Lab Report

Course title: Data Structure
Course code: CSE-156
1st Year 2nd Semester Examination 2020

Date of Submission: 02 March 2022

Submitted to-

Dr. Md. Golam Moazzam
Professor
Department of Computer Science and Engineering
Jahangirnagar University
Savar, Dhaka-1342

Sl	Class Roll	Exam Roll	Name
01	388	202200	MD. TANVIR HOSSAIN SAON

LAB Number - 1

Source Code:

```
#include<iostream>
#include<stdio.h>
#include<cmath>
using namespace std;

//////////////////////////////////////////////

void ls()
{
   int arr[1001],i,f,n,num,index;

   cout<<"\nHow Many Numbers:";
   cin>>n;

   cout<<"\nEnter the Numbers: ";
   for(i=0; i<n; i++)
   {
      cin>>arr[i];
   }
   cout<<"\nEnter a Number to Search: ";
   cin>>num;
   for(i=0; i<n; i++)
   {
```

```
        if(arr[i]==num)
        {
            index = i+1;
            f=0;
            break;
        }
    }
        if(f==0)
        {
            cout<<"\nFound at"<<index;
        }
        else{cout<<"\nNot Found!!!";}

    cout<<endl;
    }

////////////////////////////////////////////////////////////////////////////////

 void qe() {

    float a, b, c, x1, x2, discriminant, realPart, imaginaryPart;
    cout << "Enter coefficients a, b and c: ";
    cin >> a >> b >> c;
    discriminant = b*b - 4*a*c;

    if (discriminant > 0) {
        x1 = (-b + sqrt(discriminant)) / (2*a);
```

```
        x2 = (-b - sqrt(discriminant)) / (2*a);
        cout << "Roots are real and different." << endl;
        cout << "x1 = " << x1 << endl;
        cout << "x2 = " << x2 << endl;
    }

    else if (discriminant == 0) {
        cout << "Roots are real and same." << endl;
        x1 = -b/(2*a);
        cout << "x1 = x2 =" << x1 << endl;
    }

    else {
        realPart = -b/(2*a);
        imaginaryPart =sqrt(-discriminant)/(2*a);
        cout << "Roots are complex and different."  << endl;
        cout << "x1 = " << realPart << "+" << imaginaryPart << "i" << endl;
        cout << "x2 = " << realPart << "-" << imaginaryPart << "i" << endl;
    }

}

////////////////////////////////////////////////////////////////////////////////////////////

void le() {
  int n;
  double arr[10001];
```

```
  printf("Enter the number of elements: ");
  scanf("%d", &n);

  for (int i = 0; i < n; ++i) {
    printf("Enter number%d: ", i + 1);
    scanf("%lf", &arr[i]);
  }



  for (int i = 1; i < n; ++i) {
    if (arr[0] < arr[i]) {
      arr[0] = arr[i];
    }
  }

  printf("\nLargest element = %.2lf", arr[0]);
}
```

//

```
void fsle()

{

      int a[1001], i, largest = 0, second_largest = 0, pos1, pos2;
      int n;
      cout << "Enter Number of elements :";
```

```
cin>>n;
for (i = 0; i<n; ++i)
{
    cout << "Enter " << (i + 1) << "th Element :";
    cin >> a[i];
}

for (i = 0; i<n; ++i)
{
    if (a[i]>largest)
    {
        largest = a[i];
        pos1 = i;
    }
}

for (i = 0; i<n; ++i)
{
    if (a[i]>second_largest)
    {
        if (a[i] == largest)
            continue;
        second_largest = a[i];
        pos2 = i;
    }
}
cout << " Largest Number :" << largest << " at position " << (pos1 + 1)<<endl;
```

```
    cout << " Second Largest Number :"<< second_largest << " at position " << (pos2 +
1)<<endl;
}

int main()
{
   int a;
   while(1)
   {
      cout<<"-------------------------------------------------------------------------------"<<endl;
      cout<<"-------------------------------------------------------------------------------"<<endl;

        cout<<"Enter 1 For Linear Search"<<endl;
        cout<<"Enter 2 For Solution of Quadratic Equation"<<endl;
        cout<<"Enter 3 To Find Largest Element"<<endl;
        cout<<"Enter 4 For First Find largest and Second Largest element and their
Position"<<endl;
        cout<<"Enter 5 For Exit"<<endl;

      cout<<"-------------------------------------------------------------------------------"<<endl;
      cout<<"-------------------------------------------------------------------------------"<<endl;

      cout<<"Please Enter your choice"<<endl;

      cin>>a;
      if (a==1)
      {
```

```
        ls();
    }
    else if(a==2)
    {
        qe();
    }
    else if(a==3)
    {
        le();
    }
    else if(a==4)
    {
        fsle();
    }
    else if(a==5)
    {
        return 0;
    }

  }
}
```

