11/30/2021

Department Of Computer Science

Subject: Data Structure and Algorithm **Instructor:** Ma’am Zainab Malik

Lab No: 6 **Date:** 11-30-2021

Class: BSCS-3B

Students’ Name

1)	Madina Javed Iqbal Khokhar	2426
2)	Esha Mansoor	2413
3)	Sultan Zahid	2411
4)	Abdul Moeed	2419

Lab Repot 6

Task:

Implement a program that will illustrate usage of stack using linked list.

Description:

In this lab we had used linked list to implement stack . Firstly, we had created a node class for this purpose ,however , we can also import our Node class encapsulating same properties as required from previous project by specifying our path. Then, we had created a LinkedStack class in which we have declared a variable top of data type node. Then, our next step is making a constructor for top. We had created some functions in this class that are given below:

- Push
- Pop
- Is empty
- Top Value
- RemoveAnd Display

In the push function ,if top is at the zero it means our list is empty ,so, we will set top equal to ptr however if one or more than one element exist , we will set ptr to the next of the top and then top will be be equal to ptr. In the push function , if else condition is used , if top is equal to zero, it means list is empty and there is nothing to be deleted ,however , if there is only one element, the, we will store its info in any variable , then delete info at the top and return its info. If there is more than one element, then, we will store top in a node named temp and set top to its next. Then, store the info of temp in a variable temp and lastly we will delete and return the info. RemoveAndDisplay function will pop all nodes by calling pop function until our top becomes zero.

In the main function, our first step is adding our linkedlist stack in the main function. Then , we will push a word NUML char by char and call removeaAndDisplay function to remove and display each char.

We have created a numberConversion function for converting a decimal number to binary in which we use while condition that if our entered number is greater 1 then divide the number by 2 and push their remainder and store its quotient in num variable. Lastly, we will call removeAndDisplay function.Our next function is symbol balancing in which we had created an empty stack of char and read input text char by char till the end where we had use if and else if condition to define particular case that are given below

1. If char is an opening symbol, we will push it on the stack.
2. If char is a closing symbol,we will have two cases:
   - If stack is empty, report an error ("Opening symbol missing")
   - If stack is non-empty, pop a char from stack and match it with input char. If both characters do not match then report an error ("Symbol mismatch")
3. At the end of input, if stack is non-empty then report an error ("Closing Symbol missing")

Then, we had created a menu for the user and use switch statement to perform the function according to number entered by the user.

Code:

Node Class:

```
#include<iostream>

using namespace std;

template<class T> // Templates actually increase flexibility, they're easy

// to update, and they provide consistency across the project

class Node

{

        private:

                T info;  //  variable name use to store information

                Node<T> *next;  // variable use to store address  of next node,that's
why its data type

                                                        // is node and "T"  is used for Template



        public:
```

```
        Node(T i=0,Node<T> *n=0):info(i),next(n)  // constructor
        {    // constructor { having same name as class}
        }

        void setInfo(T i);  // using setter and getter
        T getInfo();
        void setNext(Node<T> *n); // calling setter and getter
        Node<T>* getNext();
};//EOC

template<class T>
void Node<T>::setInfo(T i)
{
    info=i;    // setting our info
}

template<class T>
void Node<T>::setNext(Node<T> *n)
{
    next=n; // setting our next
}

template<class T>
```

```
T Node<T>::getInfo()   // getting our info
{
        return info;
}


template<class T>
Node<T>* Node<T>::getNext()  // getting our next
{
        return next;
}
```

