

1. (a) IMPLEMENTATION OF STACK USING ARRAY

Program:-

```
#include<iostream>
using namespace std;
#define SIZE 10
static int top=-1;
class stack
{
private:
    int ar[SIZE];
public:
    void push(int item);
    void pop();
    void peep();
};
void stack::push(int item)
{
    if(top==SIZE-1)
        cout<<"\nThe Stack is Full!!!";
    else
    {
        ar[++top]=item;
        cout<<"\nElement succesfully pushed in the Stack!!!";
    }
}
void stack::pop()
{
    if(top<0)
        cout<<"\nStack Under flow!!!";
    else
    {
        top--;
        cout<<"\nElement sucessfully popped from the Stack!!!";
    }
}
void stack::peep()
{
    if(top<0)
        cout<<"\nThe Stack is Empty it cannot be Peeped!!!";
    else
        for(int i=top;i>=0;i--)
            cout<<ar[i]<<" ";
}
```

```

int main()
{
    char choice;
    int ch,num;
    stack ob;
    do
    {
        cout<<"\n\n\t\t\tS T A C K   O P E R A T I O N S";
        cout<<"\n\t\t\t-----";
        cout<<"\n\n1.PUSH";
        cout<<"\n2.POP";
        cout<<"\n3.PEEP";
        cout<<"\n4.EXIT";
        cout<<"\n\nEnter your choice:";
        cin>>ch;
        switch(ch)
        {
            case 1: cout<<"\nEnter the Element you want to Push:";
                    cin>>num;
                    ob.push(num);
                    break;
            case 2: ob.pop(); break;
            case 3: ob.peep(); break;
            case 4: ;
            default: cout<<"\nPlease Enter a Valid Choice(1-4)!!!";
        }
        cout<<"\nDo you want to Continue(Y/N):";
        cin>>choice;
    }while(choice=='y' || choice=='Y');
}

```

OUTPUT:

STACK OPERATIONS

1.PUSH

2.POP

3.PEEP

4.EXIT

Enter your choice:1

Enter the Element you want to Push:12

Element successfully pushed in the stack!!!

Do you want to Continue(Y/N);y

STACK OPERATIONS

1.PUSH

2.POP

3.PEEP

4.EXIT

Enter your choice:2

Element successfyllly popped from the Stack!!!

Do you want to Continue(Y/N);y

STACK OPERATIONS

1.PUSH

2.POP

3.PEEP

4.EXIT

Enter your choice:3

The Stack is Empty it cannot be Peeped!!!

Do you want to Continue(Y/N);y

STACK OPERATIONS

1.PUSH

2.POP

3.PEEP

4.EXIT

Enter your choice:4

1. (b) IMPLEMENTATION OF QUEUE USING ARRAY

Program:-

```
#include<iostream>
using namespace std;
#define SIZE 10
static int front=0;
static int end=-1;
class queue
{
private:
    int ar[SIZE];
public:
    void insert(int item);
    void delque();
    void viewque();
};
void queue::insert(int item)
{
    if(end==SIZE-1)
        cout<<"\nThe Queue is Full!!!";
    else
    { ar[++end]=item;
      cout<<"\nElement succesfully inserted in the Queue!!!";
    }
}
void queue::delque()
{
    if(end<0)
        cout<<"\nQueue Under flow!!!";
    else
    { front++;
      cout<<"\nElement sucessfully deleted from the Queue!!!";
    }
}
void queue::viewque()
{ if(end<0)
  cout<<"\nThe Queue is Empty it cannot be Viewed!!!";
  else
  for(int i=front;i<=end;i++)
      cout<<ar[i]<<" ";
}
```

```

int main()
{
    char choice;
    int ch,num;
    queue ob;
    do
    {
        clrscr();
        cout<<"\n\n\t\tQ U E U E   O P E R A T I O N S";
        cout<<"\n\t\t\t-----";
        cout<<"\n\n1.INSERT";
        cout<<"\n2.DELETE";
        cout<<"\n3.DISPLAY";
        cout<<"\n4.EXIT";
        cout<<"\n\nEnter your choice:";
        cin>>ch;
        switch(ch)
        {
            case 1: cout<<"\n\nEnter the Element you want to Push:";
                    cin>>num;
                    ob.insert(num);
                    break;
            case 2: ob.delque(); break;
            case 3: ob.viewque(); break;
            case 4: exit(0);
            default: cout<<"\nPlease Enter a Valid Choice(1-4)!!!";
        }
        cout<<"\n\nDo you want to Continue(Y/N):";
        cin>>choice;
    }while(choice=='y' || choice=='Y');
}

```

OUTPUT:-

Q U E U E O P E R A T I O N S

```

1.INSERT
2.DELETE
3.DISPLAY
4.EXIT

```

Enter your choice:1

Enter the Element you want to Push:25

Element succesfully inserted in the Queue!!!

Do you want to Continue(Y/N):Y

QUEUE OPERATIONS

- 1.INSERT
- 2.DELETE
- 3.DISPLAY
- 4.EXIT

Enter your choice:1

Enter the Element you want to Push:26

Element succesfully inserted in the Queue!!!