Output:

1.For Linear search :

```
"C:\Users\USER\Documents\lab no 1.exe"
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Linear Search
Enter 2 For Solution of Quadratic Equation
Enter 3 To Find Largest Element
Enter 4 For First Find largest and Second Largest element and their Position
Enter 5 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
1

How Many Numbers:10

Enter the Numbers: 12 1 45 23 67 342 44 55 78 90

Enter a Number to Search: 23

Found at4
```

2.For Quadratic Equation:

```
"C:\Users\USER\Documents\lab no 1.exe"
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Linear Search
Enter 2 For Solution of Quadratic Equation
Enter 3 To Find Largest Element
Enter 4 For First Find largest and Second Largest element and their Position
Enter 5 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
2
Enter coefficients a, b and c:  2 4 2
Roots are real and same.
x1 = x2 =-1
```

3.Find Largest Element:

```
"C:\Users\USER\Documents\lab no 1.exe"
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Linear Search
Enter 2 For Solution of Quadratic Equation
Enter 3 To Find Largest Element
Enter 4 For First Find largest and Second Largest element and their Position
Enter 5 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
3
Enter the number of elements: 4
Enter number1: 123
Enter number2: 23
Enter number3: 123
Enter number4: 56

Largest element = 123.00
```

4.Find Largest and second Largest element and their position :

```
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Linear Search
Enter 2 For Solution of Quadratic Equation
Enter 3 To Find Largest Element
Enter 4 For First Find largest and Second Largest element and their Position
Enter 5 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
4
Enter Number of elements :4
Enter 1th Element :12
Enter 2th Element :345
Enter 3th Element :123
Enter 4th Element :45
 Largest Number :345 at position 2
 Second Largest Number :123 at position 3
```

LAB Number - 2

Source Code:

```
#include<bits/stdc++.h>

using namespace std;


void insert_string()

{

        char a[10];

        char b[10];

        char c[10];

        int p=0,r=0,i=0;

        int t=0;

        int x,g,s,n,o;

        puts("Enter First String:");

        gets(a);

        puts("Enter Second String:");

        gets(b);

        printf("Enter the position where the item has to be inserted: ");

        scanf("%d",&p);

        r = strlen(a);

        n = strlen(b);

        i=0;

        while(i <= r)

        {

                c[i]=a[i];
```

```
            i++;
        }
        s = n+r;
        o = p+n;

        for(i=p;i<s;i++)
        {
            x = c[i];
            if(t<n)
            {
                a[i] = b[t];
                t=t+1;
            }
            a[o]=x;
            o=o+1;
        }

        printf("%s", a);
}

//////////////////////////////////////////////////////////////////////////////

void len() {
    char str[10000];
    cin>>str;
    cout << "String Length = " << strlen(str);
```

```
}

//////////////////////////////////////////////////////////////////////////////

void concate()
{
   string s1, s2, result;
   cout << "Enter string s1: ";
   getline (cin, s1);
   cout << "Enter string s2: ";
   getline (cin, s2);
   result = s1 + s2;
   cout << "Resultant String = "<< result;

}

//////////////////////////////////////////////////////////////////////////////

int main()
{
   int a;
   while(1)
   {
      cout<<"\n-------------------------------------------------------------------------------"<<endl;
      cout<<"-------------------------------------------------------------------------------"<<endl;

        cout<<"Enter 1 For Insert a String "<<endl;
```

```
  cout<<"Enter 2 For Lenth of the String"<<endl;
  cout<<"Enter 3 For Concatanation "<<endl;
  cout<<"Enter 4 For Exit"<<endl;

cout<<"-------------------------------------------------------------------------------"<<endl;
cout<<"-------------------------------------------------------------------------------"<<endl;


cout<<"Please Enter your choice"<<endl;

cin>>a;
if (a==1)
{
   insert_string();
}
else if(a==2)
{
   len();
}
else if(a==3)
{
   concate();
}
else if(a==4)
{
    return 0;
}
```

```
    }

}
```

