|  |  |
| --- | --- |
| https://lh6.googleusercontent.com/NfxikauFutj0NGA6tZLo-9OYuuo3IKL8zEJJ1vcz211S75EzE4cV8OiG_9xiNDI7UXEpwJU3L0HucAlThGrGrN0dBZNqE1lpoGJHT3oGyRmOalINq0YRgVxjjMgLbZ3n8Pq0wpa0thnjQcjh_kKTHUlnU-ExSfeM05HoXY1-s1p-Qtg0OyqjFjrwB05Nj4mxriktv4A | SVKM’s NMIMS  School of Technology Management & Engineering Navi Mumbai Campus |
| Department of Computer Engineering |

|  |  |
| --- | --- |
| Name: Yash Patil | SAP ID: 70472200140 |
| Semester: III | Year: II |
| Subject: Data Structures and Algorithm | Roll No.: A176 |
| Practical: 6 | Date: 4/9/2023 |
| Batch: 1 |  |

**Aim:–**

Implementation of stack/queue using linked list.

**Theory:–**

We have used static array to implement given problem statement.

Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.

To create an array, define the data type (like int) and specify the name of the array followed by **square brackets []**.

To insert values to it, use a comma-separated list, inside curly braces:

We can loop through the array elements with the for loop.

The following example outputs all elements in the myNumbers array:

int myNumbers[] = {25, 50, 75, 100};  
int i;  
  
for (i = 0; i < 4; i++) {  
  printf("%d\n", myNumbers[i]);  
}

**Code/Implementation –**

1] STACK USING LINKED LIST:

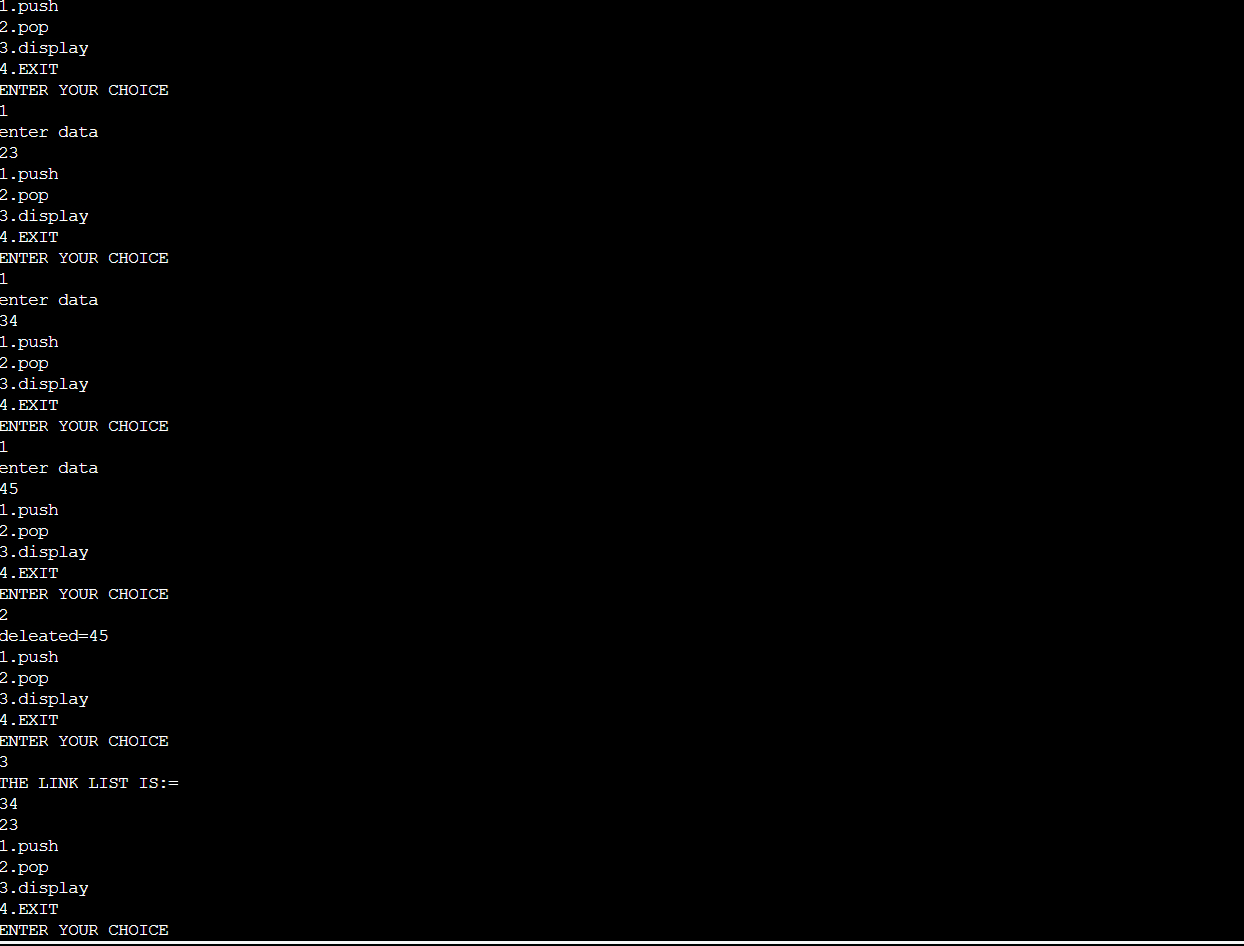
#include<iostream>  
#include<cstdlib>  
using namespace std;  
struct node  
{  
   int data;  
   struct node\*next;  
  
