3.1

```
def removeDuplicate(d):
    lst=[]
    for i in d:
        if i not in lst:
            lst.append(i)
    return lst

# Function for finding intersection between two sets (A&B)

def intersection(lst1,lst2):
    lst3=[]
    for val in lst1:
        if val in lst2:
            lst3.append(val)
    return lst3


# Function for finding union of two sets (A|B)

def union(lst1,lst2):
    lst3=lst1.copy()
    for val in lst2:
        if val not in lst3:
            lst3.append(val)
    return lst3

# Function for finding difference between two sets (A-B)

def diff(lst1,lst2):
    lst3=[]
    for val in lst1:
        if val not in lst2:
            lst3.append(val)
    return lst3


# Function for finding symmetric difference of two sets (A^B)

def sym_diff(lst1,lst2):
    lst3=[]
    D1=diff(lst1,lst2)
    print("Difference between Cricket and Badminton (C-B) is : ", D1)
    D2=diff(lst2,lst1)
    print("Difference between Badminton and Cricket (B-C) is : ", D2)
```

3.2

```
lst3=union(D1,D2)
    return lst3

# Functon for finding List of students who play both cricket and badminton

def CB(lst1,lst2):
    lst3=intersection(lst1,lst2)
    print("\n\nList of students who play both cricket and badminton is : ", lst3)
```

```
        return len(lst3)


# Function for finding List of students who play either cricket or badminton but not both

def eCeB(lst1,lst2):
        lst3=sym_diff(lst1,lst2)
        print("\nList of students who play either cricket or badminton but not both is : ",lst3)
        return len(lst3)


# Function for finding Number of students who play neither cricket nor badminton

def nCnB(lst1,lst2,lst3):
        lst4=diff(lst1,union(lst2,lst3))
        print("\n\nList of students who play neither cricket nor badminton is : ",lst4)
        return len(lst4)


# Function for finding Number of students who play cricket and football but not badminton

def CBnF(lst1,lst2,lst3):
        lst4=diff(intersection(lst1,lst2),lst3)
        print("\n\nList of students who play cricket and football but not badminton is : ",lst4)
        return len(lst4)

# Main function

# Creating an empty list for SE COMP
3.3

SEComp = []
n = int(input("\nEnter number of students in SE COMP: "))
print("Enter the names of",n,"students (Please press ENTER after entering each students name) :")
for i in range(0, n):
        ele = input()
        SEComp.append(ele) # adding the element
print("Original list of students in SEComp : " + str(SEComp))



# Creating an empty list for Cricket
Cricket = []
n = int(input("\n\nEnter number of students who play cricket : "))
print("Enter the names of",n,"students who play cricket :")
for i in range(0, n):
        ele = input()
        Cricket.append(ele)     # adding the element
print("Original list of students playing cricket is :" +str(Cricket))
Cricket=removeDuplicate(Cricket)
print("The list of students playing cricket after removing duplicates : " +str(Cricket))



# Creating an empty list for Football
Football = []
n = int(input("\n\nEnter number of students who play football : "))
```

```
print("Enter the name of",n,"students who play football (Please press ENTER after entering each
students name) :")
for i in range(0, n):
    ele = input()
    Football.append(ele)  # adding the element
print("Original list of students playing football :" +str(Football))
Football=removeDuplicate(Football)
print("The list of students playing football after removing duplicates : " +str(Football))



# Creating an empty list for Badminton
Badminton = []
n = int(input("\n\nEnter number of students who play badminton : "))
print("Enter the name of",n,"students who play badminton (Please press ENTER after entering each
students name) :")
3.4

for i in range(0, n):
    ele = input()
    Badminton.append(ele)    # adding the element
print("Original list of students playing badminton :" +str(Badminton))
Badminton=removeDuplicate(Badminton)
print("The list of students playing badminton after removing duplicates : " +str(Badminton))

flag=1
while flag==1:
    print("\n\n------------------MENU------------------\n")
    print("1. List of students who play both cricket and badminton")
    print("2. List of students who play either cricket or badminton but not both")
    print("3. List of students who play neither cricket nor badminton")
    print("4. Number of students who play cricket and football but not badminton")
    print("5. Exit\n")
    ch=int(input("Enter your Choice (from 1 to 5) :"))

    if ch==1:
        print("Number of students who play both cricket and badminton : ", CB(Cricket,Badminton))
        a = input("\n\nDo you want to continue (yes/no) :")
        if a == "yes":
            flag = 1
        else:
            flag = 0
            print("Thanks for using this program!")