Output:

```
C:\Users\USER\Documents\dummy.exe
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Insert a String
Enter 2 For Lenth of the String
Enter 3 For Concatanation
Enter 4 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
2
saon
String Length = 4
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Insert a String
Enter 2 For Lenth of the String
Enter 3 For Concatanation
Enter 4 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
```

LAB Number - 3

Source Code:

```
#include<iostream>
using namespace std;
void quick_sort(int[],int,int);
int partition(int[],int,int);
void bs()
{
   int n, i, arr[1001], j, temp;
   cout<<"Enter the Size : ";
   cin>>n;
   cout<<"Enter "<<n<<" Numbers: ";
   for(i=0; i<n; i++)
      cin>>arr[i];

   for(i=0; i<(n-1); i++)
   {
      for(j=0; j<(n-i-1); j++)
      {
         if(arr[j]>arr[j+1])
         {
            temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp;
         }
```

```
        }
    }
    cout<<"\nThe New Array is: \n";
    for(i=0; i<n; i++)
        cout<<arr[i]<<" ";
    cout<<endl;
}
//////////////////////////////////////////////////////////////
void qs()
{
    int a[50],n,i;
    cout<<"How many elements?\n";
    cin>>n;
    cout<<"\nEnter array elements:\n";

    for(i=0;i<n;i++)
        cin>>a[i];

    quick_sort(a,0,n-1);
    cout<<"\nArray after sorting:";

    for(i=0;i<n;i++)
        cout<<a[i]<<" ";
}

void quick_sort(int a[],int l,int u)
{
```

```
    int j;
    if(l<u)
    {
        j=partition(a,l,u);
        quick_sort(a,l,j-1);
        quick_sort(a,j+1,u);
    }
}

int partition(int a[],int l,int u)
{
    int v,i,j,temp;
    v=a[l];
    i=l;
    j=u+1;

    do
    {
        do
            i++;

        while(a[i]<v&&i<=u);

        do
            j--;
        while(v<a[j]);
```

```
        if(i<j)

        {

            temp=a[i];

            a[i]=a[j];

            a[j]=temp;

        }

    }while(i<j);


    a[l]=a[j];

    a[j]=v;


    return(j);

}

////////////////////////////////////////////////////////////



void bss()

{

    int i, arr[1001], num,n, first, last, middle;

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

    cout<<"Enter  Elements in ascending order: ";

    for(i=0; i<n; i++)

        cin>>arr[i];

    cout<<"\nEnter Element to be Search: ";

    cin>>num;

    first = 0;
```

```
   last = n-1;
   middle = (first+last)/2;
   while(first <= last)
   {
      if(arr[middle]<num)
         first = middle+1;
      else if(arr[middle]==num)
      {
         cout<<"\nThe number, "<<num<<" found at Position "<<middle+1;
         break;
      }
      else
         last = middle-1;
         middle = (first+last)/2;
   }
   if(first>last)
      cout<<"\nThe number, "<<num<<" is not found in given Array";
   cout<<endl;

}

//////////////////////////////////////////////////////////////////////////////



void mm()
{
int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
```

```
cout<<"enter the number of row=";
cin>>r;
cout<<"enter the number of column=";
cin>>c;
cout<<"enter the first matrix element=\n";
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
cin>>a[i][j];
}
}
cout<<"enter the second matrix element=\n";
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
cin>>b[i][j];
}
}
cout<<"multiply of the matrix=\n";
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
mul[i][j]=0;
for(k=0;k<c;k++)
```

```
{
mul[i][j]+=a[i][k]*b[k][j];
}
}
}

for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
cout<<mul[i][j]<<" ";
}
cout<<"\n";
}
}

////////////////////////////////////////////////////////////////////////////////

int main()
{
   int a;
   while(1)
   {
      cout<<"\n-------------------------------------------------------------------------------"<<endl;
      cout<<"-------------------------------------------------------------------------------"<<endl;
        cout<<"Enter 1 For Bubble Sort "<<endl;
        cout<<"Enter 2 For Quick Sort"<<endl;
```

```
  cout<<"Enter 3 For Binary Search"<<endl;
  cout<<"Enter 4 For Matrix Multiplication"<<endl;
  cout<<"Enter 5 For Exit"<<endl;
cout<<"-------------------------------------------------------------------------------"<<endl;
cout<<"-------------------------------------------------------------------------------"<<endl;
cout<<"Please Enter your choice"<<endl;
cin>>a;
if (a==1)
{
   bs();
}
else if(a==2)
{
   qs();
}
else if(a==3)
{
   bss();
}
else if(a==4)
{
   mm();
}
else if(a==5)
{
   return 0;
}
```

```
  }
}
```