Do you want to Continue(Y/N):Y

QUEUE OPERATIONS

- 1.INSERT
- 2.DELETE
- 3.DISPLAY
- 4.EXIT

Enter your choice:2

Element sucessfully deleted from the Queue!!!

Do you want to Continue(Y/N):Y

QUEUE OPERATIONS

- 1.INSERT
- 2.DELETE
- 3.DISPLAY
- 4.EXIT

Enter your choice:3

26

Do you want to Continue(Y/N):Y

QUEUE OPERATIONS

- 1.INSERT
- 2.DELETE
- 3.DISPLAY
- 4.EXIT

Enter your choice:4

1. (C) IMPLEMENTATION OF INFIX TO POSTFIX EXPRESSION USING ARRAY

Program:-

```
#include<iostream>
using namespace std;
char stack[30], postfix[30], infix[30];
int top = - 1;
int pri( char x )
{
    int value;
    switch ( x )
    {
        case ')': value=0;    break;
        case '+': case '-': value=1;    break;
        case '*': case '/': case '%': value=2;    break;
        case '^': value=3;    break;
        case '(': value=4;    break;
        default: cout<<"INVALID EXPRESSION !!!!!!";
                exit(1);
    }
    return value;
}
void push ( char x )
{
    top = top + 1;
    stack [top] = x;
}
char stacktop ( )
{
    return stack [ top ];
}
int isalnum (char x)
{
    return ( (x>='0' && x<='9') || (x>='a' && x<='z') || (x>='A' && x<='Z')));
}
char pop( )
{
    return stack[top--];
}
void intopost(char infix[ ], char postfix[ ])
{
    int i, j=0;
    char c, pc;
    for(i = 0;(c =infix[ i ])!='\0';i++)
    {
```

```

    if ( isalnum (c) )
    postfix[ j++] = c;
    else
    {
        while(top!=-1&&(pri(stacktop())>= pri(c)))
        {
            if(stacktop( )=='('&& c!=''))
                break;
            if(stacktop( )=='('&&c==''))
            {
                pop () ;
                break;
            }
            pc = pop( );
            if(pc!='(')
                postfix[j++] = pc;
            else break;
        }
        if(c!='')
            push(c);
    }
}
while(top!=-1)
postfix[ j++] = pop( );
postfix [ j ] = '\0';
}

```

```

int main ( )
{
    cout<<"ENTER INFIX EXPRESSION ::\n\n\t\t\t";
    cin>>infix;
    intopost( infix, postfix );
    cout<<"POSTFIX EXPRESSION ::\n\n\t\t\t ";
    cout<<postfix;
}

```

OUTPUT:-

ENTER INFIX EXPRESSION ::

A+(B*C-(D/EXF)*C)*H

POSTFIX EXPRESSION ::

ABC*DEXF/C*-H*+

2.(a)IMPLEMENTATION OF STACK USING POINTER

PROGRAM:-

```
#include<iostream>
using namespace std;
#define STACKSIZE 5
struct bufferstack
{
    int stk[STACKSIZE];
    int top;
} pointer;
void push();
int pop();
void display();

int main()
{
    int c;
    pointer.top=-1; //Set pointer to -1
    int x=1;
    while(x)
    {
        cout<<"\n Pointer is at : \t"<<pointer.top+1<<endl;
        cout<<"\n Please enter your choice : "<<endl<<"1: Push"<<endl<<"2: Pop"<<endl<<"3:
        Display"<<endl<<"4: Exit"<<endl;
        cin>>c;
        switch(c)
        {

            case 1:
                push();
                break;
            case 2:
                pop();
                break;
            case 3:
                display();
                break;
            case 4:
                return 0;
        }
        cout<<" \n Do you want to cotin....? press 1 or 0"<<endl;
        cin>>x;
    }
}
```

```

void push()
{
    int num;
    if(pointer.top==(STACKSIZE-1))
    {
        cout<<"\n Sorry You can't push any element into stack .... ,Stack is full";
    }
    else
    {
        cout<<"\n Enter the number to push into stack";
        cin>>num;
        pointer.top+=1;
        pointer.stk[pointer.top]= num;

    }
}

int pop()
{
    int num;
    if(pointer.top==-1)
    {
        cout<<"\nstack is empty "<<endl;
    }
    else
    {
        num=pointer.top;
        cout<<"\n Poped number is : "<<pointer.stk[num];
        pointer.stk[num]=0;
        pointer.top-=1;
    }
    return num;
}

void display()
{
    if(pointer.top==-1)
    {
        cout<<"\n Stack is empty"<<endl;
    }
    else
    {
        for(int i=pointer.top;i>=0;i--)
        {
            cout<<"\n"<<i<<":"<<pointer.stk[i]<<endl;
        }
        cout<<pointer.top;
    }
}

```

OUTPUT:-

Pointer is at : 0

Please enter your choice :

1: Push

2: Pop

3: Display

4: Exit

1

Enter the number to push into stack25

Do you want to cotin....? press 1 or 0

1

Pointer is at : 1

Please enter your choice :

1: Push

2: Pop

3: Display

4: Exit

1

Enter the number to push into stack26

Do you want to cotin....? press 1 or 0

1

Pointer is at : 2

Please enter your choice :

1: Push

2: Pop

3: Display

4: Exit

1

Enter the number to push into stack27

Do you want to cotin....? press 1 or 0

1

Pointer is at : 3

Please enter your choice :

1: Push

2: Pop

3: Display

4: Exit

3

2:27

1:26

0:25

2

Do you want to cotin....? press 1 or 0

1

Pointer is at : 3

Please enter your choice :

1: Push

2: Pop

3: Display

4: Exit

2

Poped number is : 27

Do you want to cotin....? press 1 or 0

1

Pointer is at : 2

Please enter your choice :

1: Push

2: Pop

3: Display

4: Exit

3

1:26

0:25

1

Do you want to cotin....? press 1 or 0

0

Process returned 0 (0x0) execution time : 50.164 s

Press any key to continue.