    elif ch==2:
        print("Number of students who play either cricket or badminton but not both : ",
eCeB(Cricket, Badminton))
        a = input("\n\nDo you want to continue (yes/no) :")
        if a == "yes":
            flag = 1
        else:
            flag = 0
            print("Thanks for using this program!")

    elif ch==3:
        print("Number of students who play neither cricket nor badminton : ",
nCnB(SEComp,Cricket,Badminton))
```

```
        a = input("\n\nDo you want to continue (yes/no) :")
        if a == "yes":
 3.5
        flag = 1
        else:
            flag = 0
            print("Thanks for using this program!")

    elif ch==4:
        print("Number of students who play cricket and football but not badminton : ",
CBnF(Cricket,Football,Badminton))
        a = input("\n\nDo you want to continue (yes/no) :")
        if a == "yes":
            flag = 1
        else:
            flag = 0
            print("Thanks for using this program!")

    elif ch==5:
        flag=0
        print("Thanks for using this program!")

    else:
        print("!!Wrong Choice!! ")
        a=input("\n\nDo you want to continue (yes/no) :")
        if a=="yes":
            flag=1
        else:
            flag=0
            print("exit")
```

```
output :
Enter number of students in SE COMP: 3
Enter the names of 3 students (Please press ENTER after entering each students name) :
ABC
XYZ
PQR
Original list of students in SEComp : ['ABC', 'XYZ', 'PQR']


Enter number of students who play cricket : 3
Enter the names of 3 students who play cricket :
LMN
ABC
SRT
Original list of students playing cricket is :['LMN', 'ABC', 'SRT']
The list of students playing cricket after removing duplicates : ['LMN', 'ABC', 'SRT']


Enter number of students who play football : 3
3.6


Enter the name of 3 students who play football (Please press ENTER after entering each students
name) :
POR
```

```
LMN
ABC
Original list of students playing football :['POR', 'LMN', 'ABC']
The list of students playing football after removing duplicates : ['POR', 'LMN', 'ABC']


Enter number of students who play badminton : 2
Enter the name of 2 students who play badminton (Please press ENTER after entering each students
name) :
ABC
PQR
Original list of students playing badminton :['ABC', 'PQR']
The list of students playing badminton after removing duplicates : ['ABC', 'PQR']


--------------------MENU---------------------

1. List of students who play both cricket and badminton
2. List of students who play either cricket or badminton but not both
3. List of students who play neither cricket nor badminton
4. Number of students who play cricket and football but not badminton
5. Exit

Enter your Choice (from 1 to 5) :1


List of students who play both cricket and badminton is : ['ABC']
Number of students who play both cricket and badminton : 1


Do you want to continue (yes/no) :yes


--------------------MENU---------------------

1. List of students who play both cricket and badminton
2. List of students who play either cricket or badminton but not both
3. List of students who play neither cricket nor badminton
4. Number of students who play cricket and football but not badminton
5. Exit

Enter your Choice (from 1 to 5) :5
Thanks for using this program!
```

```
#check  palindrome
def isPalindrome (str):
        rev_str =str[ : :-1]
        flag=1
        if (str !=rev_str):
                flag =0
str="dsapractical"
ans=isPalindrome(str)
if (ans):
        print ("it is a palindrome")
else:
        print ("it is not a palindrome")

#display substring
str ="SE A Computer Department"
print(str.index("Computer"))

#occurance of character
str=input("enter the string :")
count=0
for i in str:
        if i =='p':
                count=count +1
        print ("count of 'p'in given string is :" ,count)

#word with longest length
str=input("enter the strings :")
1.1
longest=max(str.split(),key=len)
print("longest string is :",longest)
print("length of longest string is :",len(longest))
```

```
//matrix operations
#include <iostream>
using namespace std;

class Matrix
{
        public :
                int a[3][3];
                int b[3][3];
                int add[3][3];
                int sub[3][3];
                int multi[3][3];

        public :
                void addition();
                void subtraction();
                void multiplication();
                void transpose();
} mat;

void Matrix::addition()
{
        int i,j;

        cout << "Enter first matrix" << endl;
        for (i=0;i<3;i++)
        {
                for (j=0;j<3; j++)
                {
                cin >> a[i][j];
                }
        }

        cout << "Enter second matrix" << endl;
        for (i=0;i<3;i++)
        {
                for (j=0;j<3; j++)
