==========

Roll Number: SYCOC303 Division: C

PRN Number: 122B2B303 Batch: C4

Name: VINAYAK MADAN SHETE

==========

Problem Statement:

- **Write a C++ program to implement the following data structures and its operations using linked list:**
  - **i) Stack**
  - **ii) Queue**

==========

INPUT: 1)STACK

```
/*
 *     ====================================
 *            Program Name: StackMain.cpp
 *     Created on: November 23, 2022
 *       Author: Vinayak Shete
 *  ====================================
 */

#include <iostream>
#include <stdlib.h>

using namespace std;

struct Node
{
	int data;
	struct Node *next;
};

class Stack
```

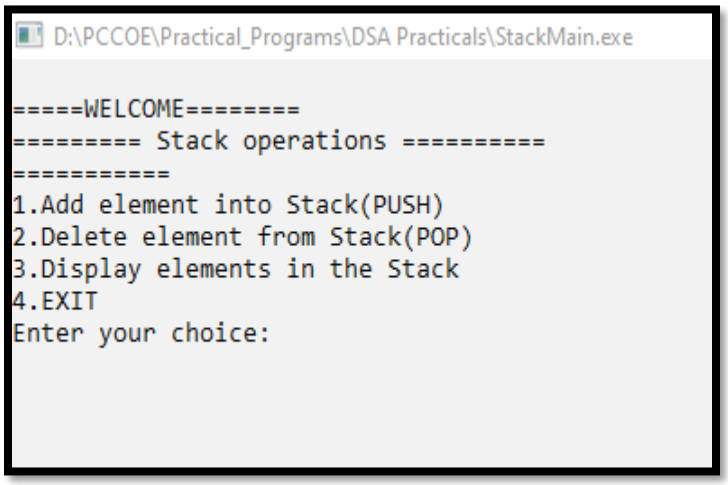
```
{
	public:
		struct Node *top;

		void push(int dt)
		{
			struct Node *nn = (struct Node*) malloc(sizeof(struct Node));
			nn->data=dt;
			nn->next= top;
			top=nn;
		}

		void pop()
		{
			if(top==NULL)
			{
				cout<<"\nStack underflow!POP operation failed!";
			}
			else
			{
				Node *tmp=top;
				int dt = top->data;
				top=top->next;
				free(tmp);
				cout<<"\nAn element "<<dt<<" which was at TOP has been
Deleted!";
			}
		}

		void display()
		{
			if(top==NULL)
			{
				cout<<"\nStack is empty!Display Operation failed!";
			}
			else
			{
```

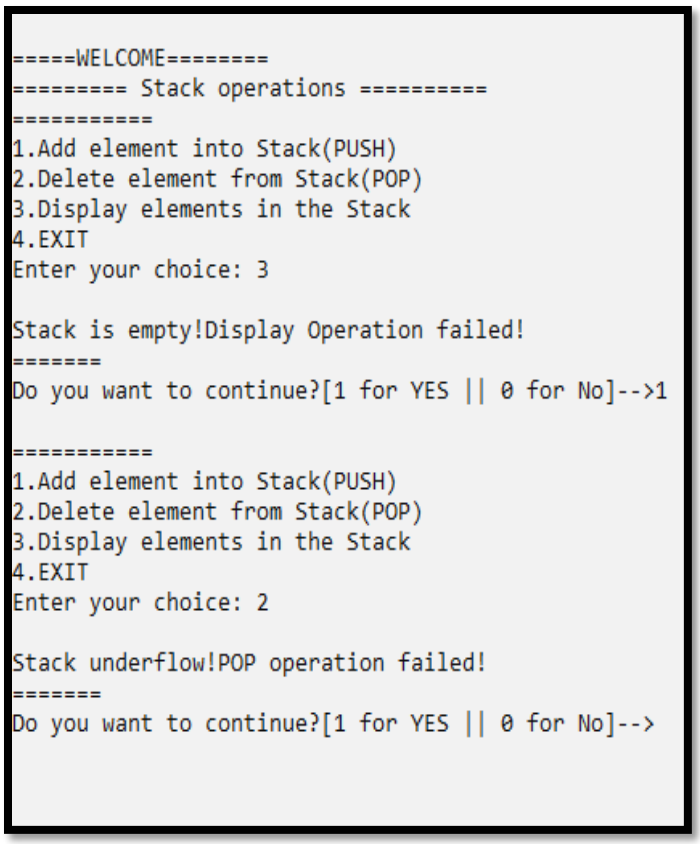
```
                        Node *tmp=top;
                        cout<<"\nStack elements are: ";
                        while(tmp!=NULL)
                        {
                                cout<<tmp->data<<" ";
                                tmp=tmp->next;
                        }
                }
        }
};

int main()
{
        Stack st;
        int val,ch,doch;
        cout<<"\n=====WELCOME========";
        cout<<"\n========= Stack operations ==========";
        do
        {
                cout<<"\n===========";
                cout<<"\n1.Add element into Stack(PUSH)\n2.Delete element from
Stack(POP)\n3.Display elements in the Stack\n4.EXIT";
                cout<<"\nEnter your choice: ";
                cin>>ch;
                switch(ch)
                {
                        case 1:
                                cout<<"Enter any number to push: ";
                                cin>>val;
                                st.push(val);
                                break;

                        case 2:
                                st.pop();
                                break;

                        case 3:
```

```
                        st.display();
                        break;
                case 4:
                        goto exit;
                        break;
                default:
                        cout<<"\nPlease enter correct choice!";
                        break;
            }
            cout<<"\n=======\nDo you want to continue?[1 for YES || 0 for No]-->";
            cin>>doch;
        }while(doch==1);
        exit:
            cout<<"\n=====THANK YOU!=======";
        return 0;
    }
```