Output:

1 . Bubble Sort

```
"C:\Users\USER\Documents\lab 2.exe"

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Bubble Sort
Enter 2 For Quick Sort
Enter 3 For Binary Search
Enter 4 For Matrix Multiplication
Enter 5 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
1
Enter the Size : 5
Enter 5 Numbers: 12 3 22 34 5

The New Array is:
3 5 12 22 34
```

2 Quick Sort

```
"C:\Users\USER\Documents\lab 2.exe"

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Bubble Sort
Enter 2 For Quick Sort
Enter 3 For Binary Search
Enter 4 For Matrix Multiplication
Enter 5 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
2
How many elements?
4

Enter array elements:
12 1 22 4

Array after sorting:1 4 12 22
```

3. Binary Search

```
"C:\Users\USER\Documents\lab 2.exe"

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Bubble Sort
Enter 2 For Quick Sort
Enter 3 For Binary Search
Enter 4 For Matrix Multiplication
Enter 5 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
3
Enter the number of elements
5
Enter  Elements in ascending order:  2 4 5 6 7

Enter Element to be Search: 5

The number, 5 found at Position 3
```

4. Matrix Multiplication

```
"C:\Users\USER\Documents\lab 2.exe"
Enter 1 For Bubble Sort
Enter 2 For Quick Sort
Enter 3 For Binary Search
Enter 4 For Matrix Multiplication
Enter 5 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
4
enter the number of row=2
enter the number of column=2
enter the first matrix element=
1
2
3
4
enter the second matrix element=
2
3
4
5
multiply of the matrix=
10 13
22 29
```

LAB Number - 4

Source Code:

```
#include<bits/stdc++.h>

using namespace std;

struct node

{

   int data;

   node *link;

};


node *head;

class linkedlist

{

public :



   void Insert(int val)

   {

      node *newnode=new node();

      newnode->data=val;

      newnode->link=NULL;

      if(head==NULL)

      {

         head=newnode;

      }
```

```
    else
    {
        node *temp=head;
        while(temp->link!=NULL)
        {
            temp=temp->link;
        }
        temp->link=newnode;
    }

}

void create()
{
    int num,val,i;
    cout<<"Number of element :  "<<endl;
    cin>>num;
    cout<<"Enter elements : "<<endl;
    for(i=0; i<num; i++)
    {

        cin>>val;
        Insert(val);
    }
}

void print()
```

```
{
   node *temp=head;
   cout<<"The Linked list is: ";
   while(temp!=NULL)
   {
      cout<<temp->data<<" ";
      temp=temp->link;
   }
   cout<<endl;
}

void Insert_at_pos()
{
   int val,pos;
   cout<<"Enter value : "<<endl;
   cin>>val;
   cout<<"Enter position : "<<endl;
   cin>>pos;

   node *newnode=new node();

   newnode->data=val;
   newnode->link=NULL;

   if(pos==1)
   {
      newnode->link=head;
```

```
        head=newnode;


    }
    else
    {
        node  *temp=new node();
        temp=head;
        for(int i=1; i<pos-1; i++)
        {
            temp=temp->link;
        }
        newnode->link=temp->link;
        temp->link=newnode;
    }
}
void Delete()
{
    int pos;
    cout<<"Enter position : "<<endl;
    cin>>pos;
    node *pre=head;
    for(int i=1; i<pos-1; i++)
        pre=pre->link;
    node *curr=pre->link;
    pre->link=curr->link;
    free(curr);
```

```
    }
};
int main()
{
    linkedlist call;
    int val,pos,i;
    while(1)
    {
        cout<<"--------------------"<<endl;
        cout<<"\n1.Create a linked list."<<endl;
        cout<<"2.Display."<<endl;
        cout<<"3.Insert at any position."<<endl;
        cout<<"4.Delete"<<endl;
        cout<< "5.Finish "<<endl;
         cout<<"\n--------------------"<<endl;
        int i;
        cin>>i;
        if(i==1)call.create();
        if(i==2)call.print();
        if(i==3)call.Insert_at_pos();
        if(i==4)call.Delete();
        if(i==5) return 0;

    }
    return 0;
}
```