                {
                cin >> b[i][j];
                }
        }

        for (i=0;i<3;i++)
        {
                for (j=0;j<3; j++)
                {
                add[i][j] = a[i][j] + b[i][j];
                }
        }

        cout << "Addition of the two matrices is : " << endl;
        for (i=0;i<3;i++)
        {
                for (j=0;j<3; j++)
                {
                cout << add[i][j] <<" ";
                }
                cout << endl;
        }
}

void Matrix::subtraction()
{
        int i,j;

        cout << "Enter first matrix" << endl;
        for (i=0;i<3;i++)
        {
                for (j=0;j<3; j++)
                {
                cin >> a[i][j];
                }
        }

        cout << "Enter second matrix" << endl;
        for (i=0;i<3;i++)
        {
                for (j=0;j<3; j++)
                {
                cin >> b[i][j];
                }
        }

        for (i=0;i<3;i++)
        {
                for (j=0;j<3; j++)
                {
                sub[i][j] = a[i][j] - b[i][j];
                }
        }

        cout << "Subtraction of the two matrices is : " << endl;
        for (i=0;i<3;i++)
        {
                for (j=0;j<3; j++)
                {
                cout << sub[i][j] <<" ";
                }
                cout << endl;
        }
}

void Matrix::transpose()
{
        int i,j;

        cout << "Enter the matrix" << endl;
        for (i=0;i<3;i++)
        {
                for (j=0;j<3; j++)
                {
                cin >> a[i][j];
                }
        }
```

```
    cout << "Transpose of the matrix is : " << endl;
    for (i=0;i<3;i++)
    {
        for (j=0;j<3; j++)
        {
        cout << a[j][i] <<" ";
        }
        cout << endl;
    }
}

void Matrix::multiplication()
{
    int i,j,k;

    cout << "Enter first matrix" << endl;
    for (i=0;i<3;i++)
    {
        for (j=0;j<3; j++)
        {
        cin >> a[i][j];
        }
    }

    cout << "Enter second matrix" << endl;
    for (i=0;i<3;i++)
    {
        for (j=0;j<3; j++)
        {
        cin >> b[i][j];
        }
    }



    for(i = 0; i < 3; ++i)
            for(j = 0; j < 3; ++j)
                    for(k = 0; k < 3; ++k)
            {
                multi[i][j] += a[i][k] * b[k][j];
            }

        cout << "Multiplcation of the two matrices is : " << endl;
    for (i=0;i<3;i++)
    {
        for (j=0;j<3; j++)
        {
        cout << multi[i][j] <<" ";
        }
        cout << endl;
    }
}
```

```
int main()
{
int chr, choice;

    do
    {
        cout << "Enter the matrix operation you want to perform" << endl;
        cout << "1.Addition" << endl;
        cout << "2.Subtraction" << endl;
        cout << "3.Transpose" << endl;
        cout << "4.Multiplication" << endl;
        cout << "Enter your choice" << endl;
        cin >> choice;

        switch(choice)
        {
            case 1: mat.addition();
                break;
            case 2: mat.subtraction();
                break;
            case 3: mat.transpose();
                break;
            case 4: mat.multiplication();
                break;
            default : cout << "Enter valid choice" << endl;
            }
        cout << "Do you want to continue? \n Press 1 for yes \n Press 0 for no\n";
        cin >> chr;
    }
    while(chr==1);

    return 0;
}
```

```
//Assignment no 4
//1. Linear Search
#include <iostream>
using namespace std;

int main()
{
	cout << "Linear search"<< endl;
	int n;
	cout << "Enter the number of elements in the array : ";
	cin >> n;

	int i, temp;
	int arr[n], key;

	cout << "Enter elements of the array : " << endl;
	for(i=0;i<n;i++)
	{
		cin >> arr[i];
	}
	cout << "Enter the key to search : " << endl;
	cin >> key;

	for (i=0; i<n; i++)
	{
		if (arr[i] == key)
		{
		cout << "Key found at : " << i << endl;
		break;
		}
	}
	if (i == n)
	cout << "key not found";

	return 0;
}
```

```
// Assignment no 4
//2. Binary search

#include <iostream>
using namespace std;

int binarySearch(int arr[], int n, int key)
{
    int lb=0;
    int ub=n;
    while(lb<=ub)
    {
        int  mid=(lb+ub)/2;

        if (arr[mid]==key)
        {
            cout << "The value is found at index number :";
            return mid;
        }
        else if(arr[mid]>key)
        {
            ub = mid-1;
        }
        else
        {
            lb= mid + 1;
        }
    }
    cout << "Key not found!";
    exit(0);
}


int main()
{
    int n;
    cout << "Enter the number of elements in an array : " << endl;
    cin >> n;

    int arr[n];
    cout << "Enter the elements in array : " << endl;
    for (int i=0; i<n; i++)
    {
        cin >> arr[i];
    }

    int key;
    cout << "Enter the value to be searched in the array : " << endl;