LinkedStack Class:

```
#include<iostream>
#include "Node.h"
using namespace std;
template<class T>
class LinkedStack
{
    private:
         Node<T> *top; //   top is declared which is the fudamental characteristics of linked stack
         public:
         LinkedStack()  // constructor
         {
             top=0; // setting top to zero
         }
         // declaring functions
         void push(T element); //add to head
         T pop(); // remove from head
         bool isEmpty(); //  is empty
         T topValue(); // to access top value
         void removeAndDisplayAll(); //to remove  and  diplay
};

template<class T>
void LinkedStack<T>::push(T element)   // to push  element
{

         Node<T> *ptr=new Node<T>(element); // creating a node
         //info=5 & next=0

         if(top==0)//list is empty
         {
             top=ptr;

         }
         else //only one element or >1 element
         {
             ptr->setNext(top); // Increment top     //next of 9 is 23
             top=ptr; // Insert value at top index

             //head will now poT Node<T> with value 9
         }

}//push
template<class T>
T LinkedStack<T>::pop() // to pop en elemnet
{
         if(top==0)   // is stack is empty
         {
             cerr<<"nothing to delete"<<endl;
         }
         else if(top->getNext()==0)   // if only one element exist
         {
             T info=top->getInfo(); //storing value of  top info in info variable
             delete top; // deleting value at tho top
             top=0;

             return info; // returning value of info( deleted value)
         }
         else//more than one element
```

```
        {
                Node<T> *temp=top;  // storing  top in temp
                top=top->getNext(); // increment top
                T info=temp->getInfo();
                delete temp;
                return info;
        }
}//pop
template<class T>
bool LinkedStack<T>::isEmpty()
{
        return top==0;
}//isEmpty

template<class T>
T LinkedStack<T>::topValue()
{
        return top->getInfo(); // getting value  at the top and then return
} //topValue

template<class T>
void LinkedStack<T>::removeAndDisplayAll()
{
    while(top!=0)  // until stack become  empty
    {
        cout<<pop()<<" "; // calling pop function
        //cout<<endl;
    }
}//removeAndDisplayAll
```

