i++)
            printf("%d\t",a[i]);
    printf("\n");
}

```

Output:-

```
1.insertion
2.Display
3.Exit
Enter ur choice:2
Queue is Empty

1.insertion
2.Display
3.Exit
Enter ur choice:1
Enter value:11
1.insertion
2.Display
3.Exit
Enter ur choice:1
Enter value:111
1.insertion
2.Display
3.Exit
Enter ur choice:1
Enter value:1111
```

```
Enter ur choice:1
Enter value:1111
1.insertion
2.Display
3.Exit
Enter ur choice:1
Enter value:11111
1.insertion
2.Display
3.Exit
Enter ur choice:1
Enter value:111111
1.insertion
2.Display
3.Exit
Enter ur choice:1
Queue Overflow
1.insertion
2.Display
3.Exit
Enter ur choice:2
11      111      1111      11111      111111
1.insertion
2.Display
3.Exit
Enter ur choice:3
```


23.Delete Operation in queue.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int a[5],f=-1,r=-1;
void rear();
void display();
void front()
{
    if(f==-1)
    {
        printf("\n Queue underflow\n");
    }
    else if(f==r)
    {
        printf("\n%d is deleted from the queue\n",a[f]);
        f=r=-1;
    }
    else
    {
        printf("\n%d is deleted from the queue\n",a[f]);
        f++;
    }
}

void main()
{
    int ch;
    do
    {
        printf("1.insertion\n");
        printf("2.Display\n");
        printf("3.Deletion\n");
        printf("4.Exit\n");
```

```
        printf("Enter ur choice:");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:rear();
                break;
            case 2:display();
                break;
            case 3:front();
                break;
            case 4:exit(0);
            default:printf("Please Enter valid Choice\n");
        }
    }while(ch!=4);
}

void rear()
{
    if(r>=4)
    {
        printf("\nQueue Overflow\n");
    }
    else
    {
        r++;
        if(r==0)
            f=0;
        printf("Enter value:");
        scanf("%d",&a[r]);
        // printf("\n Value of R is %d",r);
    }
}

void display()
{
    int i=0;
    if(f== -1)
        printf("Queue is Empty\n");
    else
```

```

        for(i=f;i<=r;i++)
            printf("%d\t",a[i]);
    printf("\n");
}

```

Output:-

```

1.Insertion
2.Display
3.Deletion
4.Exit
Enter ur choice:2
Queue is Empty
1.Insertion
2.Display
3.Deletion
4.Exit
Enter ur choice:1
Enter value:13
1.Insertion
2.Display
3.Deletion
4.Exit
Enter ur choice:1
Enter value:14
1.Insertion
2.Display
3.Deletion
4.Exit
Enter ur choice:2
13    14

```

```

1.Insertion
2.Display
3.Deletion
4.Exit
Enter ur choice:3
13 is deleted from the queue
1.Insertion
2.Display
3.Deletion
4.Exit
Enter ur choice:2
14
1.Insertion
2.Display
3.Deletion
4.Exit
Enter ur choice:4

```

24.Insert Operation in circular queue.

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int r=-1,f=-1,a[3];
void rear();
void peep();
void main()
{
    int ch;
    do
    {
        printf("\n1.Insertion");
        printf("\n2.Peep");
        printf("\n3.Exit");
        printf("\nEnter ur choice for above operations on Queue:");
        scanf("%d",&ch);
        switch (ch)
        {
            case 1: rear();
                    break;

```

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

        case 3:exit(0);

        default:printf("\n Please Enter Valid Choice for
operation");
    }
    }while(ch<5);

}
void rear()
{
    r++;
    if(r==0)
        f=0;
    else if (r==3)
        if(f>0)
            r=0;
        else
        {
            printf("\n Queue Overflow");
            exit(0);
        }
    else if(f==r)
    {
        printf("\n Queue Overflow");
        exit(0);
    }

    printf("Enter ur value:");
    scanf("%d",&a[r]);
}
void peep()
{
    if(f==-1)
        printf("\nQueue is empty");
    else
        printf("\n%d is value of %d front pointer and %d is value
of %d rear pointer of queue",a[f],f,a[r],r);
```

}

Output:-

```

1.Insertion
2.Peep
3.Exit
Enter ur choice for above operations on Queue:2

Queue is empty
1.Insertion
2.Peep
3.Exit
Enter ur choice for above operations on Queue:1
Enter ur value:12

1.Insertion
2.Peep
3.Exit
Enter ur choice for above operations on Queue:2

12 is value of 0 front pointer and 12 is value of 0 rear pointer of queue
1.Insertion
2.Peep
3.Exit
Enter ur choice for above operations on Queue:1
Enter ur value:111

1.Insertion
2.Peep
3.Exit

```

```

Enter ur value:111

1.Insertion
2.Peep
3.Exit
Enter ur choice for above operations on Queue:2

12 is value of 0 front pointer and 111 is value of 1 rear pointer of queue
1.Insertion
2.Peep
3.Exit
Enter ur choice for above operations on Queue:1
Enter ur value:121324

1.Insertion
2.Peep
3.Exit
Enter ur choice for above operations on Queue:2

12 is value of 0 front pointer and 121324 is value of 2 rear pointer of queue
1.Insertion
2.Peep
3.Exit
Enter ur choice for above operations on Queue:3

```

25.Delete Operation in circular queue.