    cin >> key;
    cout << binarySearch(arr,n,key);

    return 0;
}
```

```
//assignment no 5
//Bubblesort

#include <iostream>
using namespace std;

int main()
{

        int n;
        cout << "Enter the number of elements : ";
        cin >> n;

        int a[n],i,j,temp;
        cout<<"Enter the elements :"<<endl;
        for(i=0;i<n;i++)
        {
                cin>>a[i];
        }

        for(i=0;i<n;i++)
        {
                for(j=i+1;j<n;j++)
                {
                        if(a[j]<a[i])
                        {
                                temp=a[i];
                                a[i]=a[j];
                                a[j]=temp;
                        }
                }
        }

        cout<<"Sorted list is: "<<endl;

        for(i=0;i<n;i++)
        cout<<a[i]<<"\t";
        cout<<endl;

        return 0;
}
```

```
//Assignment no 6
//insertion sort

#include <iostream>
using namespace std;

void display(int *array, int size)
{
        for(int  i=0;  i<size;  i++)
        cout << array[i] << " ";
        cout << endl;
}

void insertionSort(int *array, int size)
{
        int key,j;
        for(int i=1; i<size; i++)
        {
                key = array[i];
                j=i;
                while(j>0 && array[j-1]>key)
                {
                        array[j]= array[j-1];
                        j--;
                }
                array[j]=key;
        }
}

int main()
{
        int n;
        cout << "Enter the number of elements : ";
        cin >> n;
        int arr[n];

        cout << "Enter the elements : " << endl;
        for (int i=0; i<n; i++)
        {
                cin >> arr[i];
        }
        cout << "Array before sorting : ";
        display(arr,n);
        insertionSort(arr,n);
        cout << "Array after sorting : ";
        display(arr,n);

        return 0;
}
```

```
//selection sort

#include <iostream>
using namespace std;

int main()
{
        int i,j,n, loc, temp, min, a[30];
        cout << "Enter the number of elements : " << endl;
        cin >> n;
        cout << "Enter the elements : " << endl;
        for(i=0; i<n;i++)
        {
                cin >> a[i];
        }

        for(i=0; i<n; 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 << "Sorted list is as flollows : " << endl;
        for (i=0; i<n; i++)
        {
                cout << a[i] << " ";
        }
        return 0;
}
```

```
//shell sort

#include <iostream>
using namespace std;
void shellSort(int array[], int n)
{
        for (int interval = n / 2; interval > 0; interval /= 2)
        {
                for (int i = interval; i < n; i += 1)
                {
                        int temp = array[i];
                        int j;
                        for (j = i; j >= interval && array[j - interval] > temp; j -= interval)
                        {
                        array[j] = array[j - interval];
                        }
                array[j] = temp;
                }
        }
}

void printArray(int array[], int size)
{
    int i;
    for (i = 0; i < size; i++)
        cout << array[i] << " ";
    cout << endl;
}

int main()
{
        int n;
        cout << "Enter the number of elements in the array : ";
        cin >> n;

        int data[n];
        cout << "Enter the elements : " << endl;
        for (int i=0; i<n; i++)
        {
                cin >> data[i];
        }

    int size = sizeof(data) / sizeof(data[0]);
    shellSort(data, size);
    cout << "Sorted array is: \n";
    printArray(data, size);
}
```

```
// Quick sort

#include <iostream>
using namespace std;

void swap(int *a, int *b)
{
    int t = *a;
    *a = *b;
    *b = t;
}

void printArray(int array[], int size)
{
    int i;
    for (i = 0; i < size; i++)
        cout << array[i] << " ";
    cout << endl;
}

int partition(int array[], int low, int high)
{
    int pivot = array[high];
    int i = (low - 1);
    for (int j = low; j < high; j++)
    {
        if (array[j] <= pivot)
        {
            i++;
            swap(&array[i], &array[j]);
        }
    }

    swap(&array[i + 1], &array[high]);
    return (i + 1);
}

void quickSort(int array[], int low, int high)
{
    if (low < high)
    {
        int pi = partition(array, low, high);
        quickSort(array, low, pi - 1);
        quickSort(array, pi + 1, high);
    }
}


int main()
{
    int n;
    cout << "Enter the number of elements : ";
    cin >> n;

    int data[n];
        cout << "Enter the elements : " << endl;
        for (int i=0; i<n; i++)
        {
                cin >> data[i];
        }

    cout << ”Unsorted Array: \n”;
    printArray(data, n);

    quickSort(data, 0, n - 1);

    cout << ”Sorted array : \n”;
    printArray(data, n);

    return 0;
}
```