==========

OUTPUT:





```
=====WELCOME========
========== Stack operations ==========
===========
1.Add element into Stack(PUSH)
2.Delete element from Stack(POP)
3.Display elements in the Stack
4.EXIT
Enter your choice: 1
Enter any number to push: 10

=======
Do you want to continue?[1 for YES || 0 for No]-->1

===========
1.Add element into Stack(PUSH)
2.Delete element from Stack(POP)
3.Display elements in the Stack
4.EXIT
Enter your choice: 1
Enter any number to push: 20

=======
Do you want to continue?[1 for YES || 0 for No]-->1

===========
1.Add element into Stack(PUSH)
2.Delete element from Stack(POP)
3.Display elements in the Stack
4.EXIT
Enter your choice: 1
Enter any number to push: 30
```

```
=======
Do you want to continue?[1 for YES || 0 for No]-->1

===========
1.Add element into Stack(PUSH)
2.Delete element from Stack(POP)
3.Display elements in the Stack
4.EXIT
Enter your choice: 3

Stack elements are: 40 30 20 10
=======
Do you want to continue?[1 for YES || 0 for No]-->
```

```
===========
1.Add element into Stack(PUSH)
2.Delete element from Stack(POP)
3.Display elements in the Stack
4.EXIT
Enter your choice: 2

An element 40 which was at TOP has been Deleted!
=======
Do you want to continue?[1 for YES || 0 for No]-->
```