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int r=-1,f=-1,a[3];
void rear();
void front();
void peep();

void main()
{
    int ch;
    do
    {
        printf("\n1.Insertion");
        printf("\n2.Deletion");
        printf("\n3.Peep");
        printf("\n4.Exit");
        printf("\nEnter ur choice for above operations on Queue:");
        scanf("%d",&ch);
    }
}

```

```
        switch (ch)
        {
            case 1: rear();
                    break;

            case 2: front();
                    break;

            case 3: peep();
                    break;

            case 4: exit(0);

            default: printf("\n Please Enter Valid Choice for
operation");
        }
    }while(ch<5);

}
void front()
{
    if(f== -1)
    {
        printf("\n Queue is empty");
    }
    else if(f==r)
    {
        printf("\n%d is deleted from the queue",a[f]);
        f=r-1;
    }
    else
    {
        printf("\n%d is deleted from the queue",a[f]);
        f++;
        if(f==3)
            f=0;
    }
}

void rear()
```

```
{
    r++;
    if(r==0)
        f=0;
    else if (r==3)
        if(f>0)
            r=0;
        else
        {
            printf("\n Queue Overflow");
            exit(0);
        }
    else if(f==r)
    {
        printf("\n Queue Overflow");
        exit(0);
    }

    printf("Enter ur value:");
    scanf("%d",&a[r]);
}

void peep()
{
    if(f== -1)
        printf("\nQueue is empty");
    else
        printf("\n%d is value of %d front pointer and %d is value
of %d rear pointer of queue",a[f],f,a[r],r);
}
```

Output:-

```
1.Insertion
2.Deletion
3.Peep
4.Exit
Enter ur choice for above operations on Queue:1
Enter ur value:12

1.Insertion
2.Deletion
3.Peep
4.Exit
Enter ur choice for above operations on Queue:3
12 is value of 0 front pointer and 12 is value of 0 rear pointer of queue
```

```
1.Insertion
2.Deletion
3.Peep
4.Exit
Enter ur choice for above operations on Queue:2

12 is deleted from the queue
1.Insertion
2.Deletion
3.Peep
4.Exit
Enter ur choice for above operations on Queue:2

Queue is empty
1.Insertion
2.Deletion
3.Peep
4.Exit
Enter ur choice for above operations on Queue:4
```


Practical-4**26.Create Linked list.**

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
}*head=0,*tem,*new_node,*last;

void display();
void create(int x);
void main()
{
    create(18);
    create(100);
    create(00);
    create(134);
    display();
    printf("\n\n");
    getch();
}

void create(int x)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=x;
    new_node->next=0;
    if(head==0)
        head=new_node;
    else
        tem->next=new_node;
    tem=new_node;
}

void display()
{
    struct node *ptr;
    ptr=head;
```

```
while(ptr!=0)
{
    printf("%d\t",ptr->data);
    ptr=ptr->next;
}
}
```

Output:



```
18      100      0      134
```

27.Insert a node at beginning.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
}*head=0,*tem,*new_node,*last;

void display();
void insert_beg();
void create(int x);
void main()
{
    create(18);
    create(100);
    create(00);
    create(134);
    display();
    printf("\nInserting new node at beginning ==>");
    insert_beg(44);
    display();
    printf("\n\n");
    getch();
}
```

```
}

void create(int x)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=x;
    new_node->next=0;
    if(head==0)
        head=new_node;
    else
        tem->next=new_node;
    tem=new_node;
}

void insert_beg(int x)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=x;
    new_node->next=0;
    if(head==0)
        head=new_node;
    else
    {
        new_node->next=head;
        head=new_node;
    }
}

void display()
{
    struct node *ptr;
    ptr=head;
    while(ptr!=0)
    {
        printf(" %d\t",ptr->data);
        ptr=ptr->next;
    }
}
```

```
18      100      0      134
Inserting new node at beginning ==> 44  18      100      0      134
```

Output:**28.Insert a node at end.**

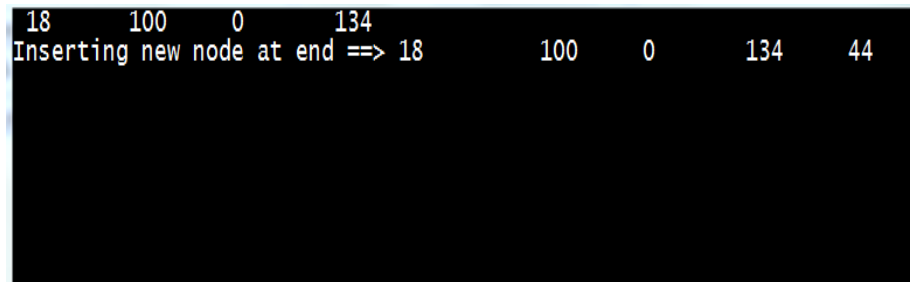
```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
}*head=0,*tem,*new_node,*last;

void display();
void insert_beg();
void insert_end(int x);
void create(int x);
void main()
{
    create(18);
    create(100);
    create(00);
    create(134);
    display();
    printf("\nInserting new node at end ==>");
    insert_end(44);
    display();
    printf("\n\n");
    getch();
}

void create(int x)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=x;
    new_node->next=0;
    if(head==0)
        head=new_node;
    else
        tem->next=new_node;
    tem=new_node;
}
```

```
void insert_end(int x)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=x;
    new_node->next=0;
    struct node *ptr=head;
    while(ptr->next!=0)
    {
        ptr=ptr->next;
    }
    ptr->next=new_node;
}

void display()
{
    struct node *ptr;
    ptr=head;
    while(ptr!=0)
    {
        printf(" %d\t",ptr->data);
        ptr=ptr->next;
    }
}
```

Output:

```
18 100 0 134
Inserting new node at end ==> 18 100 0 134 44
```

29.Insert a node at specific position.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
```

```
}*head=0,*tem,*new_node,*last;

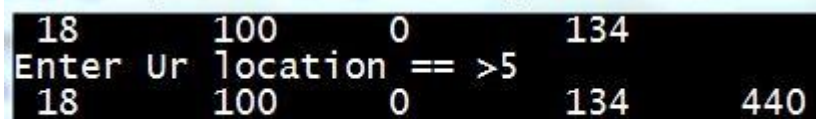
int input()
{
    char x[10];
    return atoi(gets(x));
}
int display();
int insert_loc();
void create(int x);
void main()
{
    create(18);
    create(100);
    create(00);
    create(134);
    display(1);
    if(insert_loc(440)==1)
        display(1);
    else
        printf("\nHello");
    printf("\n\n");
    getch();
}

void create(int x)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=x;
    new_node->next=0;
    if(head==0)
        head=new_node;
    else
        tem->next=new_node;
    tem=new_node;
}

int insert_loc(int x)
{
    struct node *ptr,*past;
```

```
ptr=head;
int c=0,f=1,lo;
printf("\nEnter Ur location == >");
lo=input();
if(lo<=0)
{
    printf("Invalid location....");
    return 0;
}
new_node=(struct node *)malloc(sizeof(struct node));
new_node->data=x;
new_node->next=0;
if(lo==1)
{
    new_node->next=head;
    head=new_node;
}
else
{
    c=display(2);
    if(lo>c+1)
    {
        printf("\nInvalid Location....");
        return 0;
    }
    else
    {
        while(f<lo-1)
        {
            ptr=ptr->next;
            f++;
        }
        new_node->next=ptr->next;
        ptr->next=new_node;
        if(new_node->next==0)
            tem=new_node;
    }
}
return 1;
}
```

```
int display(int x)
{
    struct node *ptr;
    ptr=head;
    int c=0;
    while(ptr!=0)
    {
        c++;
        if(x==1)
            printf(" %d\t",ptr->data);
        ptr=ptr->next;
    }
    return c;
}
```

Output:

The screenshot shows the output of the program. It displays the values of the linked list nodes: 18, 100, 0, 134. Below this, it prompts 'Enter Ur location == >5'. The output then shows the values of the nodes starting from the 5th location: 18, 100, 0, 134, and 440.

30.Delete a node from beginning.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
}*head=0,*tem,*new_node,*last;

int input()
{
    char x[10];
    return atoi(gets(x));
}
int display();
int insert_loc();
```



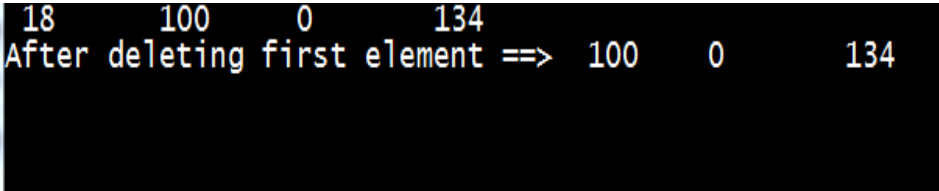
```
void del_first();
void create(int x);
void main()
{
    create(18);
    create(100);
    create(00);
    create(134);
    display(1);
    del_first();
    printf("\nAfter deleting first element ==> ");
    display(1);
    printf("\n\n");
    getch();
}

void create(int x)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=x;
    new_node->next=0;
    if(head==0)
        head=new_node;
    else
        tem->next=new_node;
    tem=new_node;
}

void del_first()
{
    struct node *ptr=head;
    if(head==0)
        printf("\nUnderflow.....\n");
    else
    {
        head=head->next;
        free(ptr);
    }
}
```

```
int display(int x)
{
    struct node *ptr;
    ptr=head;
    int c=0;
    while(ptr!=0)
    {
        c++;
        if(x==1)
            printf(" %d\t",ptr->data);
        ptr=ptr->next;
    }
    return c;
}
```

Output :



```
18 100 0 134
After deleting first element ==> 100 0 134
```

31.Delete a node from end.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
}*head=0,*tem,*new_node,*last;

int input()
{
    char x[10];
    return atoi(gets(x));
}

int display();
void del_last();
```

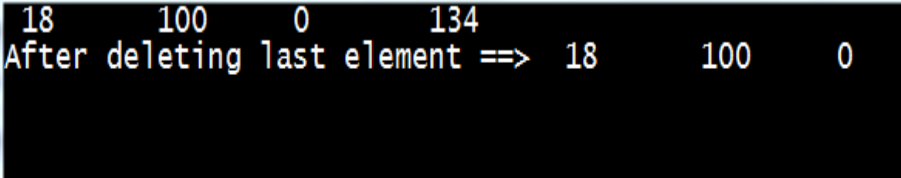
```
int insert_loc();
void create(int x);
void main()
{
    create(18);
    create(100);
    create(00);
    create(134);
    display(1);
    del_last();
    printf("\nAfter deleting last element ==> ");
    display(1);
    printf("\n\n");
    getch();
}
```

```
void del_last()
{
    struct node *ptr;
    ptr=head;
    int c=0,f=1;
    if(head==0)
        printf("\nUnderflow.....\n");
    else
    {
        if(head->next==0)
            head=0;
        else
        {
            c=display(5);
            while(f<c-1)
            {
                ptr=ptr->next;
                f++;
            }
            ptr->next=ptr->next->next;
            tem=ptr;
            ptr=ptr->next;
        }
        free(ptr);
    }
```

```
    }
}
void create(int x)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=x;
    new_node->next=0;
    if(head==0)
        head=new_node;
    else
        tem->next=new_node;
    tem=new_node;
}

int display(int x)
{
    struct node *ptr;
    ptr=head;
    int c=0;
    while(ptr!=0)
    {
        c++;
        if(x==1)
            printf(" %d\t",ptr->data);
        ptr=ptr->next;
    }
    return c;
}
```

Output :



```
18    100    0    134
After deleting last element ==> 18    100    0
```

32.Delete a node from specific position.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
```