};  
struct node\*top=NULL;  
void Push(int val)  
{  
    struct node\*new1;  
    new1=(struct node\*)malloc(sizeof(struct node));//allocating memory for insertion  
    new1->data=val;  
    if(top==NULL)  
        top=new1;  
    else  
    {  
        new1->next=top;  
        top=new1;  
    }  
}  
void Pop()//delet the beggining  
{  
    struct node\*temp=top;  
    if(top==NULL)  
        cout<<"list empty"<<endl;  
    else  
    {  
        top=top->next;  
        cout<<"deleated="<<temp->data<<endl;  
        free(temp);  
    }  
}  
void display()  
{  
    cout<<"THE LINK LIST IS:="<<endl;  
    struct node\*temp;  
    temp=top;  
    while(temp->next!=NULL)  
    {  
        cout<<temp->data<<endl;  
        temp=temp->next;  
    }  
    cout<<temp->data<<endl;  
  
}  
int main()  
{  
    int x=4;  
    int ch,ele1;  
    do  
    {  
        cout<<"1.push"<<endl<<"2.pop"<<endl<<"3.display"<<endl<<"4.EXIT"<<endl<<"ENTER YOUR CHOICE"<<endl;  
        cin>>ch;  
       switch(ch)  
       {  
       case 1:  
        {  
            cout<<"enter data"<<endl;  
            cin>>ele1;  
            Push(ele1);  
            break;  
        }  
       case 2:  
        {  
            Pop();  
            break;  
        }  
       case 3:  
        {  
            display();  
            break;  
        }  
       case 4:  
        {  
            x=7;  
            break;  
        }  
       default:  
        {  
            cout<<"INVALID INPUT"<<endl;  
        }  
       }  
  
    }while(x<5);  
}

2] Queue using linked list:

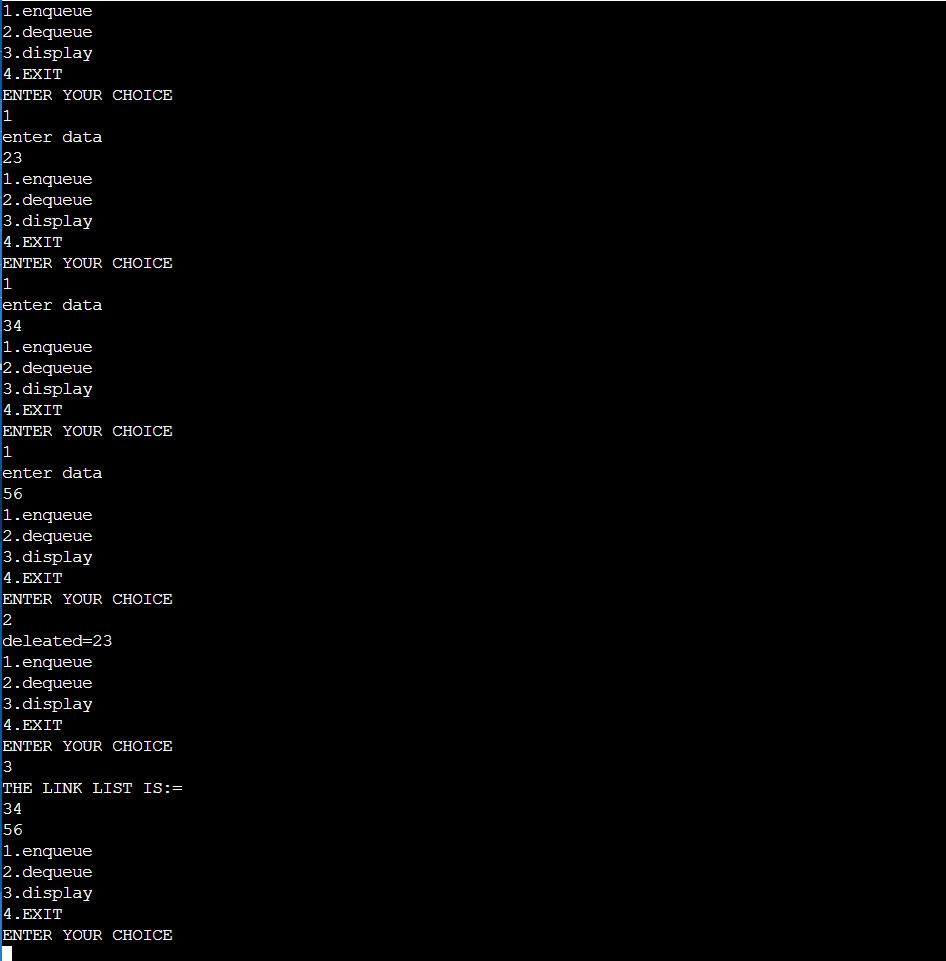
#include<iostream>  
#include<cstdlib>  
using namespace std;  
struct node  
{  
   int data;  
   struct node\*next;  
  
};  
struct node\*Front=NULL;  
struct node\*Rear=NULL;  
void enqueue(int val)  
{  
    struct node\*temp;  
    temp=Front;  
    struct node\*new1;  
    new1=(struct node\*)malloc(sizeof(struct node));//allocating memory for insertion  
    new1->data=val;  
    new1->next=NULL;  
    if (Front==NULL)  
    {  
        Front=new1;  
    }  
    else  
    {  
            while(temp->next!=NULL)//traverse till end  
        {  
            temp=temp->next; // traversing by incrementing  
  
        }  
        temp->next=new1;// new address given to temp  
  
  
    }  
  
  
}  
void dequeue()//delet the beggining  
{  
    struct node\*temp=Front;  
    if(Front==NULL)  
        cout<<"list empty"<<endl;  
    else  
    {  
  
        Front=Front->next;  
        cout<<"deleated="<<temp->data<<endl;  
        free(temp);  
    }  
  
}  
void display()  
{  
    cout<<"THE LINK LIST IS:="<<endl;  
    struct node\*temp;  
    temp=Front;  
    while(temp->next!=NULL)  
    {  
        cout<<temp->data<<endl;  
        temp=temp->next;  
    }  
    cout<<temp->data<<endl;  
  
}  
int main()  
{  
    int x=4;  
    int ch,ele1;  
    do  
    {  
        cout<<"1.enqueue"<<endl<<"2.dequeue"<<endl<<"3.display"<<endl<<"4.EXIT"<<endl<<"ENTER YOUR CHOICE"<<endl;  
        cin>>ch;  
       switch(ch)  
       {  
       case 1:  
        {  
            cout<<"enter data"<<endl;  
            cin>>ele1;  
            enqueue(ele1);  
            break;  
        }  
       case 2:  
        {  
            dequeue();  
            break;  
        }  
       case 3:  
        {  
            display();  
            break;  
        }  
       case 4:  
        {  
            x=7;  
            break;  
        }  
       default:  
        {  
            cout<<"INVALID INPUT"<<endl;  
        }  
       }  
  
    }while(x<5);  
}

**Output:-**

1] stack using linked list:



2] queue using linked list



**Conclusion:-**

I was successfully able to implememt stack and queue operations using linked list.

**Outcome: -** Identified and applied appropriate linear data structure for the given problem