**/\*WAP to implement STACK using linked list \*/**

**#include<iostream>**

**#include<cstdlib>**

**using namespace std;**

**struct node**

**{**

**int data;**

**node\* next;**

**};**

**class Stack**

**{**

**node \*head;**

**public:**

**Stack()**

**{**

**head = NULL;**

**}**

**void push()**

**{**

**int val;**

**cout<<endl<<endl<<"enter the value to push: ";**

**cin>>val;**

**node \*newNode = new node;**

**newNode->data = val;**

**if(head == NULL)**

**{**

**head = newNode;**

**head->next = NULL;**

**}**

**else**

**{**

**newNode->next = head;**

**head = newNode;**

**}**

**}**

**void pop()**

**{**

**if(head == NULL)**

**{**

**cout<<endl<<"No value to pop."<<endl;**

**}**

**else**

**{**

**node \*tmp = head;**

**head = head->next;**

**cout<<endl<<"the popped value: "<<tmp->data<<endl;**

**delete tmp;**

**}**

**}**

**void peek()**

**{**

**if(head == NULL)**

**{**

**cout<<endl<<"No value to peek"<<endl;**

**}**

**else**

**{**

**cout<<endl<<"The value at top: "<<head->data<<endl;**

**}**

**}**

**void display\_stack()**

**{**

**cout<<endl<<endl<<"Stack: "<<endl;**

**node \*ptr = head;**

**while(ptr != NULL)**

**{**

**cout<<" "<<ptr->data<<" ";**

**ptr = ptr->next;**

**}**

**cout<<endl<<endl<<endl;**

**}**

**};**

**int main()**

**{**

**Stack stackobj;**

**int choose;**

**do**

**{**

**cout<<"\n\n1. Push."<<endl;**

**cout<<"2. Pop."<<endl;**

**cout<<"3. Peek."<<endl;**

**cout<<"4.Exit"<<endl;**

**cout<<"\n\nChoose an option: ";**

**cin>>choose;**

**switch (choose)**

**{**

**case 1:**

**{**

**stackobj.push();**

**break;**

**}**

**case 2:**

**{**

**stackobj.pop();**

**break;**

**}**

**case 3:**

**{**

**stackobj.peek();**

**break;**

**}**

**case 4:**

**{**

**exit(1);**

**break;**

**}**

**default :**

**{**

**cout<<"Invalid input";**

**break;**

**}**

**}**

**stackobj.display\_stack();**

**}**

**while (choose!=4);**

**return 0;**

**}**

**/\*WAP to implement STACK using linked list \*/**

**#include<iostream>**

**#include<cstdlib>**

**using namespace std;**

**struct node**

**{**

**int info;**

**node \*next;**

**};**

**class Stack**

**{**

**node \*top;**

**bool IsEmpty()**

**{**

**if(top==NULL)**

**return true;**

**else**

**return false;**

**}**

**public:**

**Stack():top(NULL) {}**

**void push(int num)**

**{**

**node \*temp=new node;**

**if(temp==NULL)**

**cout<<"\n\nFailed to initialize the new node.\n\n";**

**else**

**{**

**temp->info=num;**

**if(top==NULL)**

**{**

**temp->next=NULL;**

**top=temp;**

**}**

**else**

**{**

**temp->next=top;**

**top=temp;**

**}**

**}**

**}**

**void pop()**

**{**

**if(IsEmpty())**

**cout<<"\n\nStack Underflow\n\n";**

**else**

**{**

**node \*temp;**

**temp=top;**

**cout<<"\n\nThe popped element of Stack is : "<<top->info<<endl<<endl;**

**top=top->next;**

**delete temp;**

**}**

**}**

**void peek()**

**{**

**if(IsEmpty())**

**cout<<"\n\nStack Underflow\n\n";**

**else**

**cout<<"\n\nThe top element of Stack is : "<<top->info<<endl<<endl;**

**}**

**void displayStack()**

**{**

**if(IsEmpty())**

**cout<<"\n\nStack Underflow\n\n";**

**else**

**{**

**node \*temp;**

**temp=top;**

**cout<<"\n\nElements of Stack are: \n";**

**while(temp!=NULL)**

**{**

**cout<<temp->info<<endl;**

**temp=temp->next;**

**}**

**cout<<"\n\n";**

**}**

**}**

**};**

**int main()**

**{**

**int choice,num;**

**Stack s;**

**while(1)**

**{**

**cout<<"1. push\n2. pop\n3. peek\n4. view stack\n5. Exit\n\nEnter your choice : ";**

**cin>>choice;**

**switch(choice)**

**{**

**case 1:**

**{**

**while(1)**

**{**

**cout<<"\n\nEnter -1 to end push operation\nEnter the value : ";**

**cin>>num;**

**if(num==-1)**

**break;**

**s.push(num);**

**}**

**break;**

**}**

**case 2:**

**{**

**s.pop();**

**break;**

**}**

**case 3:**

**{**

**s.peek();**

**break;**

**}**

**case 4:**

**{**

**s.displayStack();**

**break;**

**}**

**default :**

**exit(0);**

**}**

**}**

**return 0;**

**}**

**/\*WAP to implement STACK using linked list \*/**

**#include<iostream>**

**using namespace std;**

**class Stack**

**{**

**struct node**

**{**

**int data;**

**struct node \* next;**

**};**

**public:**

**struct node \* start;**

**struct node \* newnode,\* temp,\* ptr,\* preptr;**

**void creation()**

**{**

**newnode = new node;**

**cout<<"Enter the data for the stack(insert -1 to end the ): ";**

**cin>>newnode->data;**

**newnode->next=NULL;**

**if (start==NULL)**

**{**

**start=newnode;**

**temp=newnode;**

**}**

**else**

**{**

**temp->next=newnode;**

**temp=newnode;**

**}**

**do**

**{**

**push();**

**}**

**while (newnode->data!