```
{
    int data;
    struct node *next;
}*head=0,*tem,*new_node,*last;

int input()
{
    char x[10];
    return atoi(gets(x));
}
int display();
int del_spe();
void create(int x);
void main()
{
    create(18);
    create(100);
    create(00);
    create(134);
    display(1);
    if(del_spe()!=0)
        display(1);
    printf("\n\n");
    getch();
}

int del_spe()
{
    int c=0,f=1,lo;
    struct node *ptr=head,*pre;
    printf("\nEnter Location ==> ");
    scanf("%d",&lo);
    if(lo==1)
    {
        head=head->next;
        free(ptr);
        return 2;
    }
    else
    {
```

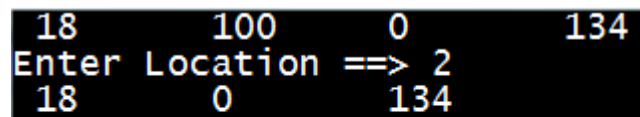
```
        c=display(5);
        if(lo>c)
        {
            printf("\nInvalid location");
            return 0;
        }
        else
        {
            while(f<lo-1)
            {
                ptr=ptr->next;
                f++;
                printf("\n\n\n\n\nHello");
            }
            pre=ptr;
            ptr=ptr->next;
            pre->next=ptr->next;
            ptr->next=0;
            free(ptr);
            return 8;
        }
    }
}

void create(int x)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=x;
    new_node->next=0;
    if(head==0)
        head=new_node;
    else
        tem->next=new_node;
    tem=new_node;
}

int display(int x)
{
    struct node *ptr;
    ptr=head;
    int c=0;
```

```
while(ptr!=0)
{
    c++;
    if(x==1)
        printf(" %d\t",ptr->data);
    ptr=ptr->next;
}
return c;
}
```

Output :



```
18      100      0      134
Enter Location ==> 2
18      0      134
```

33.Search a node in linked list.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
}*head=0,*tem,*new_node,*last;

int input()
{
    char x[10];
    return atoi(gets(x));
}

void search();
int display();
int del_spe();
void create(int x);
void main()
{
    create(18);
```

```
        create(100);
        create(00);
        create(134);
        display(1);
        printf("\n\n");
        search();
        getch();
    }

void search()
{
    int info,c=0,f=0;
    struct node *ptr=head;
    if(head==0)
        printf("\nEmpty Linked List\n");
    else
    {
        printf("\nEnter Ur data: ") ;
        scanf("%d",&info);
        while(ptr!=0)
        {
            c++;
            if(ptr->data==info)
            {
                f++;
                printf("\n %d is found at %d
node",info,c);

                break;
            }
            ptr=ptr->next;
        }
        if(f==0)
            printf("\nSorry!!No such value in linked
list..\n");
    }
}

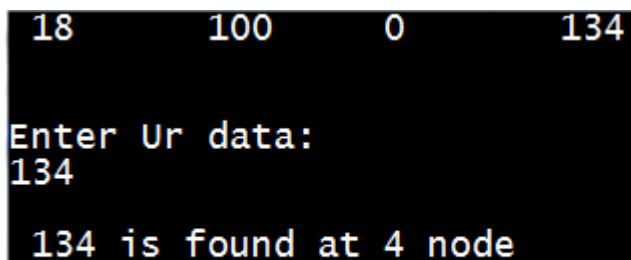
void create(int x)
{
    new_node=(struct node *)malloc(sizeof(struct node));
```



```
new_node->data=x;
new_node->next=0;
if(head==0)
    head=new_node;
else
    tem->next=new_node;
tem=new_node;
}

int display(int x)
{
    struct node *ptr;
    ptr=head;
    int c=0;
    while(ptr!=0)
    {
        c++;
        if(x==1)
            printf(" %d\t",ptr->data);
        ptr=ptr->next;
    }
    return c;
}
```

Output :



```
18      100      0      134

Enter Ur data:
134

134 is found at 4 node
```

34.Count number of nodes in linked list.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
```

```
*head=0,*tem,*new_node,*last;

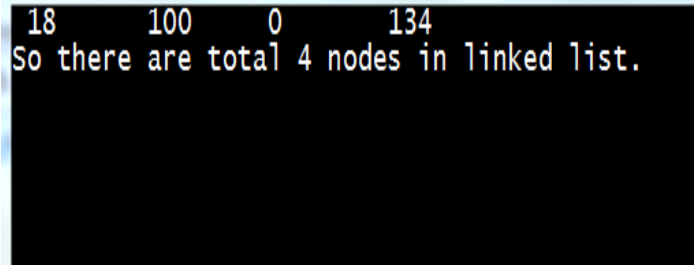
void search();
int display();
int del_spe();
void create(int x);
void main()
{
    int c;
    create(18);
    create(100);
    create(00);
    create(134);
    c=display(1);
    printf("\nSo there are total %d nodes in linked list.",c);
    printf("\n\n");
    getch();
}

void create(int x)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=x;
    new_node->next=0;
    if(head==0)
        head=new_node;
    else
        tem->next=new_node;
    tem=new_node;
}

int display(int x)
{
    struct node *ptr;
    ptr=head;
    int c=0;
    while(ptr!=0)
    {
        c++;
        if(x==1)
```

```
        printf(" %d\t",ptr->data);  
        ptr=ptr->next;  
    }  
    return c;  
}
```

Output :



```
18    100    0    134  
So there are total 4 nodes in linked list.
```

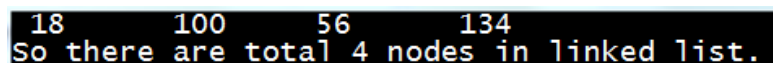
35. Count no of nodes in circular linked list.