2(b).IMPLEMENTATION OF QUEUE USING POINTER

PROGRAM:-

```
#include<iostream>
#include<stdio.h>
#include<conio.h>
using namespace std;
struct node
{
    int data;
    node *next;
} *front = NULL, *rear = NULL, *p = NULL, *np = NULL;
void push(int x)
{
    np = new node;
    np->data = x;
    np->next = NULL;
    if(front == NULL)
    {
        front = rear = np;
        rear->next = NULL;
    }
    else
    {
        rear->next = np;
        rear = np;
        rear->next = NULL;
    }
}
int remove()
{
    int x;
    if(front == NULL)
    {
        cout<<"empty queue\n";
    }
    else
    {
        p = front;
        x = p->data;
        front = front->next;
        delete(p);
        return(x);
    }
}
int main()
```

```

{
    int n,c = 0,x;
    cout<<"Enter the number of values to be pushed into queue\n";
    cin>>n;
    while (c < n)
    {
        cout<<"Enter the value to be entered into queue\n";
        cin>>x;
        push(x);
        c++;
    }
    cout<<"\n\nRemoved Values\n\n";
    while(true)
    {
        if (front != NULL)
            cout<<remove()<<endl;
        else
            break;
    }
    getch();
}

```

OUTPUT:-

Enter the number of values to be pushed into queue

3

Enter the value to be entered into queue

25

Enter the value to be entered into queue

26

Enter the value to be entered into queue

27

Removed Values

25

26

27

Process returned 0 (0x0) execution time : 20.019 s

Press any key to continue.

2 (c). LIST OPERATIONS USING POINTER

PROGRAM:-

```
#include<iostream>
using namespace std;
#define NULL 0
class node
{
    int data;
    node *next;
public:
    void create(node *);
    void print(node *);
};
void node :: create (node *list)
{
    cout<<"\n ENTER THE INPUT NO :: ";
    cout<<"\n TYPE 10 AT THE END :: ";
    cin>>list->data;
    if(list->data==10)
        list->next = NULL;
    else
    {
        list->next=new node;
        create(list->next);
    }
    return;
}
void node :: print (node *list)
{
    if(list->next!= 0)
    {
        cout<< list->data;
        cout<<"->";
    }
    else
        return;
    print(list->next);
}
int main()
{
    node *head, ob;
    head = new node;
    ob.create(head);
    cout<<"\n LIST ELEMENTS ARE:: ";
```

```
        ob.print(head);
        cout<<" 10 TO BE END";
    }
```

OUTPUT:-

ENTER THE INPUT NO ::
TYPE 10 AT THE END :: 1

ENTER THE INPUT NO ::
TYPE 10 AT THE END :: 2

ENTER THE INPUT NO ::
TYPE 10 AT THE END :: 3

ENTER THE INPUT NO ::
TYPE 10 AT THE END :: 4

ENTER THE INPUT NO ::
TYPE 10 AT THE END :: 5

ENTER THE INPUT NO ::
TYPE 10 AT THE END :: 6

ENTER THE INPUT NO ::
TYPE 10 AT THE END :: 7

ENTER THE INPUT NO ::
TYPE 10 AT THE END :: 8

ENTER THE INPUT NO ::
TYPE 10 AT THE END :: 9

ENTER THE INPUT NO ::
TYPE 10 AT THE END :: 10

LIST ELEMENTS ARE:: 1->2->3->4->5->6->7->8->9-> 10 TO BE END
Process returned 0 (0x0) execution time : 11.734 s
Press any key to continue.

2(d).IMPLEMENTATION OF DOUBLY LINKED LIST

PROGRAM:-

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
using namespace std;
struct node
{
    int info;
    struct node *next;
    struct node *prev;
}*start;

class double_llist
{
public:
    void create_list(int value);
    void add_begin(int value);
    void add_after(int value, int position);
    void delete_element(int value);
    void display_dlist();
    double_llist()
    {
        start = NULL;
    }
};

int main()
{
    int choice, element, position;
    double_llist dl;
    while (1)
    {
        cout<<endl<<"-----"<<endl;
        cout<<endl<<"Operations on Doubly linked list"<<endl;
        cout<<endl<<"-----"<<endl;
        cout<<"1.Create Node"<<endl;
        cout<<"2.Add at begining"<<endl;
        cout<<"3.Add after position"<<endl;
        cout<<"4.Delete"<<endl;
        cout<<"5.Display"<<endl;
        cout<<"6.Quit"<<endl;
        cout<<"Enter your choice : ";
        cin>>choice;
```