Output:

1. Create & Display

```
"C:\Users\USER\Documents\lab 4.exe"
--------------------

1.Create a linked list.
2.Display.
3.Insert at any position.
4.Delete
5.Finish

--------------------
1
Number of element :
4
Enter elements :
1 2 3 4
--------------------

1.Create a linked list.
2.Display.
3.Insert at any position.
4.Delete
5.Finish

--------------------
2
The Linked list is: 1 2 3 4
--------------------
```

2. Insert Element

```
--------------------

1.Create a linked list.
2.Display.
3.Insert at any position.
4.Delete
5.Finish

--------------------
3
Enter value :
6
Enter position :
3
--------------------

1.Create a linked list.
2.Display.
3.Insert at any position.
4.Delete
5.Finish

--------------------
2
The Linked list is: 1 2 6 3 4
--------------------
```

3.Delete a element

```
--------------------

1.Create a linked list.
2.Display.
3.Insert at any position.
4.Delete
5.Finish

--------------------
4
Enter position :
1
--------------------

1.Create a linked list.
2.Display.
3.Insert at any position.
4.Delete
5.Finish

--------------------
2
The Linked list is: 1 6 3 4
--------------------
```

LAB Number - 5

Source Code:

```
#include<bits/stdc++.h>

using namespace std;

int stac[100],i,j,choice=0,n,top=-1;
```

```
void push();
void pop();
void show();
#define MAX 50
void push1();
void pop1();
void display();
int queue_array[MAX];
int rear = - 1;
int front = - 1;

//////////////////////////////////////////////////////////

void st()
{

   printf("Enter the number of elements in the stack :\n");
   scanf("%d",&n);
     cout<<"------------------------------"<<endl;
     cout<<"      Stack operations         "<<endl;
     cout<<"------------------------------"<<endl;
   while(choice != 4)
   {
      printf("Chose one from the below options...\n");
      printf("\n1.Push\n2.Pop\n3.Show\n4.Exit");
      printf("\n Enter your choice \n");
      scanf("%d",&choice);
```

```
switch(choice)
{
   case 1:
   {
      push();
      break;
   }
   case 2:
   {
      pop();
      break;
   }
   case 3:
   {
      show();
      break;
   }
   case 4:
   {
      printf("Exiting....");
      break;
   }
   default:
   {
      printf("Please Enter valid choice ");
   }
};
```

```
    }
}

void push ()
{
    int val;
    if (top == n-1 )
    printf("Overflow\n ");
    else
    {
        printf("Enter the value:\n");
        scanf("%d",&val);
        top = top +1;
        stac[top] = val;
    }
}

void pop ()
{
    if(top == -1)
    printf("Underflow\n");
    else
    top = top -1;
}
void show()
{
    for (i=top;i>=0;i--)
```

```
    {
        cout<<stac[i]<<" ";
    }
    if(top == -1)
    {
        printf("Stack is empty");
    }
}

////////////////////////////////////////////////////////////////////////


void qu()
{
    int choice;
    while (1)
    {   cout<<"------------------------------"<<endl;
        cout<<"      Queue operations         "<<endl;
        cout<<"------------------------------"<<endl;
        printf("1.Push element to queue \n");
        printf("2.Pop element from queue \n");
        printf("3.Display all elements of queue \n");
        printf("4.Quit \n");
        printf("Enter your choice : ");
        scanf("%d", &choice);
        switch (choice)
        {
```

```
        case 1:
        push();
        break;
        case 2:
        pop();
        break;
        case 3:
        display();
        break;
        case 4:
        exit(1);
        default:
        printf("Wrong choice \n");
      }
    }
}

void push1()
{
    int add_item;
    if (rear == MAX - 1)
    printf("Queue Overflow \n");
    else
    {
      if (front == - 1)
      front = 0;
      printf("Inset the element in queue : ");
```

```
        scanf("%d", &add_item);
        rear = rear + 1;
        queue_array[rear] = add_item;
    }
}

void pop1()
{
    if (front == - 1 || front > rear)
    {
        printf("Queue Underflow \n");
        return ;
    }
    else
    {
        printf("Element deleted from queue is : %d\n", queue_array[front]);
        front = front + 1;
    }
}

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

