//SE 47 (Stack and Queue)

#include<iostream>

#include<stack>

#include<queue>

using namespace std;

void showStack(stack <int> s)

{

while(!s.empty())

{

cout<<s.top();

s.pop();

cout<<" ";

}

cout<<endl;

}

void showQueue(queue <int> q)

{

while (!q.empty())

{

cout<<q.front();

q.pop();

cout<<" ";

}

cout<<endl;

}

int main()

{

int elements;

int i;

int choice;

int stackChoice;

int queueChoice;

stack <int> s;

queue <int> q;

do

{

cout<<endl;

cout<<"\t\t\*\*\*\*Menu\*\*\*\*";

cout<<"\n\t\t1. Stack";

cout<<"\n\t\t2. Queue";

cout<<"\n\t\t3. Exit";

cout<<"\nEnter the choice : ";

cin>>choice;

switch(choice)

{

case 1:

do

{

cout<<"\nOperations of Stack : ";

cout<<"\n\t\t1. Push an Element";

cout<<"\n\t\t2. Pop an Element";

cout<<"\n\t\t3. Display Stack";

cout<<"\n\t\t4. Exit";

cout<<"\nSelect Stack Operation : ";

cin>>stackChoice;

switch(stackChoice)

{

case 1:

cout<<"Enter the number of Elements to be Added : ";

cin>>elements;

while(elements!=0)

{

cout<<"Enter the Element : ";

cin>>i;

s.push(i);

--elements;

}

break;

case 2:

cout<<"The Deleted Element is : ";

cout<<s.top();

s.pop();

break;

case 3:

cout << "The stack is : ";

showStack(s);

}

}while(stackChoice>=1 && stackChoice<4);

break;

case 2:

do

{

cout<<"\nOperations of Queue : ";

cout<<"\n\t\t1. Push an Element";

cout<<"\n\t\t2. Pop an Element";

cout<<"\n\t\t3. Display Queue";

cout<<"\n\t\t4. Exit";

cout<<"\nSelect Queue Operation : ";

cin>>queueChoice;

switch(queueChoice)

{

case 1:

cout<<"Enter the number of Elements to be Added : ";

cin>>elements;

while(elements!=0)

{

cout<<"Enter the Element : ";

cin>>i;

q.push(i);

--elements;

}

break;

case 2:

cout<<"The Deleted Element is : ";

cout<<q.front();

q.pop();

break;

case 3:

cout << "The Queue is : ";

showQueue(q);

}

}while(queueChoice>=1 && queueChoice<4);

break;

case 3:

break;

}

}while(choice>=1 && choice<4);

}

\*\*\*\*Menu\*\*\*\*

1. Stack

2. Queue

3. Exit

Enter the choice : 1

Operations of Stack :

1. Push an Element

2. Pop an Element

3. Display Stack

4. Exit

Select Stack Operation : 1

Enter the number of Elements to be Added : 2

Enter the Element : 1

Enter the Element : 2

Operations of Stack :

1. Push an Element

2. Pop an Element

3. Display Stack

4. Exit

Select Stack Operation : 3

The stack is : 2 1

Operations of Stack :

1. Push an Element

2. Pop an Element

3. Display Stack

4. Exit

Select Stack Operation : 2

The Deleted Element is : 2

Operations of Stack :

1. Push an Element

2. Pop an Element

3. Display Stack

4. Exit

Select Stack Operation : 4

\*\*\*\*Menu\*\*\*\*

1. Stack

2. Queue

3. Exit

Enter the choice : 2

Operations of Queue :

1. Push an Element

2. Pop an Element

3. Display Queue

4. Exit

Select Queue Operation : 1

Enter the number of Elements to be Added : 3

Enter the Element : 1

Enter the Element : 2

Enter the Element : 3

Operations of Queue :

1. Push an Element

2. Pop an Element

3. Display Queue

4. Exit

Select Queue Operation : 3

The Queue is : 1 2 3

Operations of Queue :

1. Push an Element

2. Pop an Element

3. Display Queue

4. Exit

Select Queue Operation : 2

The Deleted Element is : 1

Operations of Queue :

1. Push an Element

2. Pop an Element

3. Display Queue

4. Exit

Select Queue Operation : 3

The Queue is : 2 3

Operations of Queue :

1. Push an Element

2. Pop an Element

3. Display Queue

4. Exit

Select Queue Operation : 4

\*\*\*\*Menu\*\*\*\*

1. Stack

2. Queue

3. Exit

Enter the choice : 3