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

class Queue
{
        public:
                int arr[5];
                int front;
                int rear;

        public:
                void enqueue();
                int dequeue();
                int isfull();
                int isempty();
                void display();
}q;

void Queue:: enqueue()
{
        int val;
        cout << "Enter value to be inserted in the queue : ";
        cin>>val;
        if ( isfull() ==1)
        {
                cout << "\nOverflow!!!\n";
        }
        else
        {
                q.rear ++ ;
                q.arr[q.rear]=val;
        }
}

int Queue :: isfull()
{
        if (q.rear == 4)
        {
                return 1;
        }
        else
        {
                return 0;
        }
}

int Queue :: dequeue()
{
        int n;
        if(q.isempty()==1)
        {
                cout<<"\nUnderflow!!!\n";
        }
        else
        {
                q.front++;
                n = q.arr[q.front];
                q.arr[q.front] = 0;
        }
        return n;
}

int Queue :: isempty()
{
        if (q.front == q.rear)
        {
                return 1;
        }
        else
        {
                return 0;
        }
}

void Queue :: display()
{
        int i;
        for (i=0; i<5; i++)
        {
        cout<<endl<<q.arr[i];
        }
}

int main()
{
        q.front = -1;
        int ch;
        q.rear = -1;
        int choice;
        int n;
        n = q.dequeue();

        do
        {
                cout<<”\n**********************************************************”;
                cout<<”\nChoose the operation you want to perform on queue\n”;
                cout<<”\n1.Enqueue()”;
                cout<<”\n2.Dequeue()”;
                cout<<”\n3.Isfull()”;
                cout<<”\n4.Isempty()”;
                cout<<”\n5.Display()”;
                cout<<”\nEnter your choice : \n”;
                cin >> ch;

        switch(ch)
        {
                case 1: q.enqueue();
                cout<<”\n”;
                break;
                case 2: q.dequeue();
                cout<<”\n”<<n<<endl;
                break;
                case 3: q.isfull();
                cout<<”\n”;
```