```

switch ( choice )
{
case 1:
    cout<<"Enter the element: ";
    cin>>element;
    dl.create_list(element);
    cout<<endl;
    break;
case 2:
    cout<<"Enter the element: ";
    cin>>element;
    dl.add_begin(element);
    cout<<endl;
    break;
case 3:
    cout<<"Enter the element: ";
    cin>>element;
    cout<<"Insert Element after postion: ";
    cin>>position;
    dl.add_after(element, position);
    cout<<endl;
    break;
case 4:
    if (start == NULL)
    {
        cout<<"List empty,nothing to delete"<<endl;
        break;
    }
    cout<<"Enter the element for deletion: ";
    cin>>element;
    dl.delete_element(element);
    cout<<endl;
    break;
case 5:
    dl.display_dlist();
    cout<<endl;
    break;
case 6:
    exit(1);
default:
    cout<<"Wrong choice"<<endl;
}
}
return 0;
}

```

```

void double_llist::create_list(int value)
{
    struct node *s, *temp;
    temp = new(struct node);
    temp->info = value;
    temp->next = NULL;
    if (start == NULL)
    {
        temp->prev = NULL;
        start = temp;
    }
    else
    {
        s = start;
        while (s->next != NULL)
            s = s->next;
        s->next = temp;
        temp->prev = s;
    }
}

void double_llist::add_begin(int value)
{
    if (start == NULL)
    {
        cout<<"First Create the list."<<endl;
        return;
    }
    struct node *temp;
    temp = new(struct node);
    temp->prev = NULL;
    temp->info = value;
    temp->next = start;
    start->prev = temp;
    start = temp;
    cout<<"Element Inserted"<<endl;
}

void double_llist::add_after(int value, int pos)
{
    if (start == NULL)
    {
        cout<<"First Create the list."<<endl;
        return;
    }
    struct node *tmp, *q;

```

```

int i;
q = start;
for (i = 0; i < pos - 1; i++)
{
    q = q->next;
    if (q == NULL)
    {
        cout<<"There are less than ";
        cout<<pos<<" elements."<<endl;
        return;
    }
}
tmp = new(struct node);
tmp->info = value;
if (q->next == NULL)
{
    q->next = tmp;
    tmp->next = NULL;
    tmp->prev = q;
}
else
{
    tmp->next = q->next;
    tmp->next->prev = tmp;
    q->next = tmp;
    tmp->prev = q;
}
cout<<"Element Inserted"<<endl;
}
void double_llist::delete_element(int value)
{
    struct node *tmp, *q;
    /*first element deletion*/
    if (start->info == value)
    {
        tmp = start;
        start = start->next;
        start->prev = NULL;
        cout<<"Element Deleted"<<endl;
        free(tmp);
        return;
    }
    q = start;
    while (q->next->next != NULL)
    {
        /*Element deleted in between*/

```

```

        if (q->next->info == value)
        {
            tmp = q->next;
            q->next = tmp->next;
            tmp->next->prev = q;
            cout<<"Element Deleted"<<endl;
            free(tmp);
            return;
        }
        q = q->next;
    }
    if (q->next->info == value)
    {
        tmp = q->next;
        free(tmp);
        q->next = NULL;
        cout<<"Element Deleted"<<endl;
        return;
    }
    cout<<"Element "<<value<<" not found"<<endl;
}

void double_llist::display_dlist()
{
    struct node *q;
    if (start == NULL)
    {
        cout<<"List empty,nothing to display"<<endl;
        return;
    }
    q = start;
    cout<<"The Doubly Link List is :"<<endl;
    while (q != NULL)
    {
        cout<<q->info<<" <-> ";
        q = q->next;
    }
    cout<<"NULL"<<endl;
}

```

OUTPUT:-

Operations on Doubly linked list

- 1.Create Node
- 2.Add at begining
- 3.Add after position
- 4.Delete
- 5.Display
- 6.Quit

Enter your choice : 1
Enter the element: 25

Operations on Doubly linked list

- 1.Create Node
- 2.Add at begining
- 3.Add after position
- 4.Delete
- 5.Display
- 6.Quit

Enter your choice : 2
Enter the element: 24
Element Inserted

Operations on Doubly linked list

- 1.Create Node
- 2.Add at begining
- 3.Add after position
- 4.Delete
- 5.Display
- 6.Quit

Enter your choice : 3
Enter the element: 26
Insert Element after postion: 2
Element Inserted

Operations on Doubly linked list

- 1.Create Node
- 2.Add at begining
- 3.Add after position
- 4.Delete
- 5.Display

6.Quit

Enter your choice : 5

The Doubly Link List is :

24 <-> 25 <-> 26 <-> NULL

Operations on Doubly linked list

1.Create Node

2.Add at begining

3.Add after position

4.Delete

5.Display

6.Quit

Enter your choice : 6

Process returned 1 (0x1) execution time : 51.563 s

Press any key to continue.

2(e).IMPLEMENTING POLYNOMIAL ADDITION USING LINKED LIST

PROGRAM:-

