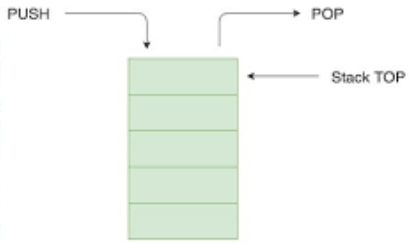
# Ngôn ngữ lập trình C++

## 9 CẤU TRÚC DỮ LIỆU NGĂN XẾP - STACK

### A. LÝ THUYẾT

#### 1. Khái niệm

Ngăn xếp – Stack là một danh sách các phần tử cùng kiểu mà phép bổ sung (**push**) một phần tử và phép lấy ra (**pop**) một phần tử được thực hiện một đầu của danh sách.




Các phần tử được được bổ sung vào sau cùng sẽ được lấy ra đầu tiên, vì vậy ta còn nói việc bổ sung hay lấy ra các phần tử theo quy tắc *Last In – First Out* (*vào sau ra trước*). Phần tử ở trên cùng được gọi là đỉnh ***stack***.

#### 2. Khai báo

Ngôn ngữ lập trình C++ hỗ trợ kiểu dữ liệu stack và được khai báo như sau:

```
stack <kiểu phần tử>  <tên biến stack >;
```

*Ví dụ:*

```
stack <int> myStack;
```

Khai báo một stack mà kiểu các phần tử của nó là kiểu ***int***.

#### 3. Các phương thức trên kiểu dữ liệu stack

| Phương thức | Thực hiện |
|---|---|
| *myStack*.empty() | Trả về giá trị true nếu ngăn xếp rỗng, ngược lại trả về false. |
| *myStack*.push(x) | Đưa phần tử x vào ngăn xếp |
| *myStack*.top() | Trả về giá trị ở đỉnh ngăn xếp |
| *myStack*.pop() | Đưa phần tử ở đỉnh ngăn xếp ra ngoài ngăn xếp |
| *myStack*.size() | Trả về số phần tử hiện có trong ngăn xếp |

Ví dụ:

```cpp
#include<bits/stdc++.h>
using namespace std;
stack<int>  myStack;
int main(){
    //push 1, 2, 3, .., 10 into stack
    for(int i = 1; i <= 10; ++i)
      myStack.push(i);
    cout<<"So phan tu trong stack la: "<< myStack.size()<<'\n';
    while (myStack.empty() == false){
        cout<<"Phan tu o dinh "<<myStack.top()<<'\n';
        myStack.pop();
    }
}
```

*Kết quả:*

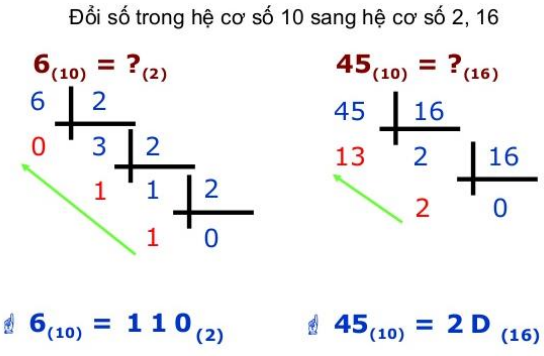
```
So phan tu trong stack la: 10
Phan tu o dinh 10
Phan tu o dinh 9
Phan tu o dinh 8
Phan tu o dinh 7
Phan tu o dinh 6
Phan tu o dinh 5
Phan tu o dinh 4
Phan tu o dinh 3
Phan tu o dinh 2
Phan tu o dinh 1
```

## B. BÀI TẬP

### 1. Chuyển đổi nhị phân

Cho số tự nhiên $n$ ($n \leq 10^9$). Chuyển số $n$ sang hệ nhị phân.

Ví dụ về chuyển đổi sang số nhị phân.




**Dữ liệu** cho trong file BASE2.INP gồm một số tự nhiên $n$ ($n \leq 10^9$).

**Kết quả** ghi ra file BASE2.OUT là số ở hệ nhị phân.

*Ví dụ*:

| BASE2.INP | BASE2.OUT |
|---|---|
| 2 | 10 |

## 2☼. Xâu ngoặc đúng có trọng số

Một xâu kí tự gồm các kí tự mở '(' và kí tự đóng ')' được gọi là một xâu ngoặc đúng khi nó thỏa mãn định nghĩa sau đây.

-Xâu rỗng là một xâu ngoặc đúng.

- Nếu A là một xâu ngoặc đúng thì xâu (A) cũng là một xâu ngoặc đúng.

- Nếu A, B là hai xâu ngoặc đúng thì xâu ghép AB cũng là xâu ngoặc đúng.

*Ví dụ*: Các xâu $(), (()), ()(), ()(()())$ là các xâu ngoặc đúng, xâu $()(())$ không là xâu ngoặc đúng.

Như vậy trong một xâu ngoặc đúng, có sự tương ứng của ngoặc đóng và mở. Ví dụ **(( )( ))** , thì ngoặc mở vị trí 1 tương ứng với đóng ở vị trí 6, mở ở vị trí 2 tương ứng với đóng ở vị trí 3. Các cặp đóng mở cùng màu sẽ tương ứng nhau. Độ dài của xâu ngoặc đúng luôn là một số chẵn vì số ngoặc đóng bằng số ngoặc mở.