```
        printf("Queue is : \n");
        for (i = front; i <= rear; i++)
            printf("%d ", queue_array[i]);
        printf("\n");
    }
}



////////////////////////////////////////////////////////////////////////////////

void TOH(int n,char Sour, char Aux,char Des)
{
        if(n==1)
        {
                cout<<"Move Disk "<<n<<" from "<<Sour<<" to "<<Des<<endl;
                return;
        }

        TOH(n-1,Sour,Des,Aux);
        cout<<"Move Disk "<<n<<" from "<<Sour<<" to "<<Des<<endl;
        TOH(n-1,Aux,Sour,Des);
}

void th()
{
        int n;
```

```
	cout<<"Enter no. of disks:";
	cin>>n;

	TOH(n,'A','B','C');
}
//////////////////////////////////////////////////////////////////////////////
int prec(char c) {
	if(c == '^')
		return 3;
	else if(c == '/' || c=='*')
		return 2;
	else if(c == '+' || c == '-')
		return 1;
	else
		return -1;
}

void infixToPostfix(string s) {

	stack<char> st;
	string result;

	for(int i = 0; i < s.length(); i++) {
		char c = s[i];
		if((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') || (c >= '0' && c <= '9'))
			result += c;
		else if(c == '(')
```

```
                st.push('(');
            else if(c == ')') {
                while(st.top() != '(')
                {
                    result += st.top();
                    st.pop();
                }
                st.pop();
            }

            else {
                while(!st.empty() && prec(s[i]) <= prec(st.top())) {
                    result += st.top();
                    st.pop();
                }
                st.push(c);
            }
        }
        while(!st.empty()) {
            result += st.top();
            st.pop();
        }

        cout << result << endl;
}
void ip() {
        string exp;
```

```
        cout<<"Enter the Expression:"<<endl;
        cin>>exp;
        infixToPostfix(exp);
}



////////////////////////////////////////////////////////////////////////////////
int main()
{
   int a;
   while(1)
   {
      cout<<"\n--------------------------------------------------------------------------------"<<endl;
      cout<<"--------------------------------------------------------------------------------"<<endl;

        cout<<"Enter 1 For Stack "<<endl;
        cout<<"Enter 2 For Queue"<<endl;
        cout<<"Enter 3 For Tower of hanoi"<<endl;
        cout<<"Enter 4 For infix to postfix"<<endl;
        cout<<"Enter 5 For Exit"<<endl;

      cout<<"--------------------------------------------------------------------------------"<<endl;
      cout<<"--------------------------------------------------------------------------------"<<endl;



      cout<<"Please Enter your choice"<<endl;
```

```
    cin>>a;
    if (a==1)
    {
      st() ;
    }
    else if(a==2)
    {
      qu();
    }
    else if(a==3)
    {
      th();
    }
    else if(a==4)
    {
      ip();
    }
    else if(a==5)
    {
       return 0;
    }
}
}
```

Output:

```
"C:\Users\USER\Documents\lab 5.exe"
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Stack
Enter 2 For Queue
Enter 3 For Tower of hanoi
Enter 4 For infix to postfix
Enter 5 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
1
Enter the number of elements in the stack :
4
-----------------------------
      Stack operations
-----------------------------
Chose one from the below options...