```
#include <iostream>
#include <iomanip>
using namespace std;
struct poly{
    int coeff;
    int pow;
    poly *next;
};
class add2poly
{
    poly *poly1, *poly2, *poly3;
public:
    add2poly(){poly1=poly2=poly3=NULL;}
    void addpoly();
    void display();
};
void add2poly :: addpoly(){
    int i,p;
    poly *newl=NULL,*end=NULL;
    cout<<"Enter highest power for x\n";
    cin>>p;
    cout<<"\nFirst Polynomial\n";
    for(i=p;i>=0;i--)
    {
        newl=new poly;
        newl->pow=p;
        cout<<"Enter Co-efficient for degree"<<i<<":: ";
        cin>>newl->coeff;
        newl->next=NULL;
        if(poly1==NULL)
            poly1=newl;
        else
            end->next=newl;
        end=newl;
    }
    cout<<"\n\nSecond Polynomial\n";
    end=NULL;
    for(i=p;i>=0;i--)
    {
        newl=new poly;
        newl->pow=p;
```



```

        cout<<"Enter Co-efficient for degree"<<i<<":: ";
        cin>>newl->coeff;
        newl->next=NULL;
        if(poly2==NULL)
            poly2=newl;
        else
            end->next=newl;
        end=newl;
    }
    //Addition Logic
    poly *p1=poly1,*p2=poly2;
    end=NULL;
    while(p1 !=NULL && p2!=NULL){
        if(p1->pow == p2->pow){
            newl=new poly;
            newl->pow=p--;
            newl->coeff=p1->coeff + p2->coeff;
            newl->next=NULL;
            if(poly3==NULL)
                poly3=newl;
            else
                end->next=newl;
            end=newl;
        }
        p1=p1->next;
        p2=p2->next;
    }
}

void add2poly :: display(){
    poly *t=poly3;
    cout<<"\n\nAnswer after addition is : ";
    while(t!=NULL){
        cout.setf(ios::showpos);
        cout<<t->coeff;
        cout.unsetf(ios::showpos);
        cout<<"X"<<t->pow;
        t=t->next;
    }
}

int main(){
    add2poly obj;
    obj.addpoly();
    obj.display();
}

```

OUTPUT:-

Enter highest power for x

2

First Polynomial

Enter Co-efficient for degree2:: 12

Enter Co-efficient for degree1:: 13

Enter Co-efficient for degree0:: 14

Second Polynomial

Enter Co-efficient for degree2:: 9

Enter Co-efficient for degree1:: 8

Enter Co-efficient for degree0:: 7

Answer after addition is : $+21X^2+21X+21X^0$

Process returned 0 (0x0) execution time : 30.859 s

Press any key to continue.

2(f).IMPLEMENTING POLYNOMIAL MULTIPLICATION USING LINKED LIST

PROGRAM:-

```
#include <iostream>
#include <iomanip>
using namespace std;
struct poly{
    int coeff;
    int pow;
    poly *next;
};

class mul2poly
{
    poly *poly1, *poly2, *poly3;
public:
    mul2poly(){poly1=poly2=poly3=NULL;}
    void mulpoly();
    void display();
};

void mul2poly :: mulpoly(){
    int i,p;
    poly *newl=NULL,*end=NULL;
    cout<<"Enter highest power for x\n";
    cin>>p;
    cout<<"\nFirst Polynomial\n";
    for(i=p;i>=0;i--){
        newl=new poly;
        newl->pow=i;
        cout<<"Enter Co-efficient for degree"<<i<<":: ";
        cin>>newl->coeff;
        newl->next=NULL;
        if(poly1==NULL)
            poly1=newl;
        else
            end->next=newl;
        end=newl;
    }
    cout<<"\n\nSecond Polynomial\n";
    end=NULL;

    for(i=p;i>=0;i--){
        {
```

```

newl=new poly;
newl->pow=i;
cout<<"Enter Co-efficient for degree"<<i<<":: ";
cin>>newl->coeff;
newl->next=NULL;
if(poly2==NULL)
    poly2=newl;
else
    end->next=newl;
end=newl;
}
poly *p1=poly1,*p2=poly2;
int flag;
end=NULL;
while(p1 !=NULL){
    p2=poly2;
    while(p2!=NULL){
        if(p1->pow == p2->pow){
            newl=new poly;
            newl->pow=p1->pow + p2->pow;
            newl->coeff=p1->coeff * p2->coeff;
            newl->next=NULL;
            if(poly3==NULL)
                poly3=newl;
            else{
                flag=1;
                poly *tmp=poly3;
                while(tmp!=NULL){
                    if(tmp->pow==newl->pow){
                        tmp->coeff += newl->coeff;
                        flag=0;
                    }
                    tmp=tmp->next;
                }

                if(flag==1)
                    end->next=newl;
            }

            end=newl;
            p2=p2->next;
        }
        p1=p1->next;
    }
}
}

```

```

void mul2poly :: display(){
    poly *t=poly3;
    cout<<"\n\nAnswer after multiplication is : ";
    while(t!=NULL){
        cout.setf(ios::showpos);
        cout<<t->coeff;
        cout.unsetf(ios::showpos);
        cout<<"X"<<t->pow;
        t=t->next;
    }
}

int main(){
    mul2poly obj;
    obj.mulpoly();
    obj.display();
}

```

OUTPUT:-

Enter highest power for x

2

First Polynomial

Enter Co-efficient for degree2:: 11

Enter Co-efficient for degree1:: 13

Enter Co-efficient for degree0:: 10

Second Polynomial

Enter Co-efficient for degree2:: 9

Enter Co-efficient for degree1:: 8

Enter Co-efficient for degree0:: 7

Answer after multiplication is : +99X4+205X3+271X2

Process returned 0 (0x0) execution time : 17.969 s

Press any key to continue.

3. IMPLEMENTATION OF BINARY SEARCH TREE

PROGRAM:-