```
#include<stdio.h>  
#include<conio.h>  
#include<stdlib.h>  
  
struct node  
{  
    int data;  
    struct node *next;  
}*head=0,*tem,*new_node,*last;  
  
void main()  
{  
    int c;  
    create(18);  
    create(100);  
    create(56);  
    create(134);  
    c=display(1);  
    printf("\nSo there are total %d nodes in linked list.",c);  
    printf("\n\n");  
    getch();  
}  
  
void create(int x)  
{  
    new_node=(struct node *)malloc(sizeof(struct node));
```

```
new_node->data=x;
if(head==0)
    head=new_node;
else
    tem->next=new_node;
tem=new_node;
tem->next=head;
}

int display(int x)
{
    struct node *ptr;
    ptr=head;
    int c=0;
    do
    {
        c++;
        if(x==1)
            printf(" %d\t",ptr->data);
        ptr=ptr->next;
    }while(ptr!=head);
    return c;
}
```

} Output:



```
18    100    56    134
So there are total 4 nodes in linked list.
```

36.Insert a node at beginning in doubly linked list.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
    struct node *prev;
```

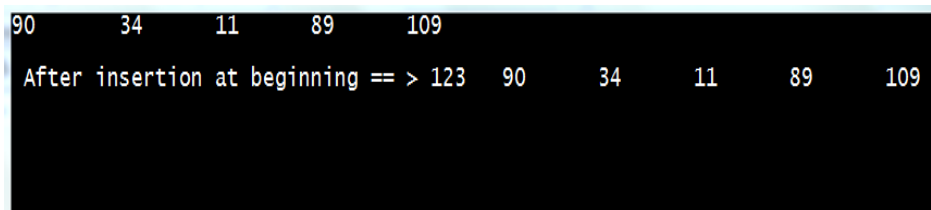
```
*head=0,*new_node,*tem;
void create(int a)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=a;
    new_node->next=0;
    new_node->prev=0;
    if(head==0)
    {
        head=new_node;
        tem=head;
    }
    else
    {
        tem->next=new_node;
        new_node->prev=tem;
        tem=new_node;
    }
}

void insert_beg(int a)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=a;
    if(head==0)
    {
        new_node->next=head;
        head->prev=new_node;
        head=new_node;
    }
    else
    {
        new_node->next=head;
        head->prev=new_node;
        head=new_node;
    }
}

int display(int a)
{

```

```
struct node *ptr;
ptr=head;
int c=0;
if(head==0)
    printf("\nEmpty linked List.....\n");
else
{
    do
    {
        if(a==1)
            printf("%d\t",ptr->data);
        c++;
        ptr=ptr->next;
    }while(ptr!=0);
    if(a==2)
        printf(" You have created %d nodes...\n",c);
    printf("\n");
}
return c;
}
void main()
{
    create(90);
    create(34);
    create(11);
    create(89);
    create(109);
    display(1);
    insert_beg(123);
    printf("\n After insertion at beginning == > ");
    display(1);
    getch();
}
```



```
90      34      11      89      109
After insertion at beginning == > 123  90      34      11      89      109
```

Output:**37. Insert a node at end in doubly linked list.**

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

struct node
{
    int data;
    struct node *next;
    struct node *prev;
} *head=0,*new_node,*tem;

void create(int a)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=a;
    new_node->next=0;
    new_node->prev=0;
    if(head==0)
    {
        head=new_node;
        tem=head;
    }
    else
    {
        tem->next=new_node;
        new_node->prev=tem;
        tem=new_node;
    }
}

void insert_end(int a)
{
    struct node * ptr=head;
    while(ptr->next!=0)
```

```
    {
        ptr=ptr->next;
    }
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=a;
    ptr->next=new_node;
    new_node->prev=ptr;
    new_node->next=0;
}
```

```
int display(int a)
{
    struct node *ptr;
    ptr=head;
    int c=0;
    if(head==0)
        printf("\nEmpty linked List.....\n");
    else
    {
        do
        {
            if(a==1)
                printf("%d\t",ptr->data);
            c++;
            ptr=ptr->next;
        } while(ptr!=0);
        if(a==2)
            printf(" You have created %d nodes...\n",c);
        printf("\n");
    }
    return c;
}
```

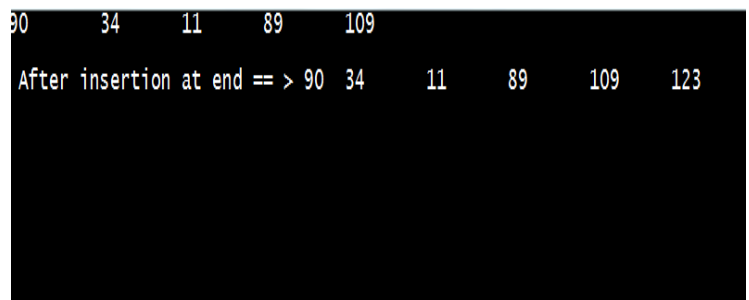
```
void main()
{
    create(90);
    create(34);
    create(11);
}
```



```

        create(89);
        create(109);
        display(1);
        insert_end(123);
        printf("\n After insertion at end == > ");
        display(1);
        getch();
    }

```

Output:


```

90      34      11      89      109
After insertion at end == > 90  34      11      89      109      123

```

38.Delete a node from beginning in doubly linked list.

```

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

struct node
{
    int data;
    struct node *next;
    struct node *prev;
} *head=0,*new_node,*tem;

void create(int a)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=a;
    new_node->next=0;
    new_node->prev=0;
    if(head==0)
    {
        head=new_node;
        tem=head;
    }
}

```

```
        }
        else
        {
            tem->next=new_node;
            new_node->prev=tem;
            tem=new_node;
        }

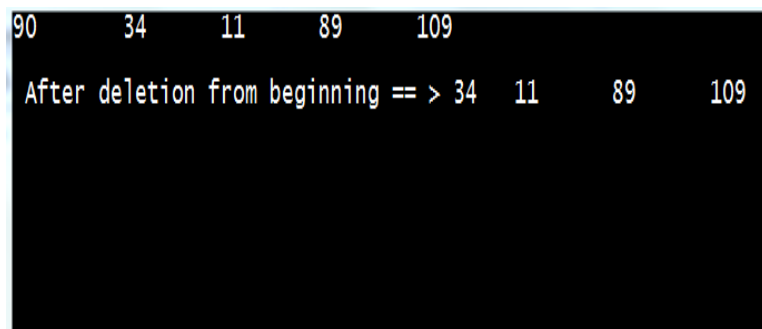
    }

void del_first()
{
    struct node *ptr=head;
    if(head==0)
        printf("\nUnderflow.....\n");
    else
    {
        head=head->next;
        free(ptr);
    }
}

int display(int a)
{
    struct node *ptr;
    ptr=head;
    int c=0;
    if(head==0)
        printf("\nEmpty linked List.....\n");
    else
    {
        do
        {
            if(a==1)
                printf("%d\t",ptr->data);
            c++;
            ptr=ptr->next;
        }
    }
}
```