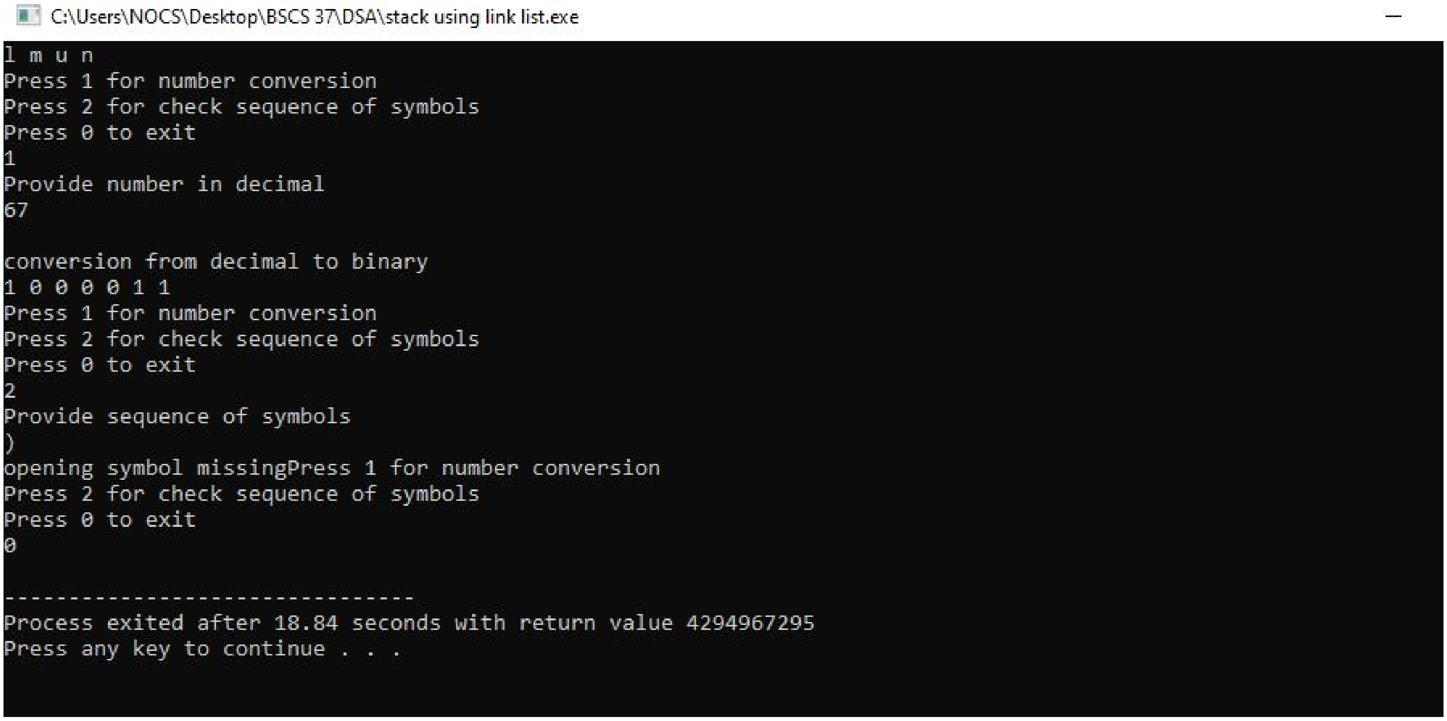
Main Function:

```
#include <iostream>
#include "LinkedStack.h" // including LinkedStack
/* run this program using the console pauser or add your own getch,
system("pause") or input loop */
void numberConversion(int num); // declaring functions
void symbolBalancing(string text);
int main(int argc, char** argv) {

    LinkedStack<char>list1;
    list1.push('n');
    list1.push('u');
    list1.push('m');
    list1.push('l');
    list1.removeAndDisplayAll(); // calling function
    cout<<endl;
    int opt;
```

```
do
{  // menu

    cout<<"Press 1 for number conversion "<<endl;
    cout<<"Press 2 for check sequence of symbols "<<endl;
    cout<<"Press 0 to exit "<<endl;
    cin>>opt;
    string text;
    switch (opt) // using switch
    {
        case 1:
            // call function for number conversion
            int num;
            cout<<"Provide number in decimal "<<endl;
            cin>>num; // getting a number for conversion
            cout<<endl;
            numberConversion(num); // calling  a number conversion function
            break;
        case 2:

            cout<<"Provide sequence of symbols"<<endl;
            cin>>text;  // getting a string of symbols
            symbolBalancing(text); // calling a symbol balancing funnction
            break;
        case 0:
          exit(-1);
    }
}while(true);

    return 0;
}

void numberConversion(int num) // number conversion function
{
    LinkedStack<int> stack;
    while(num>=1) //it should be greater than 0  as 0 is  binary
    {
        stack.push(num%2);// for  remiander
        num=num/2; // for quotient
    }
    cout<<"conversion from decimal to binary";
    cout<<endl;
    stack.removeAndDisplayAll(); // calling  remove  and display function
    cout<<endl;
}// numbertConversion
```

```
void symbolBalancing(string text) // symbolBalancing function
{
    LinkedStack<char> stack;
    for(int i=0;i<text.length();i++)
    {
        if(text[i]=='('||text[i]=='['||text[i]=='{'||text[i]=='<')
        {  // If char is an opening symbol, push it on the stack

            stack.push(text[i]);
        }//2.1

        // If char is a closing symbol, then we will two situations
        else if(text[i]==')'||text[i]==']'||text[i]=='}'||text[i]=='>')
        {
            // If stack is empty, report an error ("Opening symbol missing")
            if(stack.isEmpty())//2.2.1
            {
                cout<<"opening symbol missing";
            }
            else//2.2.2
            {
                //If stack is non-empty, pop a char from stack and match it with input char.
                char stackVal=stack.pop();

                if((text[i]==']'&& stackVal!='[')||(text[i]=='}'&&
stackVal!='{')||(text[i]==')'&& stackVal!='(')||(text[i]=='>'&& stackVal!='<'))
                {
                    // If both characters do not match then report an error
(“Symbol mismatch”)
                    cout<<"Symbol mismatch"<<endl;
                    return;
                }
            }
        }
    }

    if(!stack.isEmpty()) // is stack is not empty
    {
        cout<<"valid sequence"<<endl;
    }
}
```

Output:



```
l m u n
Press 1 for number conversion
Press 2 for check sequence of symbols
Press 0 to exit
1
Provide number in decimal
67

conversion from decimal to binary
1 0 0 0 0 1 1
Press 1 for number conversion
Press 2 for check sequence of symbols
Press 0 to exit
2
Provide sequence of symbols
)
opening symbol missingPress 1 for number conversion
Press 2 for check sequence of symbols
Press 0 to exit
0

--------------------------------
Process exited after 18.84 seconds with return value 4294967295
Press any key to continue . . .
```

Thanks