1.Push
2.Pop
3.Show
4.Exit
 Enter your choice
1
Enter the value:
3
Chose one from the below options...

1.Push
2.Pop
3.Show
4.Exit
 Enter your choice
3
3 Chose one from the below options...

1.Push
2.Pop
3.Show
4.Exit
 Enter your choice
```

```
"C:\Users\USER\Documents\lab 5.exe"
--------------------------------------------------------
Please Enter your choice
2
-----------------------------
      Queue operations
-----------------------------
1.Push element to queue
2.Pop element from queue
3.Display all elements of queue
4.Quit
Enter your choice :
1
Overflow
 -----------------------------
      Queue operations
-----------------------------
1.Push element to queue
2.Pop element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 3
Queue is empty
-----------------------------
      Queue operations
-----------------------------
1.Push element to queue
2.Pop element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 2
Underflow
-----------------------------
      Queue operations
-----------------------------
1.Push element to queue
2.Pop element from queue
3.Display all elements of queue
4.Quit
Enter your choice :
```

```
"C:\Users\USER\Documents\lab 5.exe"

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Stack
Enter 2 For Queue
Enter 3 For Tower of hanoi
Enter 4 For infix to postfix
Enter 5 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
3
Enter no. of disks:4
Move Disk 1 from A to B
Move Disk 2 from A to C
Move Disk 1 from B to C
Move Disk 3 from A to B
Move Disk 1 from C to A
Move Disk 2 from C to B
Move Disk 1 from A to B
Move Disk 4 from A to C
Move Disk 1 from B to C
Move Disk 2 from B to A
Move Disk 1 from C to A
Move Disk 3 from B to C
Move Disk 1 from A to B
Move Disk 2 from A to C
Move Disk 1 from B to C

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Stack
Enter 2 For Queue
Enter 3 For Tower of hanoi
Enter 4 For infix to postfix
Enter 5 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
```

```
"C:\Users\USER\Documents\lab 5.exe"

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Stack
Enter 2 For Queue
Enter 3 For Tower of hanoi
Enter 4 For infix to postfix
Enter 5 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
4
Enter the Expression:
a+b*(c^d-e)^(f+g*h)-i
abcd^e-fgh*+^*+i-

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Stack
Enter 2 For Queue
Enter 3 For Tower of hanoi
Enter 4 For infix to postfix
Enter 5 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
```

LAB Number - 6

Source Code:

```
#include<bits/stdc++.h>

using namespace std;

struct node

 {
```

```
  int data;
  struct node *left;
  struct node *right;
 };
struct node *newNode(int data)
   {
  struct node *node = (struct node *)malloc(sizeof(struct node));
  node->data = data;
  node->left = NULL;
  node->right = NULL;
  return (node);
  }
void traversePreOrder(struct node *temp) {
  if (temp != NULL) {
    cout << " " << temp->data;
    traversePreOrder(temp->left);
    traversePreOrder(temp->right);
  }
}
void traverseInOrder(struct node *temp) {
  if (temp != NULL) {
    traverseInOrder(temp->left);
    cout << " " << temp->data;
    traverseInOrder(temp->right);
  }
}
```

```
void traversePostOrder(struct node *temp) {
  if (temp != NULL) {
    traversePostOrder(temp->left);
    traversePostOrder(temp->right);
    cout << " " << temp->data;
  }
}

void bt() {
  struct node *root = newNode(1);
  root->left = newNode(2);
  root->right = newNode(3);
  root->left->left = newNode(4);
  cout << "preorder traversal: ";
  traversePreOrder(root);
  cout << "\nInorder traversal: ";
  traverseInOrder(root);
  cout << "\nPostorder traversal: ";
  traversePostOrder(root);
}