```
#include <iostream>
using namespace std;
#include <conio.h>
struct tree
{
    tree *l, *r;
    int data;
} *root = NULL, *p = NULL, *np = NULL, *q;

void create()
{
    int value, c = 0;
    while (c < 7)
    {
        if (root == NULL)
        {
            root = new tree;
            cout<<"enter value of root node\n";
            cin>>root->data;
            root->r=NULL;
            root->l=NULL;
        }
        else
        {
            p = root;
            cout<<"enter value of node\n";
            cin>>value;
            while(true)
            {
                if (value < p->data)
                {
                    if (p->l == NULL)
                    {
                        p->l = new tree;
                        p = p->l;
                        p->data = value;
                        p->l = NULL;
                        p->r = NULL;
                        cout<<"value entered in left\n";
                        break;
                    }
                }
                else if (p->l != NULL)
```

```

        {
            p = p->l;
        }
    }
    else if (value > p->data)
    {
        if (p->r == NULL)
        {
            p->r = new tree;
            p = p->r;
            p->data = value;
            p->l = NULL;
            p->r = NULL;
            cout<<"value entered in right\n";
            break;
        }
        else if (p->r != NULL)
        {
            p = p->r;
        }
    }
}
c++;
}
}

void inorder(tree *p)
{
    if (p != NULL)
    {
        inorder(p->l);
        cout<<p->data<<endl;
        inorder(p->r);
    }
}

void preorder(tree *p)
{
    if (p != NULL)
    {
        cout<<p->data<<endl;
        preorder(p->l);
        preorder(p->r);
    }
}

void postorder(tree *p)
{

```

```
    if (p != NULL)
    {
        postorder(p->l);
        postorder(p->r);
        cout<<p->data<<endl;
    }
}
int main()
{
    create();
    cout<<"printing traversal in inorder\n";
    inorder(root);
    cout<<"printing traversal in preorder\n";
    preorder(root);
    cout<<"printing traversal in postorder\n";
    postorder(root);
    getch();
}
```


OUTPUT:-

enter value of root node

7

enter value of node

8

value entered in right

enter value of node

4

value entered in left

enter value of node

6

value entered in right

enter value of node

3

value entered in left

enter value of node

5

value entered in left

enter value of node

2

value entered in left

printing traversal in inorder

2

3

4

5

6

7

8

printing traversal in preorder

7

4

3

2

6

5

8

printing traversal in postorder

2

3

5

6

4

8

7

4(a).IMPLEMENTATION OF BINARY SEARCH

PROGRAM:-

```
#include<iostream>
using namespace std;
int main()
{
    int n, i, arr[50], search, first, last, middle;
    cout<<"Enter total number of elements :";
    cin>>n;
    cout<<"Enter "<<n<<" number :";
    for (i=0; i<n; i++)
    {
        cin>>arr[i];
    }
    cout<<"Enter a number to find :";
    cin>>search;
    first = 0;
    last = n-1;
    middle = (first+last)/2;
    while (first <= last)
    {
        if(arr[middle] < search)
        {
            first = middle + 1;
        }
        else if(arr[middle] == search)
        {
            cout<<search<<" found at location "<<middle+1<<"\n";
            break;
        }
        else
        {
            last = middle - 1;
        }
        middle = (first + last)/2;
    }
    if(first > last)
    {
        cout<<"Not found! "<<search<<" is not present in the list.";
    }
}
```

OUTPUT:-

Enter total number of elements :6

Enter 6 number :

11

12

10

9

15

16

Enter a number to find :16

16 found at location 6

Process returned 0 (0x0) execution time : 21.101 s

Press any key to continue.

4. (b) INSERTION SORT

PROGRAM:

```
#include<iostream>
using namespace std;
int main()
{
    int i,j,n,temp,a[30];
    cout<<"Enter the number of elements:";
    cin>>n;
    cout<<"\nEnter the elements\n";
    for(i=0;i<n;i++)
    {
        cin>>a[i];
    }
    for(i=1;i<=n-1;i++)
    {
        temp=a[i];
        j=i-1;

        while((temp<a[j])&&(j>=0))
        {
            a[j+1]=a[j];    //moves element forward
            j=j-1;
        }
        a[j+1]=temp;    //insert element in proper place
    }
    cout<<"\nSorted list is as follows\n";
    for(i=0;i<n;i++)
    {
        cout<<a[i]<<" ";
    }

    return 0;
}
```

OUTPUT:-

Enter the number of elements:**10**

Enter the elements **96 1 78 2 4 6 7 8 10 58**

Sorted list is as follows **1 2 4 6 7 8 10 58 78 96**

Process returned 0 (0x0) execution time : **19.577 s**

Press any key to continue.

4. (c) QUICK SORT

PROGRAM