```
            break;
            case 4: q.isempty();
            cout<<"\n";
            break;
            case 5: q.display();
            cout<< "\n";
            break;
            default : cout<<"Please enter a valic choice!!\n"<<endl;
        }
        cout<<endl<<"Do you want to continue?\n1 for yes\n0 for no  \n";
        cin>>choice;
        }
        while(choice);

        return 0;
}
```

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

class Stack
{
    public:
    int s[5],top;
    void push();
    int pop();
    int isfull();
    int isempty();
    void display();

}sp;

int Stack:: isfull()
{
    if(sp.top==4)
    {
        return 1;
    }
    else
    {
        return 0;
    }
 }

 void Stack ::push()
 {

    int p;
    cout<<"\n Enter no to push :";
    cin>>p;

    if(sp.isfull()==1)
    {
        cout<<"\n Overflow";
    }
    else
    {
        sp.top++;
        sp.s[sp.top]=p;
    }
 }

 int Stack::isempty()
 {
    if(sp.top==-1)
    {
        return 1;
    }
    else
    {
        return 0;
    }
 }
 int Stack::pop()
   {
        int r;

        if(sp.isempty()==1)
        {
                cout<<"\n Underflow";
        }
        else
        {
                r=sp.s[sp.top];
                sp.s[sp.top]=0;
                sp.top--;
                return r;
        }
    }

void Stack:: display()
{
        int i=0;
        for(i=4;i>=0;i--)
        {
                cout<<"\n "<<sp.s[i];
        }
}

int main()
{
        sp.top=-1;
        int c,ch,choice;

        do
        {

        cout<<"\n 1.Push";
        cout<<"\n 2.Pop";
        cout<<"\n  3.Disply";
        cout<<"\n Enter your choice :";
        cin>>ch;

        switch(ch)
        {

        case 1:sp.push();
                        break;
        case 2: cout<<"\n Element removed from stack :"<<sp.pop();
                break;
        case 3: sp.display();

        }

        cout<<"\n Do you want to continue? \n If yes, select 1 \n If no, select 0 \n";
        cin>>choice;
        }
        while(choice==1);

        return 0;
}
```

```
//Infix to postfix expression conversion using stack

#include <iostream>
#include <stack>

using namespace std;

bool isoperator(char);
bool isopereand(char);
bool eqlorhigher(char,char);
string convert(string);

int main()
{
    string infix_expression,postfix_expression;
    int ch;
    do
    {
      cout<<"Enter an infix expression: ";
      cin>>infix_expression;
      postfix_expression=convert(infix_expression);
      cout<<"\nYour infix expression is: "<<infix_expression<<endl;
      cout<<"\nPostfix expression is: "<<postfix_expression<<endl;
      cout<<"\nDo you want to enter another infix expression? \nFor yes, enter 1\n For no, enter 0\n";
      cin>>ch;
    }while(ch==1);

    return 0;
}

bool isoperator(char c)
{
      if(c=='+'||c=='-'||c=='*'||c=='/'||c=='^')
              return true;
      return false;
}

bool isoperand(char c)
{
      if(c>='A' && c<='Z')
              return true;
      if(c>='a' && c<='z')
              return true;
      if(c>='0' && c<='9')
              return true;
      return false;
}

int precedence(char op)
{
      if(op=='+'||op=='-')
              return 1;
      if(op=='*'||op=='/')
              return 2;
     if(op=='^')
              return 3;
      return 0;
```

```
}

bool eqlorhigher(char op1,char op2)
{
        int p1=precedence(op1);
        int p2=precedence(op2);
        if(p1==p2)
        {
                if(op1=='^')
                        return false;
                return true;
        }
        return(p1>p2?true:false);
}

string convert(string infix)
{
        stack <char>s;
        string postfix=" ";
        char ch;
        s.push('(');
        infix +=')';
        for(int i=0;i<infix.length();i++)
        {
                ch=infix[i];
                if(ch==' ')
                continue;
                else if(ch=='(')
                s.push(ch);
                else if(isoperand(ch))
                postfix +=ch;
                else if(isoperator(ch))
                {
                        while(!s.empty()&&eqlorhigher(s.top(),ch))
                        {
                                postfix +=s.top();
                                s.pop();