```
        }while(ptr!=0);
        if(a==2)
            printf(" You have created %d nodes...\n",c);
        printf("\n");
    }
    return c;
}

void main()
{
    create(90);
    create(34);
    create(11);
    create(89);
    create(109);
    display(1);
    del_first();
    printf("\n After deletion from beginning == > ");
    display(1);
    getch();
}
```

Output:

```
90      34      11      89      109
After deletion from beginning == > 34      11      89      109
```

39.Delete a node from end in doubly linked list.

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

```
struct node
```

```
{
    int data;
    struct node *next;
    struct node *prev;
} *head=0,*new_node,*tem;

void create(int a)
{
    new_node=(struct node *)malloc(sizeof(struct node));
    new_node->data=a;
    new_node->next=0;
    new_node->prev=0;
    if(head==0)
    {
        head=new_node;
        tem=head;
    }
    else
    {
        tem->next=new_node;
        new_node->prev=tem;
        tem=new_node;
    }
}

void del_end()
{
    struct node *ptr;
    ptr=tem;
    if(head==0)
        printf("\nUnderflow.....\n");
    else
    {
        if(head->next==0)
            head=0;
        else
        {
            ptr->prev->next=ptr->next;
            tem=ptr->prev;
        }
    }
}
```

```
        free(ptr);
    }
}

int display(int a)
{
    struct node *ptr;
    ptr=head;
    int c=0;
    if(head==0)
        printf("\nEmpty linked List.....\n");
    else
    {
        do
        {
            if(a==1)
                printf("%d\t",ptr->data);
            c++;
            ptr=ptr->next;
        }while(ptr!=0);
        if(a==2)
            printf(" You have created %d nodes...\n",c);
        printf("\n");
    }
    return c;
}

void main()
{
    create(90);
    create(34);
    create(11);
    create(89);
    create(109);
    display(1);
    del_end();
    printf("\n After deletion from end == > ");
    display(1);
}
```

Sub name: Data structure

Enrollment no: 196170307149

```
    getch();  
}
```

Output:

```
90      34      11      89      109  
After deletion from end == > 90      34      11      89
```

Practical-5**40.Bubble sort**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[50],n,i,j,c=0;
    printf("\n How many elements you want in your array?? ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("\nEnter For a[%d]=",i+1);
        scanf("%d",&a[i]);
    }

    for(i=1;i<n;i++)
    {
        c=0;
        for(j=0;j<n-i;j++)
        {
            if(a[j]>a[j+1])
            {
                a[j]=a[j]+a[j+1];
                a[j+1]=a[j]-a[j+1];
                a[j]=a[j]-a[j+1];
                c=1;
            }
        }
        if(c==0)
            break;
    }

    for(i=0;i<n;i++)
    {
        printf("\na[%d]=%d",i+1,a[i]);
    }
}
```

Output:-

```
How many elements you want in your array?? 6
Enter For a[1]=11
Enter For a[2]=9
Enter For a[3]=10
Enter For a[4]=23
Enter For a[5]=4
Enter For a[6]=12
a[1]=4
a[2]=9
a[3]=10
a[4]=11
a[5]=12
a[6]=23
```

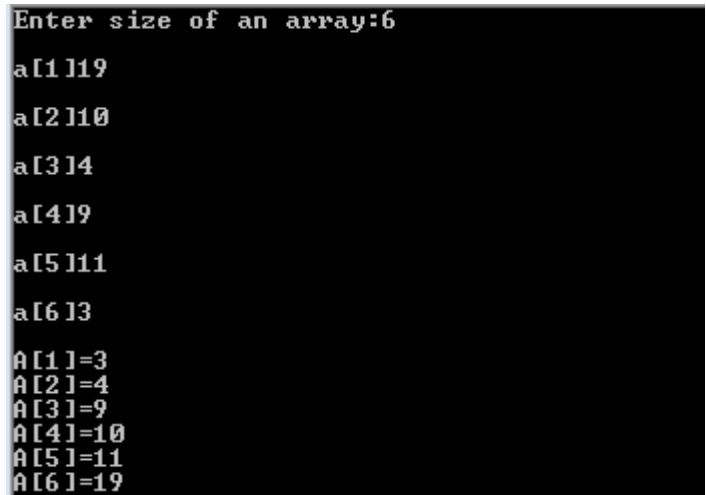
41.Selection sort

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

void main()
{
    int a[50],n,i,j;
    printf("Enter size of an array:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("\na[%d]",i+1);
        scanf("%d",&a[i]);
    }
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(a[i]>=a[j])
            {
                a[i]=a[i]+a[j];
                a[j]=a[i]-a[j];
                a[i]=a[i]-a[j];
            }
        }
    }
    for(i=0;i<n;i++)
```



```
{
    printf("\nA[%d]=%d",i+1,a[i]);
}
}
```

Output:-

```
Enter size of an array:6
a[1]19
a[2]10
a[3]14
a[4]19
a[5]11
a[6]13
A[1]=3
A[2]=4
A[3]=9
A[4]=10
A[5]=11
A[6]=19
```

42.Quick sort