=-1);**

**}**

**void push()**

**{**

**newnode=new node;**

**cout<<"Enter the data to be stored at the top: ";**

**cin>>newnode->data;**

**if (newnode->data!=-1)**

**{**

**ptr=start;**

**while(ptr->next!=NULL)**

**{**

**ptr=ptr->next;**

**}**

**ptr->next=newnode;**

**newnode->next=NULL;**

**}**

**}**

**void pop()**

**{**

**ptr=start;**

**while(ptr->next!=NULL)**

**{**

**preptr=ptr;**

**ptr=ptr->next;**

**}**

**cout<<"The deleted value is: "<<ptr->data;**

**delete ptr;**

**preptr->next=NULL;**

**}**

**void peek()**

**{**

**ptr=start;**

**while(ptr->next!=NULL)**

**{**

**ptr=ptr->next;**

**}**

**cout<<"Value at the top = "<<ptr->data;**

**}**

**void display\_stack()**

**{**

**ptr=start;**

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

**cout<<"\n\nThe stack is: "<<endl;**

**cout<<"\t\t\t"<<ptr->data<<endl;**

**while(ptr->next!=NULL)**

**{**

**ptr=ptr->next;**

**cout<<"\t\t\t"<<ptr->data<<endl;**

**}**

**cout<<endl;**

**cout<<"Note: top is at the bottom"<<endl;**

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

**}**

**};**

**int main()**

**{**

**class Stack st;**

**st.start=NULL;**

**int choice=0,c=0;**

**while(choice!=10)**

**{**

**c++;**

**cout<<"\n\nyour Choice please: "<<endl;**

**if (c==1)**

**{**

**cout<<"0-Creating a new Stack "<<endl;**

**}**

**cout<<"1-Push "<<endl;**

**cout<<"2-Pop "<<endl;**

**cout<<"3-Peek "<<endl;**

**cout<<"10-Exit.\n"<<endl;**

**cout<<"\t\tyour choice: ";**

**cin>>choice;**

**switch (choice)**

**{**

**case 0:**

**st.creation();**

**break;**

**case 1:**

**st.push();**

**break;**

**case 2:**

**st.pop();**

**break;**

**case 3:**

**st.peek();**

**break;**

**}**

**st.display\_stack();**

**}**

**cout<<"THANK YOU";**

**}**

**/\*WAP to implement STACK using linked list \*/**

**#include<iostream>**

**using namespace std;**

**class linkList**

**{**

**struct Node**

**{**

**int data;**

**Node \*next;**

**};**

**typedef struct Node\* nodeptr;**

**nodeptr head;**

**public:**

**linkList() //constructor**

**{**

**head=NULL;**

**}**

**void push(int new\_data) // insert at the top**

**{**

**nodeptr p;**

**p=new Node;**

**p->data= new\_data;**

**p->next=head;**

**head=p;**

**}**

**int pop() // delete from the top**

**{**

**nodeptr ptr=head;**

**if(head!=NULL)**

**{**

**head=ptr->next;**

**cout<<ptr->data<<" is popped\n\n"<<endl;**

**delete ptr;**

**return ptr->data;**

**}**

**else**

**{**

**cout<<"Empty\n\n"<<endl;**

**return -1;**

**}**

**}**

**int peek() // delete from the front**

**{**

**nodeptr ptr=head;**

**if(head!=NULL)**

**{**

**cout<<ptr->data<<" is in top\n\n"<<endl;**

**return ptr->data;**

**}**

**else**

**{**

**cout<<"Empty\n\n"<<endl;**

**return -1;**

**}**

**}**

**void display() // display the list**

**{**

**nodeptr p=head;**

**cout<<"\n\t=================X================"<<endl;**

**cout<<"\taddress"<<"\t\tdata"<<"\tnext"<<endl;**

**while(p!=NULL)**

**{**

**cout<<"\t"<<p<<"\t"<<p->data<<"\t"<<p->next<<endl;**

**p=p->next;**

**}**

**if(head==NULL)**

**{**

**cout<<"\tEmpty"<<endl;**

**}**

**cout<<"\tthats it"<<endl;**

**cout<<"\t=================X================\n"<<endl;**

**}**

**};**

**int main()**

**{**

**linkList li;**

**int x,a;**

**int choice=-1;**

**while(choice!=0)**

**{**

**cout<<"\n\nyour Choice please: "<<endl;**

**cout<<"1-push "<<endl;**

**cout<<"2-pop "<<endl;**

**cout<<"3-peek "<<endl;**

**cout<<"0-Exit\n"<<endl;**

**cout<<"\t\tyour choice: ";**

**cin>>choice;**

**system("CLS");**

**cout<<"\tBEFORE LIST";**

**li.display();**

**switch (choice)**

**{**

**case 1:**

**cout<<"enter data to push: ";**

**cin>>x;**

**li.push(x);**

**break;**

**case 2:**

**li.pop();**

**break;**

**case 3:**

**li.peek();**

**break;**

**}**

**cout<<"\tAFTER LIST";**

**li.display();**

**}**

**cout<<"\n============X==========="<<endl;**

**cout<<"\t THANK YOU "<<endl;**

**return 0;**

**}**