

Lab 6 Stack Application&Queue

เป็นการบ้านที่เขียนสนุกๆ

- 1) Balancing Symbol : ใช้โครงสร้างข้อมูลแบบ stack ทำการตรวจสอบว่าเครื่องหมายที่ให้มาครบถูกต้องหรือไม่ โดยกำหนด error 3 แบบคือ เครื่องหมายไม่ตรงกัน เช่น {) , ไม่มีเครื่องหมายเปิด) , และ เครื่องหมายเกิน { { } เครื่องหมาย ประกอบด้วย { } () []

Input : {[.

Error! Inconsistent symbol //แสดงข้อความผิดพลาด ไม่ตรงกัน

หรือ

Input : {(.)}.

Error! no open symbol //แสดงข้อความผิดพลาด ไม่มีเครื่องหมายเปิด

หรือ

Input : {{(.[.]}

Error! Excess symbols //แสดงข้อความผิดพลาด เครื่องหมายเกิน

หมายเหตุ การ input จะรับ character ติดกัน ไม่มีช่องว่าง สิ้นสุดด้วยเครื่องหมาย .

ตัวอย่าง

```
#include <iostream>
using namespace std;

int main()
{ char c;

  while(c!='.')
  {   c=getchar();
      if(c=='(')
          cout << "open" << endl; //นิสิตไป push หรือ pop เอง
      else if(c==')')
          cout << "close"<< endl;

  }
}
```

- 2) จงแปลงนิพจน์ infix เป็น postfix โดยใช้ stack กำหนดเครื่องหมายประกอบด้วย + - * / () ถ้าแปลงนิพจน์ได้ 2 ตัวอย่างแรก ได้คะแนนครึ่งหนึ่ง ถ้าทำตัวอย่างสุดท้ายได้ ได้คะแนนเต็ม

Input : 10 + 2 .

Output : 10 2 +

หรือ

Input : 10 * 2 + 1 - 15 .

Output : 10 2 * 1 + 15 -

หรือ

Input : 10 + 15 * 3 + (7 * 40 + 8) * 20 .

Output : 10 15 3 * + 7 40 * 8 + 20 * +

- 3) Postfix expression : ใช้โครงสร้างข้อมูลแบบ stack ทำ postfix expression

- กำหนด input เป็นข้อมูลประกอบด้วยตัวเลข และ operator * + - (3 ชนิดเท่านั้น)
สิ้นสุดการ input ด้วยเครื่องหมาย .

Input : 15 3 * . // เว้นแต่ละพจน์ด้วยช่องว่าง 1 ช่อง

Output : 45

หรือ

Input : 15 3 - .

Output : 12

หรือ

Input : 12 31 2 * + 5 - .

Output : 69

ตัวช่วย : โปรแกรมแยกแยะระหว่างข้อมูลที่ได้รับเข้ามาว่าตัวใดเป็นเครื่องหมายตัวใดเป็น string และ

เปลี่ยน string ให้เป็นตัวเลขเพื่อให้คำนวณได้

```
#include <iostream>
#include <string>
#include <sstream> // this will allow you to use stringstream
```

```

using namespace std;

int main()
{
    stringstream ss; //type stringstream เป็นตัวกลางระหว่าง string กับ ตัวเลข
    string str="";
    int num;
    while(str!=".")
    {
        cin >> str;
        if(str==".")
            break;

        if(str=="+")
            cout << "ADD"<< endl; //push pop
        else
        {
            ss << str; //นำ string ใส่ในตัวกลาง
            ss >> num; //นำข้อมูลจากตัวกลางไปใส่ตัวแปรที่เป็นตัวเลข
            ss.clear(); //clear ค่า
            cout << num << endl;
        }
    }
}
ทดสอบ
15 3 + .

```

4) จงเขียนโปรแกรม Queue โดยใช้ Linked list เป็นโครงสร้าง ทำงานตาม menu ดังนี้

=====

MENU

=====

1. Enqueue

2. Dequeue

3. Exit

Please choose >

ให้ทำเหมือน lab ที่แล้ว คือ insert เสร็จ 1 ค่าแล้วให้กลับมาที่ menu เลย

กรณีเลือกข้อ 1

Enqueue : 10

Q = 10 // แสดงข้อมูลทั้งหมดในคิว

ถ้าเลือกข้อ 1 อีก

Enqueue : 12

Q = 10 12

ให้ทดสอบใส่ข้อมูลให้คิวมีข้อมูล 10 12 15 2 16 18 โดยใส่เรียงลำดับ สังเกตว่าข้อมูลในคิวเรียงลำดับตามเวลาเข้าคิว ไม่ได้เรียงตามค่า

กรณีเลือกข้อ 2

Deque 10 Success! //ดึงข้อมูลหัวคิวออกมาแสดง

Q = 12 15 2 16 18

ให้ทดสอบโดยดึงข้อมูลจนคิวว่าง

ถ้าคิวว่างแล้ว Dequeue จะแสดงข้อความ

Queue Empty Can't Dequeue!!!

ให้ทดสอบโดย insert ข้อมูลตามลำดับดังนี้ 6 5 8 10 2 15

5) Josephus problem

- N people, numbered 1 to N, are sitting in a circle.
- Starting at person 1, a hot potato is passed.
- After m passed, the person holding the hot potato is eliminated, the circle closes ranks,
- The game continues with the person who was sitting after the eliminated person picking up the hot potato.
- The last remaining person wins.
- Thus, if $M = 0$ and $N = 5$, players are eliminated in order, and player 5 wins.
- If $M=1$ and $N = 5$, the order of elimination is 2,4,1,5.

จงเขียนโปรแกรมแก้ปัญหา josephus problem ให้ทำการรับ input ค่า n คือจำนวนผู้เล่น และ m คือจำนวน pass แสดงผลลัพธ์เป็นหมายเลขผู้ชนะ เช่น

Input N : 5

Input M : 1

Winner = 3

หรือ

Input N : 20

Input M : 200

Winner = ?