////////////////////////////////////////////////////////////////////////////////

struct nod {
  int key;
```

```
  struct nod *left, *right;
};
struct nod *newNode(int item) {
  struct nod *temp = (struct nod *)malloc(sizeof(struct nod));
  temp->key = item;
  temp->left = temp->right = NULL;
  return temp;
}
void inorder(struct nod *root) {
  if (root != NULL) {
 inorder(root->left);
 cout << root->key << " -> ";
inorder(root->right);
  }
}
struct nod *insert(struct nod *nod, int key) {
  if (nod == NULL) return newNode(key);
  if (key < nod->key)
    nod->left = insert(nod->left, key);
  else
    nod->right = insert(nod->right, key);
  return nod;
}

struct nod *minValueNode(struct nod *nod) {
  struct nod *current = nod;
```

```
  while (current && current->left != NULL)
    current = current->left;
  return current;
}
void bst() {
  struct node *root = NULL;
  root = insert(root, 8);
  root = insert(root, 3);
  root = insert(root, 1);
  root = insert(root, 6);
  root = insert(root, 7);
  root = insert(root, 10);
  root = insert(root, 14);
  root = insert(root, 4);
  cout << "Inorder traversal: ";
  inorder(root);
  cout << "Inorder traversal: ";
  inorder(root);
}
////////////////////////////////////////////////////////////////////////////////
void swap(int *a, int *b)
{
  int temp = *b;
  *b = *a;
  *a = temp;
}
```

```
void heapify(vector<int> &hT, int i)
{
  int size = hT.size();
  int largest = i;
  int l = 2 * i + 1;
  int r = 2 * i + 2;
  if (l < size && hT[l] > hT[largest])
    largest = l;
  if (r < size && hT[r] > hT[largest])
    largest = r;

  if (largest != i)
  {
    swap(&hT[i], &hT[largest]);
    heapify(hT, largest);
  }
}
void insert(vector<int> &hT, int newNum)
{
  int size = hT.size();
  if (size == 0)
  {
    hT.push_back(newNum);
  }
  else
  {
```

```
    hT.push_back(newNum);
    for (int i = size / 2 - 1; i >= 0; i--)
    {
      heapify(hT, i);
    }
  }
}
void printArray(vector<int> &hT)
{
  for (int i = 0; i < hT.size(); ++i)
    cout << hT[i] << " ";
  cout << "\n";
}
void hp()
{
  vector<int> heapTree;
  insert(heapTree, 3);
  insert(heapTree, 4);
  insert(heapTree, 9);
  insert(heapTree, 5);
  insert(heapTree, 2);
  cout << "Max-Heap array: ";
  printArray(heapTree);


}
```

```
//////////////////////////////////////////////////////////////////////

int main()
{
   int a;
   while(1)
   {
      cout<<"\n-------------------------------------------------------------------------------"<<endl;
      cout<<"-------------------------------------------------------------------------------"<<endl;
        cout<<"Enter 1 For Binary Tree Creation & Traversal "<<endl;
        cout<<"Enter 2 For Binary Search Tree"<<endl;
        cout<<"Enter 3 For Heap"<<endl;
        cout<<"Enter 4 For Exit"<<endl;
      cout<<"-------------------------------------------------------------------------------"<<endl;
      cout<<"-------------------------------------------------------------------------------"<<endl;
      cout<<"Please Enter your choice"<<endl;
      cin>>a;
      if (a==1)
      {
        bt();
      }
      else if(a==2)
      {
        bst();
      }
      else if(a==3)
```

```
    {
      hp();
    }
    else if(a==4)
    {
       return 0;
    }
  }
}
```

Output:

"C:\Users\USER\Documents\lab 6.exe"

```
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Binary Tree Creation & Traversal
Enter 2 For Binary Search Tree
Enter 3 For Heap
Enter 4 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
1
preorder traversal:  1 2 4 3
Inorder traversal:  4 2 1 3
Postorder traversal:  4 2 3 1
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Enter 1 For Binary Tree Creation & Traversal
Enter 2 For Binary Search Tree
Enter 3 For Heap
Enter 4 For Exit
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Please Enter your choice
3
Max-Heap array: 9 5 3 4 2
```