

## Lab4 Solutions

Note: ในเฉลยของแลปนี้จะใช้ Stack ของ Standard Template Library(STL) แทนการ implement stack เองโดยใช้ linked list จะถือว่าเข้าใจหลักการเขียน stack มาบ้างแล้วนิดนึง อีกอย่างโค้ดตัวอย่างจะได้ไม่ยาวเกิน มีคำสั่งที่ใช้บ่อยประมาณนี้

- |                             |                             |
|-----------------------------|-----------------------------|
| - stack<variable type> name | ประกาศ stack                |
| - name.push(variable)       | เพิ่มข้อมูล                 |
| - name.top()                | ดูข้อมูลที่อยู่ด้านบนสุด    |
| - name.pop()                | ลบข้อมูลที่อยู่ด้านบนสุดออก |

ศึกษาเพิ่มเติมได้ที่ <https://cplusplus.com/reference/stack/>

### Dynamic Stack

Implement stack เพื่อรับคำสั่งทั้งหมด 4 คำสั่ง ระวังเรื่องการ pop stack ที่ว่างและการเข้าถึงข้อมูลที่อยู่ด้านบนสุดตอนที่ stack ว่างอาจทำให้ error ได้

### Example Code

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    stack<int> stack;
    int n;
    cin >> n;
    while(n-->0)
    {
        int command;
        cin >> command;
        if(command == 1)
        {
            int a;
            cin >> a;
            stack.push(a);
        }
        else if(command == 2 && !stack.empty()) stack.pop();
    }
}
```

```
    else if(command == 3)
    {
        if(stack.empty()) cout << "EMPTY" << endl;
        else cout << stack.top() << endl;
    }
    else if(command == 4) cout << stack.size() << endl;
}
}
```

*time complexity :  $O(n)$*

## Parenthesis

ใช้ stack เพื่อช่วยตรวจสอบ วนลูปทั้งสายอักขระ ถ้าหากเป็นวงเล็บเปิด '(' ให้ push เข้าไปใน stack ถ้าหากเจอวงเล็บปิด ')' ให้ pop ข้อมูลออก 1 ตัว วิธีตรวจสอบคือ ถ้าหากเจอวงเล็บปิดแล้ว stack ว่างแสดงว่าวงเล็บไม่สมดุล และวงเล็บจะสมดุลก็ต่อเมื่อจบลูปแล้วไม่มีวงเล็บเหลือใน stack เลย

### Example Code

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    string str;
    cin >> str;

    bool balanced = true;
    stack<char> stack;
    for(int i=0;i<str.length();i++)
    {
        if(str[i]=='(') stack.push(str[i]);
        else
        {
            if(stack.empty())
            {
                balanced = false;
                break;
            }
            stack.pop();
        }
    }
    if(stack.empty() && balanced) cout << "YES";
    else cout << "NO";
}
```

*time complexity :  $O(n)$*

## Packaging

ใช้ stack ในการเก็บข้อมูลราคาสินค้าและราคาห่อของแต่ละชั้น ( เก็บเป็นคู่ (price,charge) )

- ถ้าเจอวงเล็บเปิดหมายความว่ามีการห่อสินค้าชั้นเพิ่ม ให้ push ข้อมูล (0,ค่าห่อตามชนิด)
- ถ้าเจอตัวอักษรให้เพิ่มราคาตามชนิดสินค้าไปยังราคาสินค้าของข้อมูลที่อยู่ด้านบนสุด
- ถ้าเจอวงเล็บปิด ให้ลบข้อมูลบนสุดออกมากำหนดราคารวม แล้วเพิ่มไปยังข้อมูลบนสุดอันใหม่

ปล. ใช้ pair เป็นประเภทข้อมูล ให้มองว่าเป็น struct ชนิดหนึ่งที่มีข้อมูลแค่ 2 ตัวก็ได้ ตัวแรกเป็น first และตัวที่สองเป็น second ( ดูเพิ่มได้ที่ <https://cplusplus.com/reference/utility/pair/> )

## Example Code

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int a,b,c;
    cin >> a >> b >> c;
    string str;
    cin >> str;

    stack<pair<int,double>> stack; // store price and packaging cost
    stack.push(make_pair(0,0)); // initialize stack
    for(int i=0;i<str.length();i++)
    {
        if(str[i]=='{') stack.push(make_pair(0,0.05));
        else if(str[i]=='(') stack.push(make_pair(0,0.07));
        else if(str[i]=='[') stack.push(make_pair(0,0.1));
        else if(str[i]==')'||str[i]=='}'||str[i]==']')
        {
            pair<int,double> top = stack.top();
            stack.pop();
            stack.top().first += top.first * (1 + top.second);
        }
        else if(str[i]=='A') stack.top().first += a;
        else if(str[i]=='B') stack.top().first += b;
        else if(str[i]=='C') stack.top().first += c;
    }
    cout << stack.top().first;
}
```

time complexity :  $O(n)$