```
#include<stdio.h>
void quickSort(int[], int, int);
int partition(int[], int, int);
void swap(int*, int*);

int main()
{
    int n;
    printf("Enter Array Size\n");
    scanf("%d",&n);
    int arr[n],i;
    printf("Enter Array Elements\n");
    for(i=0;i<n;i++)
        scanf("%d",&arr[i]);
    quickSort(arr,0,n-1);
    printf("\nAfter the QuickSort\n");
    for(i=0;i<n;i++)
        printf("%d ",arr[i]);
    printf("\n");
    return 0;
}
```

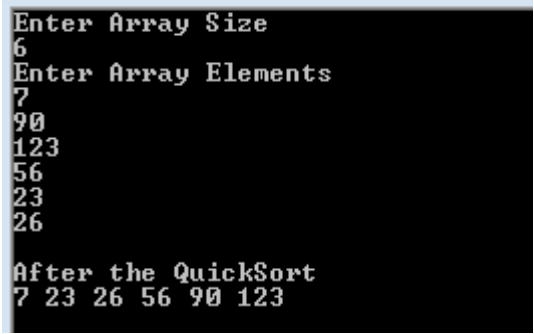
```
}

void quickSort(int arr[], int start, int end)
{
    if(start < end)
    {
        int pIndex = partition(arr, start, end);
        quickSort(arr, start, pIndex-1);
        quickSort(arr, pIndex+1, end);
    }
}

int partition(int arr[], int start, int end)
{
    int pIndex = start;
    int pivot = arr[end];
    int i;
    for(i = start; i < end; i++)
    {
        if(arr[i] < pivot)
        {
            swap(&arr[i], &arr[pIndex]);
            pIndex++;
        }
    }
    swap(&arr[end], &arr[pIndex]);
    /* for(i=start;i<=end;i++)
        printf("\t %d",arr[i]);
    printf("\n"); */
    return pIndex;
}

void swap(int *x, int *y)
{
    int t = *x;
    *x = *y;
    *y = t;
}
```

```
}
```



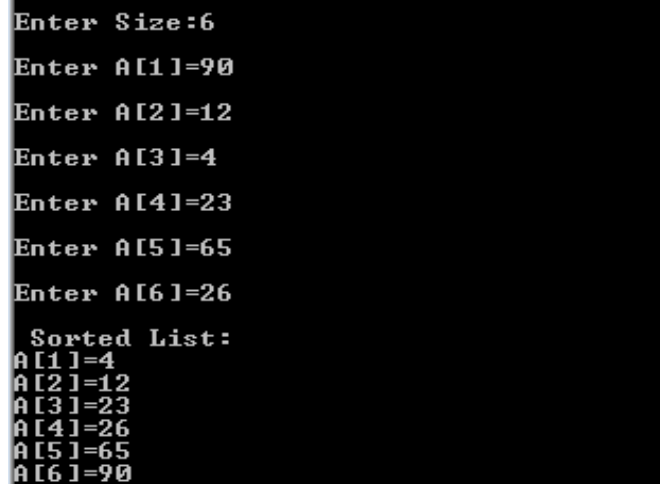
A screenshot of a terminal window with a black background and white text. The text shows the execution of a C program. It starts with the prompt 'Enter Array Size' followed by the input '6'. Then it prompts 'Enter Array Elements' followed by the inputs '7', '90', '123', '56', '23', and '26' on separate lines. Finally, it displays the output 'After the QuickSort' followed by the sorted array '7 23 26 56 90 123'.

43.Insertion sort

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

void main()
{
    int a[50],n,i,j,keyindex,b,t;
    printf("\nEnter Size:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("\nEnter A[%d]=",i+1);
        scanf("%d",&a[i]);
    }
    keyindex=1;
    for(i=1;i<n;i++)
    {
        for(j=0;j<keyindex;j++)
        {
            if(a[j]>a[keyindex])
            {
                t=a[keyindex];
                for(b=keyindex;b>j;b--)
                {
                    a[b]=a[b-1];
                }
                a[b]=t;
            }
        }
    }
}
```

```
    }  
    keyindex++;  
}  
printf("\n Sorted List:");  
for(i=0;i<n;i++)  
{  
    printf("\nA[%d]=%d",i+1,a[i]);  
}  
}
```

Output:-

```
Enter Size:6  
Enter A[1]=90  
Enter A[2]=12  
Enter A[3]=4  
Enter A[4]=23  
Enter A[5]=65  
Enter A[6]=26  
  
Sorted List:  
A[1]=4  
A[2]=12  
A[3]=23  
A[4]=26  
A[5]=65  
A[6]=90
```

44.Merge sort.

```
#include <stdio.h>  
#include <stdlib.h>  
void merge(int arr[], int l, int m, int r)  
{  
    int i, j, k;  
    int n1 = m - l + 1;  
    int n2 = r - m;  
    int L[n1], R[n2];  
    for (i = 0; i < n1; i++)  
        L[i] = arr[l + i];  
    for (j = 0; j < n2; j++)  
        R[j] = arr[m + 1 + j];  
    i = 0;  
    j = 0;  
    k = l;  
    while (i < n1 && j < n2)
```

```
{
    if (L[i] <= R[j])
    {
        arr[k] = L[i];
        i++;
    }
    else
    {
        arr[k] = R[j];
        j++;
    }
    k++;
}

while (i < n1)
{
    arr[k] = L[i];
    i++;
    k++;
}
while (j < n2)
{
    arr[k] = R[j];
    j++;
    k++;
}
}

void mergeSort(int arr[], int l, int r)
{
    if (l < r)
    {
        int m = l + (r - l) / 2;
        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);
        merge(arr, l, m, r);
    }
}

int main()
{
    int arr[50],n,i;
```

```

printf("\nEnter Size of array:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("\nEnter:");
scanf("%d",&arr[i]);
}
mergeSort(arr, 0, n - 1);
printf("\nSorted array is \n");
for (i = 0; i < n; i++)
    printf("%d ",arr[i]);
printf("\n");
return 0;
}

```

```

Enter Size of array:6
Enter:34
Enter:56
Enter:32
Enter:12
Enter:22
Enter:46
Sorted array is
12 22 32 34 46 56

```

45.Solve hash table example using division method, mid square method.

