

LINKED LIST IMPLEMENTATION OF STACK IN C ++

Write a class which will implement the Stack data structure in C++. Give the declaration & definition of the class. Also define the Node. For simplicity, you may assume stack to hold integer data type .

Solution :

There can be multiple type of implementations of Stack. Two most popular ones are

1: Array implementation

- ☐ Stack is stored in an Array .
- ☐ Easy to implement .
- ☐ Is of fixed size (i.e if the array is of size 100, then it will occupy 100 memory locations even if there are, say 2 elements in the stack, also we cannot store more than 100 elements in the Stack).

2 Linked List Implementation :

- ☐ Stack is stored in a Linked List .
- ☐ Have to manipulate pointers .
- ☐ The size of this implementation is dynamic and it takes memory proportional to the actual number of elements in Stack .

In both the implementations, the push and pop operations are constant-time operations, i.e $O(1)$

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node*link;
} *rear,*front,*p,*q;
void add()
{
    p=new node;
    cout<<"input new element"<<endl;
    cin>>p->data;
    p->link=NULL;
    if(rear==NULL)
```

```

front=p;
else rear->link=p;
rear=p;
}
void deleteq()
{
int item;
p=front;
if(p==NULL)
{
cout<<"error...the linked queue is empty"<<endl;
cout<<"press any key to exit"<<endl;
getch();
exit(0);
}
else
{
q=p->link;
item=p->data;
delete(p);
front=q;
if(front==NULL)
rear=NULL;
}
}
void main()
{
int choice,l,m,i;
front=NULL; rear=NULL;
clrscr();
do{
cout<<"representation of the linked queue operations"<<endl;
cout<<"-----"<<endl;
cout<<"1-add a new element    "<<endl;
cout<<"2-delete an element      "<<endl;
cout<<"3-display the content of the linked queue"<<endl;
cout<<"4-exit                    "<<endl;

```

```

cout<<"select your choice"<<endl;
cin>>choice;
switch (choice)
{
    case(1):
    {
        cout<<"how many elements you like to enter";
        cin>>m;
        for(i=0;i<m;i++)
            add( );
        break;
    }
    case(2):
    {
        cout<<endl<<"how many elements you want to delete";
        cin>>l;
        for(i=0;i<l;i++)
            deleteq();
        break;
    }
    case(3):
    {
        if(front==NULL)
            cout<<"error...the linked queue is empty"<<endl;
        else
        {
            q=front;
            cout<<"the content of the stack is:"<<endl;
            while(q!=NULL)
            {
                printf("\n%d\n",q->data);
                q=q->link;
            }
            break;
        }
    }
}

```

```
}while(choice!=4);}
```