ព្រះរាជាណាចក្រកម្ពុជា

ជាតិ សាសនា ព្រះមហាក្សត្រ

Institute of technology of Cambodia

Department of Information and communication Engineering

The lesson taking about queue and stack in c++

TP17: Queue and Stack

TP: Algorithm and Programming ll

Lecturer: BOU CHANNA

Student: VEN THON

ID: e20191250

Group: I3-GIC-C

Contents

Poblem1:

Create an implementation of queue and stack as linked list. Put the implementation of as a header file (MyStack-Yourname.h) and the implement of queue as another header file (MyQueue-Yourname.h). you can reuse the Single linked list implementation then rename function and variable needed by to the terms(push, pop, top, rear, front, dequeue) used by Stack and Queue.

tart here | MyQueue_Ven_Thon.h

```
1  struct Element{
2      int data;
3      Element*next;
4  };
5  struct Queue{
6      Element *myrear, *myfront;
7      int n;
8  };
9  Queue*createEmptyQueue()
10 {
11     Queue*q;
12     q=new Queue;
13     q->myrear=NULL;
14     q->myfront=NULL;
15     q->n=0;
16     return q;
17 }
18 void enqueue(Queue*q, int newData)
19 {
20     Element*e;
21     e=new Element;
22     e->data=newData;
23     e->next=NULL;
24     if(q->n==0)
25     {
26         q->myfront=e;
27         q->myrear=e;
28     }else{
29         q->myrear->next=e;
30         q->myrear=e;
31     }
32     q->n=q->n+1;
33 }
```

```
34 void dequeue(Queue*q)
35 {
36     if(q->n==0)
37     {
38         cout<<"can not delete Queue is empty"<<endl;
39     }else if(q->n==1){
40         Element*e;
41         e=q->myfront;
42         q->myfront=NULL;
43         q->myrear=NULL;
44         delete e;
45         q->n=q->n+1;
46     }else{
47         Element*e;
48         e=q->myfront;
49         q->myfront=q->myfront->next;
50         delete e;
51         q->n=q->n-1;
52     }
53 }
54 void displayMyQueue(Queue*q)
55 {
56     Element*tmp;
57     tmp=q->myfront;
58     while(tmp!=NULL)
59     {
60         cout<<tmp->data<<" ";
61         tmp=tmp->next;
62     }
63 }
64
```

tart here | MyStack_Ven_Thon.h

```
struct Element{
    int data;
    Element *next;
};

struct Stack{
    Element *top;
    int n;
};

Stack *createEmptyStack(){
    Stack *st;
    st = new Stack;
    st->top = NULL;
    st->n = 0;
    return st;
}

void push(Stack *st,int newData){
    Element *e;
    e = new Element;
    e->data = newData;

    e->next = NULL;
    if(st->n==0){
        st->top = e;
        e->next=NULL;
    }else{
        e->next = st->top;
        st->top = e;
    }
    st->n = st->n+1;
}
```

```
void pop(Stack *st){
    Element *t;
    if(st->n==0){
        cout<<"Stack is underflow! Stack is empty! Can not delete\n";
    }else{
        t = st->top;
        st->top = st->top->next;
        delete t;
        st->n = st->n-1;
    }
    cout<<endl;


}
void displayMyStack(Stack *st){
    Element *t;
    t=st->top;
    while(t!=NULL){
        cout<<t->data<<" ";
        t = t -> next;
    }
    cout<<endl;
}
```

Problem2:

Process each character in following text and store in an empty queue (FIFO method). When a character is a letter, add it to queue. When a character is an asterisk, remove data from queue. Make sure to display the character from the queue before it is removed.

E A S * Y * Q U E * * * S T * * * I O * N * * *

tart here | Ex2 VENTHON,e20191250.cpp

```
#include<iostream>
using namespace std;
#include"MyQueue_Ven_Thon.h"

void removeCharacter(string s)
{
    Element*e;
    e=new Element;
    e->next=NULL;
    int j=0;
    for(int i=0; i<s.size(); i++)
    {
        if((s[i]>='A' && s[i]<='Z') || (s[i]>='a' && s[i]<='z'))
        {
            s[j]=s[i];
            j++;
        }
    }
    cout<<s.substr(0, j);
}
main()
{
    Queue*q1;
    q1=createEmptyQueue();
    string s="E A S * Y Q U E * * * S T * * * I O * N * * *";
    cout<<endl<<"\t";
    removeCharacter(s);
    displayMyQueue(q1);
}
```

```
"C:\Users\Admin\Desktop\code c++\Queue&stack\Ex2 VENTHON,e20191250.exe"

        EASYQUESTION
Process returned 0 (0x0)   execution time : 0.041 s
Press any key to continue.
```