Với mỗi kí tự ngoặc đóng và ngoặc mở sẽ được gắn một trọng số là số nguyên.

**Yêu cầu:** Tìm một cặp ngoặc đóng, mở tương ứng mà tổng trọng số của nó là lớn nhất.

**Dữ liệu** cho trong file OPENCLOSE.INP gồm:

- Dòng đầu ghi số nguyên dương $n$ ($n$ là số chẵn, $n \leq 5.10^5$) là độ dài của xâu ngoặc đúng.
- Dòng thứ hai ghi xâu ngoặc đúng có độ dài $n$.
- Dòng thứ 3 ghi $n$ số nguyên $a_1, a_2, .., a_n$ ($|a_i| \leq 10^6$).

**Kết quả** ghi ra file OPENCLOSE.OUT là tổng trọng số lớn nhất của một cặp ngoặc đóng – mở tương ứng.

*Ví dụ*:

| OPENCLOSE.INP | OPENCLOSE.OUT |
|---|---|
| 4<br>()()<br>1 2 3 4 | 7 |

## 3☼. Xóa số

Cho một số nguyên dương $X$.

**Yêu cầu**: Hãy tìm cách xóa $k$ chữ số của $X$ để các chữ số còn lại (vẫn dữ nguyên thứ tự) tạo thành một số có giá trị lớn nhất.

**Dữ liệu** cho trong file văn bản XOASO.INP như sau:

- Dòng đầu tiên ghi số $k$ (là số kí từ cần xóa);
- Dòng thứ hai ghi số nguyên dương $X$ (số chữ số của $X$ không quá $10^6$).

**Kết quả** ghi trong file văn bản XOASO.OUT là dãy các chữ số còn lại (vẫn dữ nguyên thứ tự) tạo thành số lớn nhất.

*Ví dụ:*

| XOASO.INP | XOASO.OUT |
|---|---|
| 4<br>1277624 | 776 |

## 4☼. Bạn đứng trước thấp hơn

Có $N$ học sinh đứng xếp hàng thành một hàng dọc. Bạn thứ $i$ có chiều cao là $H_i$ đơn vị chiều cao ($i$ = 1, 2, 3, …, $N$).

**Yêu cầu:** Với mỗi bạn thứ $i$, hãy tìm bạn thứ $j$ sao cho:

- $1 \le j < i$;
- $H_j < H_i$;
- $j$ lớn nhất có thể.

Ta đặt $S(i) = j$ tìm được. Nếu không có bạn j thỏa màn thì $S(i) = 0$.

**Dữ liệu** cho trong file ShorterFront.Inp gồm:

- Dòng đầu ghi số nguyên dương $N$ ($N \le 5 \times 10^5$).
- Dòng thứ 2 ghi $N$ số nguyên dương $H_1$, $H_2$, …, $H_N$ ($H_i \le 10^6$) là chiều cao của $N$ bạn.

**Kết quả** ghi ra file ShorterFront.Out gồm $N$ dòng. Dòng thứ $i$ ghi giá trị $S(i)$.

*Ví dụ:*

| ShorterFront.Inp | ShorterFront.Out |
|---|---|
| 6<br>10 23 4 4 7 8 | 0<br>1<br>0<br>0<br>4<br>5 |

## 5☼. Sort - Stack

Cấu trúc dữ liệu stack là một cấu trúc dữ liệu quan trọng trong tin học. Trong bài tập này, ta sử dụng cấu trúc dữ liệu stack để sắp xếp.

Cho một xâu kí tự *St* chỉ gồm các chữ số, một stack *SK* và một xâu *S*. Ban đầu stack *SK* rỗng và xâu *S* cũng rỗng. Quá trình sắp xếp được thực hiện như sau:

- Với mỗi bước, ta có thể làm một trong hai công việc:
  + Lấy khỏi xâu *St* kí tự đầu tiên và đưa kí tự đó vào stack (chỉ thực hiện khi xâu *St* không rỗng).
  + Lấy ra khỏi stack *SK* một kí tự và thêm kí tự đó vào cuối xâu *S* (chỉ thực hiện khi stack *SK* không rỗng).
- Quá trình sắp xếp kết thúc khi xâu *St* và stack *SK* đều rỗng, tức là xâu *S* có độ dài bằng xâu *St* ban đầu.

**Yêu cầu:** Tìm cách sắp xếp để được xâu *S* có giá trị nhỏ nhất.

**Dữ liệu** cho trong file SORTSTACK.INP gồm một số dòng (không quá 5 dòng), mỗi dòng ghi một xâu *St* có độ dài không quá $10^5$.

**Kết quả** ghi ra file SORTSTACK.OUT mỗi dòng là một xâu *S* tìm được.

*Ví dụ*:

| SORTSTACK.INP | SORTSTACK.OUT |
|---|---|
| 123<br>2113<br>21314<br>102 | 123<br>1123<br>11324<br>012 |