```
#include<iostream>
#include<iomanip>
using namespace std;
int partition(int a[],int start,int end)
{
    int pivot=a[end];
    int pindex=start;
    for(int i=start;i<end;i++)
    {
        if(a[i]<=pivot)
        {
            swap(a[i],a[pindex]);
            pindex++;
        }
    }
    swap(a[pindex],a[end]);
    return pindex;
}
void quicksort(int a[],int start,int end)
{
    if(start<end)
    {
        int p=partition(a,start,end);
        quicksort(a,start,p-1);
        quicksort(a,p+1,end);
    }
}

int main()
{
    int a[]={ 1,0,2,9,3,8,4,7,5,6},n=10;
    cout<<"\ngiven numbers\n";
    for(int i=0;i<n;i++)
        cout<<setw(5)<<a[i];
    quicksort(a,0,n-1);
    cout<<"\nSorted Numbers:\n";
    for(int i=0;i<n;i++)
        cout<<setw(5)<<a[i];
    return 0;
}
```

OUTPUT:-

Given Numbers

1 0 2 9 3 8 4 7 5 6

Sorted Numbers:

0 1 2 3 4 5 6 7 8 9

Process returned 0 (0x0) execution time : **0.484 s**

Press any key to continue.

4. (d) SELECTION SORT

PROGRAM

```
#include<iostream>
using namespace std;
int main()
{
    int i,j,n,loc,temp,min,a[30];
    cout<<"Enter the number of elements:";
    cin>>n;
    cout<<"\nEnter the elements\n";
    for(i=0;i<n;i++)
    {
        cin>>a[i];
    }
    for(i=0;i<n-1;i++)
    {
        min=a[i];
        loc=i;
        for(j=i+1;j<n;j++)
        {
            if(min>a[j])
            {
                min=a[j];
                loc=j;
            }
        }

        temp=a[i];
        a[i]=a[loc];
        a[loc]=temp;
    }
    cout<<"\nSorted list is as follows\n";
    for(i=0;i<n;i++)
    {
        cout<<a[i]<<" ";
    }

    return 0;
}
```

OUTPUT:-

Enter the number of elements:**10**

Enter the elements **96 35 21 45 10 36 78 100 01 42**

Sorted list is as follows **1 10 21 35 36 42 45 78 96 100**

Process returned 0 (0x0) execution time : **20.875 s**

Press any key to continue.

4. (e) BUBBLE SORT

PROGRAM

```
#include<iostream>
using namespace std;
int main()
{
    int a[50],n,i,j,temp;
    cout<<"Enter the size of array: ";
    cin>>n;
    cout<<"Enter the array elements: ";
    for(i=0;i<n;++i)
        cin>>a[i];
    for(i=1;i<n;++i)
    {
        for(j=0;j<(n-i);++j)
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
    }
    cout<<"Array after bubble sort:";
    for(i=0;i<n;++i)
        cout<<" "<<a[i];

    return 0;
}
```

OUTPUT:-

Enter the size of array: **10**

Enter the array elements: **31 12 38 40 16 19 21 02 03 04**

Array after bubble sort: **2 3 4 12 16 19 21 31 38 40**

Process returned 0 (0x0) execution time : **24.606 s**

Press any key to continue.

4.(f) MERGE SORT

PROGRAM

```
#include<iostream>
#include<iomanip>
using namespace std;
void merge(int a[],int start,int mid,int end)
{
    int temp[10],i,j,k;
    for(i=start;i<=end;i++)
        temp[i]=a[i];
    i=start;k=start;j=mid+1;
    while(i<=mid && j<=end)
        if(temp[i]<temp[j])
            a[k++]=temp[i++];
        else
            a[k++]=temp[j++];
    while(i<=mid) a[k++]=temp[i++];
    while(j<=end) a[k++]=temp[j++];
}
void mergesort(int a[],int start,int end)
{
    if(start<end)
    {
        int mid=(start+end)/2;
        mergesort(a,start,mid);
        mergesort(a,mid+1,end);
        merge(a,start,mid,end);
    }
}

int main()
{
    int a[]={ 1,0,2,9,3,8,4,7,5,6},n=10;
    cout<<"\ngiven numbers\n";
    for(int i=0;i<n;i++)
        cout<<setw(5)<<a[i];
    mergesort(a,0,n-1);
    cout<<"\nSorted Numbers:\n";
    for(int i=0;i<n;i++)
        cout<<setw(5)<<a[i];
    return 0;
}
```

OUTPUT:-

Given numbers

1 0 2 9 3 8 4 7 5 6

Sorted Numbers:

0 1 2 9 3 8 4 7 5 6

Process returned 0 (0x0) execution time : **0.375 s**

Press any key to continue.

4(g) HEAP SORT:-

PROGRAM:-