32, 53, 22, 92, 17, 34, 24, 37, and 56 into a hash table of size $M = 10$.

Key	Division method ($\text{key} \% (m-1)$)	MidSquare Method (key^2)
32	$32 \% 10 = 2$	$(32)^2 = 02$
53	$53 \% 10 = 3$	$(53)^2 = 80$
22	$22 \% 10 = 2$	$(22)^2 = 8$
92	$92 \% 10 = 2$	$(92)^2 = 46$
17	$17 \% 10 = 7$	$(17)^2 = 8$
34	$34 \% 10 = 4$	$(34)^2 = 15$

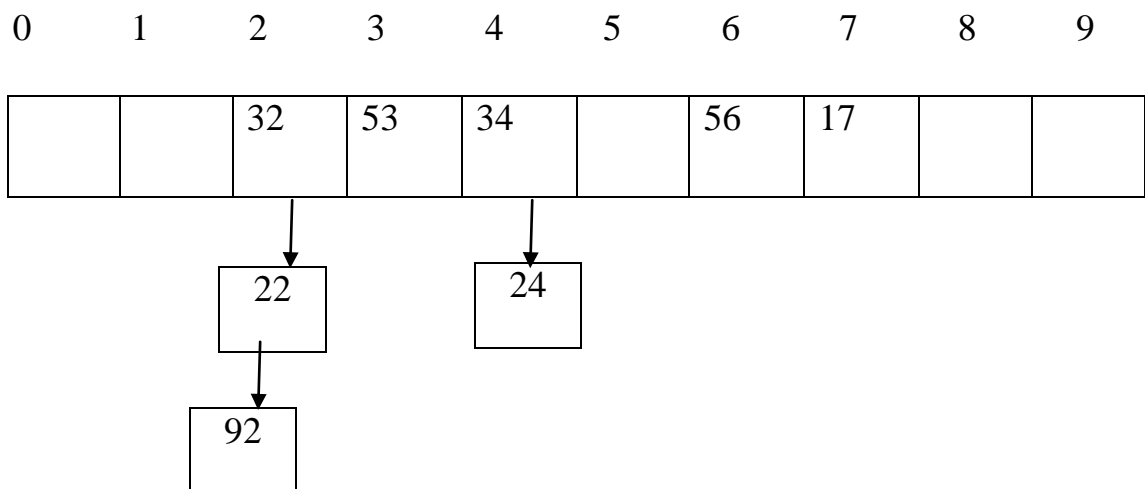
24	$24 \% 10 = 4$	$(24)^2 = 7$
56	$56 \% 10 = 6$	$(56)^2 = 13$

Mid Square Method:

In mid square method, first we have to take square of the key and after that we take middle value as our index.

Division Method:

In this method, first we have to take mode of the key with the size of the array. And if the mode of the keys have same value than linked list will generate on that index.



Practical-6**46. Construct Binary Search Tree, with In-Order, Pre-Order, Post-Order Traversal.**

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

struct node
{
    int data;
    struct node *left, *right;
};

struct node *newNode(int item)
{
    struct node *temp = (struct node *)malloc(sizeof(struct node));
    temp->data = item;
    temp->left = temp->right = NULL;
    return temp;
}

void preorder(struct node *ptr)
{
    if(ptr)
    {
        printf("\t%d", ptr->data);
        preorder(ptr->left);
        preorder(ptr->right);
    }
}

void inorder(struct node *root)
{
    if (root != NULL)
    {
        inorder(root->left);
        printf("%d \t", root->data);
        inorder(root->right);
    }
}
```



```
void postorder(struct node *ptr)
{
    if(ptr)
    {
        postorder(ptr->left);
        postorder(ptr->right);
        printf("\t%d",ptr->data);
    }
}

struct node* insert(struct node* node, int data)
{
    if (node == NULL)
        return newNode(data);
    if (data <= node->data)
        node->left = insert(node->left, data);
    else if (data > node->data)
        node->right = insert(node->right, data);
    return node;
}

int main()
{
    struct node *root = NULL;
    root = insert(root, 8);
    insert(root, 13);
    insert(root, 4);
    insert(root, 16);
    insert(root, 0);
    insert(root, 24);
    insert(root, 8);
    printf("\n\nInorder:\n(Left,Root,Right):");
    inorder(root);
    printf("\n\nPostorder:\n(Left,Right,Root):");
    postorder(root);
    printf("\n\nPreorder:\n(Root,Left,Right):");
    preorder(root);
    printf("\n\n");
}
```

```

    return 0;
}

```

```

Inorder:
<Left,Root,Right>: 0    4    8    8    13    16    24

Postorder:
<Left,Right,Root>:    0    8    4    24    16    13    8

Preorder:
<Root,Left,Right>:    8    4    0    8    13    16    24

```

47. Algorithm of Binary search Tree

Algorithm

1. Create a new BST node and assign values to it.
2. insert(node, key)
 - i) If root == NULL,
 - return the new node to the calling function.
 - ii) if root=>data < key
 - call the insert function with root=>right and assign the return value in root=>right.
 - root->right = insert(root=>right, key)
 - iii) if root=>data > key
 - call the insert function with root->left and assign the return value in root=>left.
 - root->left = insert(root=>left, key)
3. Finally, return the original root pointer to the calling function.

48. Algorithm of Pre-Order Traversal

Algorithm

Until all nodes are traversed

Step 1 – Visit root node.

Step 2 – Recursively traverse left subtree.

Step 3 – Recursively traverse right subtree.

49. Algorithm of In-Order Traversal

Algorithm

Until all nodes are traversed

Step 1 – Recursively traverse left subtree.

Step 2 – Visit root node.

Step 3 – Recursively traverse right subtree.

50.Algorithm of Post-Order Traversal

Algorithm

Until all nodes are traversed

Step 1 – Recursively traverse left subtree.

Step 2 – Recursively traverse right subtree.

Step 3 – Visit root node.