==========

INPUT: 2)QUEUE

```
/*
 *     ====================================
 *            Program Name: QueueMain.cpp
 *     Created on: November 23, 2022
 *       Author: Vinayak Shete
 *  ====================================
 */

#include <iostream>
using namespace std;

struct Node
{
	int data;
	struct Node *next;
};

class Queue
{
	struct Node *front= NULL;
	struct Node  *rear = NULL;
	public:
		void insert(int dt)
		{
			struct Node *nn = new Node();
			nn->data=dt;
			nn->next=NULL;

			if(front==NULL && rear==NULL)
			{
				front=nn;
				rear=nn;
			}
```

```
		else
		{
				rear->next=nn;
				rear=nn;
		}
	}

	void dequeue()
	{
		if(front==NULL)
		{
			cout<<"\nQueue is empty!Deletion Operation failed!";
		}
		else
		{
			struct Node *tmp=front;
			front=front->next;
			cout<<"\nAn element "<<tmp->data<<" which was at FRONT has
been Deleted!";
		}
	}

	void display()
	{
		if(front==NULL)
		{
			cout<<"\nQueue is empty!Display Operation failed!";
		}
		else
		{
			struct Node *tmp=front;
			cout<<"\nThe elements in the Queue are: \n";
			while(tmp!=NULL)
			{
				cout<<tmp->data<<"|->";
				tmp=tmp->next;
			}
```

```
                        cout<<"NULL\n";
                }
        }
};

int main()
{
    Queue q;
    int n,ch,doch;
    cout<<"\n=====WELCOME========";
    cout<<"\n========= Queue Operations using Linkedlist =========";
    do
    {
        cout<<"\n===========";
        cout<<"\n1.Add Element into Queue(Enqueue)\n2.Delete Element from
Queue(Dequeue)\n3.Display elements in the Queue\n4.Exit";
        cout<<"\nEnter your choice: ";
        cin>>ch;
        switch(ch)
        {
            case 1:
                cout<<"Enter any number to insert in Queue: ";
                cin>>n;
                q.insert(n);
                break;

            case 2:
                q.dequeue();
                break;

            case 3:
                q.display();
                break;

            case 4:
                goto exit;
                break;
```

```
                default:
                        cout<<"\nPlease enter correct choice!";
                        break;
                }
                cout<<"\n=======\nDo you want to continue?[1 for YES || 0 for No]-->";
                cin>>doch;
        }while(doch==1);
        exit:
                cout<<"\n=====THANK YOU!=======";
        return 0;
}
```

==========

OUTPUT:

```
=====WELCOME========
========= Queue Operations using Linkedlist =========
===========
1.Add Element into Queue(Enqueue)
2.Delete Element from Queue(Dequeue)
3.Display elements in the Queue
4.Exit
Enter your choice:
```

```
=====WELCOME========
========= Queue Operations using Linkedlist =========
===========
1.Add Element into Queue(Enqueue)
2.Delete Element from Queue(Dequeue)
3.Display elements in the Queue
4.Exit
Enter your choice: 3

Queue is empty!Display Operation failed!
=======
Do you want to continue?[1 for YES || 0 for No]-->1

===========
1.Add Element into Queue(Enqueue)
2.Delete Element from Queue(Dequeue)
3.Display elements in the Queue
4.Exit
Enter your choice: 2

Queue is empty!Deletion Operation failed!
=======
Do you want to continue?[1 for YES || 0 for No]-->
```

```
==========
1.Add Element into Queue(Enqueue)
2.Delete Element from Queue(Dequeue)
3.Display elements in the Queue
4.Exit
Enter your choice: 1
Enter any number to insert in Queue: 10

=======
Do you want to continue?[1 for YES || 0 for No]-->1

==========
1.Add Element into Queue(Enqueue)
2.Delete Element from Queue(Dequeue)
3.Display elements in the Queue
4.Exit
Enter your choice: 1
Enter any number to insert in Queue: 20

=======
Do you want to continue?[1 for YES || 0 for No]-->1

==========
1.Add Element into Queue(Enqueue)
2.Delete Element from Queue(Dequeue)
3.Display elements in the Queue
4.Exit
Enter your choice: 1
Enter any number to insert in Queue: 30
```

```
=======
Do you want to continue?[1 for YES || 0 for No]-->1

==========
1.Add Element into Queue(Enqueue)
2.Delete Element from Queue(Dequeue)
3.Display elements in the Queue
4.Exit
Enter your choice: 1
Enter any number to insert in Queue: 40

=======
Do you want to continue?[1 for YES || 0 for No]-->1

==========
1.Add Element into Queue(Enqueue)
2.Delete Element from Queue(Dequeue)
3.Display elements in the Queue
4.Exit
Enter your choice: 1
Enter any number to insert in Queue: 50

=======
Do you want to continue?[1 for YES || 0 for No]-->
```

```
==========
1.Add Element into Queue(Enqueue)
2.Delete Element from Queue(Dequeue)
3.Display elements in the Queue
4.Exit
Enter your choice: 3

The elements in the Queue are:
10|->20|->30|->40|->50|->NULL

=======
Do you want to continue?[1 for YES || 0 for No]-->
```

```
==========
1.Add Element into Queue(Enqueue)
2.Delete Element from Queue(Dequeue)
3.Display elements in the Queue
4.Exit
Enter your choice: 2

An element 10 which was at FRONT has been Deleted!
=======
Do you want to continue?[1 for YES || 0 for No]-->
```

==========