```
#include <iostream>
#include <conio.h>
using namespace std;
void max_heapify(int *a, int i, int n)
{
    int j, temp;
    temp = a[i];
    j = 2*i;
    while (j <= n)
    {
        if (j < n && a[j+1] > a[j])
            j = j+1;
        if (temp > a[j])
            break;
        else if (temp <= a[j])
        {
            a[j/2] = a[j];
            j = 2*j;
        }
    }
    a[j/2] = temp;
    return;
}
void heapsort(int *a, int n)
{
    int i, temp;
    for (i = n; i >= 2; i--)
    {
        temp = a[i];
        a[i] = a[1];
        a[1] = temp;
        max_heapify(a, 1, i - 1);
    }
}
void build_maxheap(int *a, int n)
{
    int i;
    for(i = n/2; i >= 1; i--)
    {
        max_heapify(a, i, n);
    }
}
```

```

int main()
{
    int n, i, x;
    cout<<"enter no of elements of array\n";
    cin>>n;
    int a[20];
    for (i = 1; i <= n; i++)
    {
        cout<<"enter element"<<(i)<<endl;
        cin>>a[i];
    }
    build_maxheap(a,n);
    heapsort(a, n);
    cout<<"sorted output\n";
    for (i = 1; i <= n; i++)
    {
        cout<<a[i]<<endl;
    }
    getch();
}

```

OUTPUT:-

```

enter no of elements of array 6
enter element1 12
enter element2 10
enter element3 8
enter element4 15
enter element5 11
enter element6 4
sorted output
4
8
10
11
12
15

```

Process returned 0 (0x0) execution time : 40.406 s
Press any key to continue.

4. (h) RADIX SORT:-

PROGRAM:-

```
#include <iostream>
#include <cstdlib>
using namespace std;

int getMax(int arr[], int n)
{
    int max = arr[0];
    for (int i = 1; i < n; i++)
        if (arr[i] > max)
            max = arr[i];
    return max;
}

void countSort(int arr[], int n, int exp)
{
    int output[n];
    int i, count[10] = {0};
    for (i = 0; i < n; i++)
        count[(arr[i] / exp) % 10]++;
    for (i = 1; i < 10; i++)
        count[i] += count[i - 1];
    for (i = n - 1; i >= 0; i--)
    {
        output[count[(arr[i] / exp) % 10] - 1] = arr[i];
        count[(arr[i] / exp) % 10]--;
    }

    for (i = 0; i < n; i++)
        arr[i] = output[i];
}

void radixsort(int arr[], int n)
{
    int m = getMax(arr, n);
    for (int exp = 1; m / exp > 0; exp *= 10)
        countSort(arr, n, exp);
}

int main()
{
    int arr[] = {170, 45, 75, 90, 802, 24, 2, 66};
    int n = sizeof(arr)/sizeof(arr[0]);
    radixsort(arr, n);
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
    return 0;
}
```

```
}
```

OUTPUT:-

2 24 45 66 75 90 170 802

Process returned 0 (0x0) execution time : 0.359 s

Press any key to continue.

5(a).GRAPH IMPLEMENTATION OF DEPTH FIRST SEARCH

PROGRAM:-

```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
using namespace std;
int cost[10][10],i,j,k,n,stk[10],top,v,visit[10],visited[10];
int main()
{
int m;
cout <<"enter no of vertices";
cin >> n;
cout <<"enter no of edges";
cin >> m;
cout <<"\nEDGES \n";
for(k=1;k<=m;k++)
{
cin >>i>>j;
cost[i][j]=1;
}
cout <<"enter initial vertex";
cin >>v;
cout <<"ORDER OF VISITED VERTICES";
cout << v <<" ";
visited[v]=1;
k=1;
while(k<n)
{
for(j=n;j>=1;j--)
if(cost[v][j]!=0 && visited[j]!=1 && visit[j]!=1)
{
visit[j]=1;
stk[top]=j;
top++;
}
v=stk[--top];
cout<<v << " ";
k++;
visit[v]=0; visited[v]=1;
}
}
```


OUTPUT:-

enter no of vertices 9

enter no of edges 9

EDGES

1 2

2 3

2 6

1 5

1 4

4 7

5 7

7 8

8 9

enter initial vertex 1

ORDER OF VISITED VERTICES

1 2 3 6 4 7 8 9 5

5(b).GRAPH IMPLEMENTATION BREADTH FIRST SEARCH

PROGRAM:-

```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
using namespace std;
int cost[10][10],i,j,k,n,qu[10],front,rear,v,visit[10],visited[10];
int main()
{
int m;
cout <<"enter no of vertices";
cin >> n;
cout <<"enter no of edges";
cin >> m;
cout <<"\nEDGES \n";
for(k=1;k<=m;k++)
{
cin >>i>>j;
cost[i][j]=1;
}
cout <<"enter initial vertex";
cin >>v;
cout <<"Visited vertices\n";
cout << v;
visited[v]=1;
k=1;
while(k<n)
{
for(j=1;j<=n;j++)
if(cost[v][j]!=0 && visited[j]!=1 && visit[j]!=1)
{
visit[j]=1;
qu[rear++]=j;
}
v=qu[front++];
cout<<v << " ";
k++;
visit[v]=0; visited[v]=1;
}
}
```

OUTPUT:-

enter no of vertices 9
enter no of edges 9

EDGES

1 2

2 3

1 5

1 4

4 7

7 8

8 9

2 6

5 7

enter initial vertex 1

Visited vertices

1 2 4 5 3 6 7 8 9