*/\**

*\* C++ Program To Implement Stack using Linked List*

*\*/*

#include<iostream>

#include<cstdlib>

using namespace std;

*/\**

*\* Node Declaration*

*\*/*

struct node

{

int info;

struct node \*link;

}\*top;

*/\**

*\* Class Declaration*

*\*/*

class stack\_list

{

public:

node \*push(node \*, int);

node \*pop(node \*);

void traverse(node \*);

stack\_list()

{

top = NULL;

}

};

*/\**

*\* Main: Contains Menu*

*\*/*

int main()

{

int choice, item;

stack\_list sl;

while (1)

{

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

cout<<"Operations on Stack"<<endl;

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

cout<<"1.Push Element into the Stack"<<endl;

cout<<"2.Pop Element from the Stack"<<endl;

cout<<"3.Traverse the Stack"<<endl;

cout<<"4.Quit"<<endl;

cout<<"Enter your Choice: ";

cin>>choice;

switch(choice)

{

case 1:

cout<<"Enter value to be pushed into the stack: ";

cin>>item;

top = sl.push(top, item);

break;

case 2:

top = sl.pop(top);

break;

case 3:

sl.traverse(top);

break;

case 4:

exit(1);

break;

default:

cout<<"Wrong Choice"<<endl;

}

}

return 0;

}

*/\**

*\* Push Element into the Stack*

*\*/*

node \*stack\_list::push(node \*top, int item)

{

node \*tmp;

tmp = new (struct node);

tmp->info = item;

tmp->link = top;

top = tmp;

return top;

}

*/\**

*\* Pop Element from the Stack*

*\*/*

node \*stack\_list::pop(node \*top)

{

node \*tmp;

if (top == NULL)

cout<<"Stack is Empty"<<endl;

else

{

tmp = top;

cout<<"Element Popped: "<<tmp->info<<endl;

top = top->link;

delete(tmp);

}

return top;

}

*/\**

*\* Traversing the Stack*

*\*/*

void stack\_list::traverse(node \*top)

{

node \*ptr;

ptr = top;

if (top == NULL)

cout<<"Stack is empty"<<endl;

else

{

cout<<"Stack elements :"<<endl;

while (ptr != NULL)

{

cout<<ptr->info<<endl;

ptr = ptr->link;

}

}

}