                        }
                        s.push(ch);
                }
                else if(ch==')')
                {
                        while(!s.empty() && s.top()!='(')
                        {
                                postfix += s.top();
                                s.pop();
                        }
                        s.pop();
                }
        }
        return postfix;
}
/*
Enter an infix expression: A+B*C+D

Your infix expression is: A+B*C+D
```

```
Postfix expression is:  ABC*+D+

Do you want to enter infix expression(1/0)?1
Enter an infix expression: (a+b)*(c+d)

Your infix expression is: (a+b)*(c+d)

Postfix expression is:  ab+cd+*

Do you want to enter infix expression(1/0)?1
Enter an infix expression: ((A-(B+C))*D)^(E+F)

Your infix expression is: ((A-(B+C))*D)^(E+F)

Postfix expression is:  ABC+-D*EF+^
*/
```

```
//Pizza parlor accepting maximum M orders. Orders are served in first come first served basis. Order
//once placed cannot be cancelled. Write C++ program to simulate the system using circular queue using
//array.

#include<iostream>
#include<windows.h>
using namespace std;
const int MAX=5;

class PizzaParlour
{
        int front,rear;
        int orders[MAX];
        public:
                PizzaParlour()
                {
                        front=rear=-1;
                }
                bool addOrder(int data);
                void serveOrder();
                void display();
};
bool PizzaParlour::addOrder(int id){
        if(rear==-1)
        {
                front=rear=0;
                orders[rear]=id;
                return true;
         }
         else
         {
                int pos=(rear+1)%MAX;
                if(pos==front)
                {
                        cout<<"\nCafe is Full.Please wait.\n";
                        return false;
                 }
                 else
                 {
                        rear=pos;
                        orders[rear]=id;
                        return true;
                 }
         }
 }

 void PizzaParlour::serveOrder()
 {
        if(front==-1)
        {
                cout<<"\n No Orders in Cafe.[Cafe is Empty)\n";
                return;
         }
         else
         {
                cout<<"\n Order No. "<<orders[front]<<" is processed.\n";
                if(front==rear) //only one order
                {
                        front=rear=-1;
                 }
                else
                {
                        front=(front+1)%MAX;
                }
        }
}

void PizzaParlour::display()
{
        int i=0;
        if(front==-1)
        {
                cout<<"\nCafe is Empty.No orders.\n";
                return;
        }
        else
        {
                cout<<"Order Id's: \n";
                for(i=front;i!=rear;i=((i+1)%MAX))
                {
                        cout<<orders[i]<<"  ";
                }
                cout<<orders[rear];
        }
}
void intro()
{       char name[50]={"\n Vaibhav Cafe \n"};
                for(int i=0;name[i]!='\0';i++)
        {
                Sleep(50);
                cout<<name[i];

        }
}
int main()
{
        int ch,id=0;

        PizzaParlour q;

        do
        {
                cout<<"\n-----------------";
        intro();
                cout<<"-----------------";
                cout<<"\n****Menu*****\n";
                cout<<"1. Accept order\n";
                cout<<"2. Serve order\n";
                cout<<"3. Display orders\n";
                cout<<"4. Exit";

                cout<<"\nChoice: ";
                cin>>ch;

                switch(ch)
                {
                case 1:
                                id++;
                                if(q.addOrder(id))
                                {
```

```
                    cout<<"Thank you for order.Order id is : "<<id;
                }
                else
                {
                    id--;
                }
                break;

        case 2: q.serveOrder();
                break;

        case 3: q.display();
                break;
        }
    }while(ch!=4);
    cout<<"\nThank You.Keep Visiting.";

}
```