Problem3:

Write a program to ask a piece of text from a user. Display the text in a reverse order using stack.Hint: Push each character in the text into stack. Display stack.

tart here | *Ex3 VENTHON,e20191250.cpp

```
#include<iostream>
#include<string.h>
#include<stack>
using namespace std;
#include"MyStack_Ven_Thon.h"

void reverse(char *p)
{
    stack<char>s;
    for(int i=0; i<strlen(p); i++)
    {
        s.push(p[i]);
    }
    for(int i=0; i<strlen(p); i++)
    {
        p[i]=s.top();
        s.pop();
    }
}
main()
{
    Stack*s1;
    s1=createEmptyStack();
    char string[30];
   cout<<"\n Enter of text: ";
   cin>>string;
   reverse(string);
   cout<<"\n results after reverse: "<<string;
}
```

C:\Users\Admin\Desktop\code c++\Queue&stack\Ex3 VENTHON,e20191250.exe

```
 Enter of text: boy.loy.hello

 results after reverse: olleh.yol.yob
Process returned 0 (0x0)   execution time : 12.029 s
Press any key to continue.
```

Problem4:

Write a program to get a number from a user. Use stack to storethat number asthe binary numberrepresentation in the stack. Finally, display the binary representation from the stack. Hint: divide the integer by 2 then store remainder in stack.

rt here | Ex4 VENTHON,e20191250.cpp

```cpp
#include<iostream>
using namespace std;
#include"MyStack_Ven_Thon.h"

main()
{
    Stack*s1;
    s1=createEmptyStack();
    int num, binaryNum=0, rem, mul=1;
    cout<<"\nEnter the number: ";cin>>num;
    while(num>0)
    {
        rem=num%2;
        binaryNum=binaryNum+(rem*mul);
        mul=mul*10;
        num=num/2;
    }
    cout<<"\nValue of binary number is: "<<binaryNum;
    cout<<endl;

}
```

```
"C:\Users\Admin\Desktop\code c++\Queue&stack\Ex4 VENTHON,e20191250.exe"

Enter the number: 105

Value of binary number is: 1101001

Process returned 0 (0x0)   execution time : 6.479 s
Press any key to continue.
```

Problem5:

Store 20random numbers(1 to 1000)in a queue. Create a function voiddequeueKtime(int k) that display and deletek elements from the queue, or until the queueis empty.

art here × Ex5 VENTHON,e20191250.cpp ×

```cpp
#include<iostream>
#include<time.h>
#include<queue>
#include<stdlib.h>
using namespace std;
struct Element{
    int data;
    Element*next;
};
struct Queue{
    Element *myrear, *myfront;
    int n;
};
Queue*createEmptyQueue()
{
    Queue*q;
    q=new Queue;
    q->myrear=NULL;
    q->myfront=NULL;
    q->n=0;
    return q;
}
```

```cpp
void enqueue(Queue*q, int newData)
{
    Element*e;
    e=new Element;
    e->data=newData;
    e->next=NULL;
    if(q->n==0)
    {
        q->myfront=e;
        q->myrear=e;

    }else{
        q->myrear->next=e;
        q->myrear=e;
    }
    q->n=q->n+1;
}
void dequeue(Queue*q)
{
    /*if(q->n==0)
    {
        cout<<"can not delete Queue is empty"<<endl;
    }else*/ if(q->n==1){
        Element*e;
        e=q->myfront;
        q->myfront=NULL;
        q->myrear=NULL;
        delete e;
        q->n=q->n+1;
    }else{
        Element*e;
        e=q->myfront;
        q->myfront=q->myfront->next;
        delete e;
        q->n=q->n-1;
    }
}
```

```
60  void displayMyQueue(Queue*q)
61  {
62      Element*tmp;
63      tmp=q->myfront;
64      while(tmp!=NULL)
65      {
66          //cout<<tmp->data<<" ";
67          tmp=tmp->next;
68      }
69  }
70  queue<int>q;
71  void dequeuektime(int k)
72  {
73      for(int i=0; i<k&&!q.empty(); i++)
74      {
75          cout<<q.front()<<"\n";
76          q.pop();
77      }
78  }
79  main()
80  {
81      Queue*ql;
82      ql=createEmptyQueue();
83      srand(time(0));
84      for(int i=0; i<=20; i++)
85      {
86          q.push(rand()%1000);
87      }
88      dequeuektime(20);
89  }
90
```

```
"C:\Users\Admin\Desktop\code c++\Queue&stack\Ex5 VENTHON,e20191250.exe"
335
454
987
326
826
636
300
951
459
760
322
61
622
216
340
421
979
647
987
372

Process returned 0 (0x0)   execution time : 0.061 s
Press any key to continue.
```