BÀI TẬP THỰC HÀNH BÀI HỌC SỐ 4.1

Bài 1. Viết chương trình đảo ngược các từ trong chuỗi kí tự đầu vào.

- Input nhập vào từ bàn phím với định dạng:
  - Dòng đầu là số bộ test 0 < t <= 100.
  - T dòng kế tiếp mỗi dòng là 1 bộ test.
- Output hiển thị kết quả mỗi bộ test trên 1 dòng kết quả tìm được.

Ví dụ:

INPUT	OUTPUT
2 I love You Welcome to Branium Academy	You love I Academy Branium to Welcome

Bài 2. Viết chương trình kiểm tra tính đối xứng của mảng các số nguyên n phần tử sử dụng stack. Nếu mảng đối xứng in ra YES ngược lại in ra NO.

- Input nhập vào từ bàn phím với định dạng:
  - Dòng đầu là số bộ test 0 < t <= 100.
  - Mỗi bộ test gồm hai dòng, dòng đầu là số phần tử của mảng input của bộ test đó.
  - Dòng còn lại của bộ test là n phần tử nguyên của mảng phân tách nhau bởi dấu cách.
- Output hiển thị kết quả mỗi bộ test trên 1 dòng với định dạng Test i: kết luận YES hoặc NO tương ứng.

Ví dụ:

INPUT	OUTPUT
2 5 1 2 3 2 1 6 1 2 3 4 2 1	Test 1: YES Test 2: NO

Bài 3. Viết chương trình chuyển biểu thức dạng trung tố sang dạng hậu tố. Giả định các phép toán cần thực hiện trong biểu thức là +, -, *, /, ^. Biểu thức có thể chứa dấu ngoặc để gom nhóm ưu tiên. Thứ tự ưu tiên của các phép toán là lũy thừa = 3, nhân = chia = 2, cộng = trừ = 1.

Gọi thành phần cấu thành nên biểu thức là các phần tử. Nó có thể là các con số, dấu ngoặc (,) hoặc các phép toán. Bỏ qua dấu cách. Quy trình thực hiện chuyển đổi trung tố sang hậu tố:

- Lần lượt lấy từng phần tử e trong biểu thức:
  - Nếu e là toán hạng, hiển thị ra màn hình.

- Nếu e là dấu ngoặc (thì thêm nó vào stack.
- Nếu e là dấu ngoặc), thì pop các phần tử trong stack ra tới khi gặp ngoặc (. Sau đó pop bỏ ngoặc (.
- Nếu e là phép toán và stack không rỗng, trong khi thứ tự ưu tiên của e <= thứ tự ưu tiên của phần tử đầu stack, pop phần tử đầu stack cho hiển thị ra màn hình. Sau đó push e vào stack.

- Trong khi stack còn chưa rỗng, pop các phần tử còn lại hiển thị ra màn hình.
- Input nhập vào từ bàn phím với định dạng:
  - Dòng đầu là số bộ test 0 < t <= 10.
  - Các dòng sau mỗi dòng là một biểu thức trung tố cần chuyển đổi.
- Output: mỗi kết quả ghi trên một dòng biểu thức sau khi chuyển từ trung tố sang hậu tố.

Ví dụ

INPUT	OUTPUT
3 10 * 25 100 * 20 + 6 – 2 20^5 / (5 * 90) + 7	 10 25 * 100 20 * 6 + 2 – 20 5 ^ 5 90 * / 7 +

Bài 4. Viết chương trình tính giá trị biểu thức dạng hậu tố. Bỏ qua các khoảng trắng phân tách từng phần tử. Thuật toán được thực hiện như sau:

Lần lượt lấy từng giá trị e trong biểu thức hậu tố:

- Nếu e là một toán tử t:
  - Pop phần tử đầu stack gán cho biến b.
  - Pop phần tử đầu stack tiếp theo gán cho biến a.
  - Tính kết quả a t b sau đó push kết quả vào stack.
- Nếu e là một toán hạng, push e vào stack.

Sau khi thực hiện xong, phần tử còn lại trong stack là kết quả của biểu thức.

Ví dụ tính giá trị của biểu thức 100 20 * 6 + 2 – ta xét lần lượt từng phần tử e của biểu thức:

- Khi e = '100', đây là một toán hạng, push 100 vào stack. Stack hiện tại: [100].
- Khi e = '20', đây là toán hạng, push 20 vào stack. Stack hiện tại: [100, 20](20 là phần tử top).
- Khi e = '*', đây là một toán tử, b = 20, a = 100, a e b = 100 * 20 = 2000. Push 2000 vào stack. Stack hiện tại: [2000].
- Khi e = '6', đây là toán hạng, push vào stack. Stack hiện tại: [2000, 6].
- Khi e = '+', đây là toán tử, b = 6, a = 2000, a + b = 2006. Push vào stack. Stack hiện tại: [2006].
- Khi e = '2', đây là toán hạng, push vào stack. Stack hiện tại: [2006, 2].

- Khi e = '-', đây là toán tử, pop các phần tử trong stack: a = 2006, b = 2. Tính a – b = 2006-2 = 2004. Push 2004 vào stack. Stack hiện tại: 2004.
- Kết thúc biểu thức, phần tử còn lại trong stack là kết quả: 2004.

Input nhập vào từ bàn phím với định dạng:

- Dòng đầu là số bộ test 0 < t <= 10.
- T dòng sau mỗi dòng là một biểu thức hậu tố.

Output: ghi ra trên t dòng, mỗi dòng là kết quả của bộ test tương ứng.

Ví dụ:

INPUT	OUTPUT
3 10 25 * 100 20 * 6 + 2 – 5 3 + 6 2 * 3 5 * + +	 250 2004 35

Bài 5. Giả sử các phần tử trong stack được sắp xếp tăng dần. Viết chương trình thêm các phần tử vào trong stack sao cho vẫn giữ được tính chất sắp xếp.

- Input gồm nhiều dòng:
  - Dòng đầu là số bộ test t thỏa mãn 0 < t <= 100.
  - T dòng sau mỗi dòng chứa các phần tử của một bộ test phân tách nhau bằng dấu cách.
- Output: kết quả mỗi bộ test in ra trên 2 dòng:
  - Dòng đầu là thứ tự bộ test dạng Test k: trong đó k tăng từ 1.
  - Dòng thứ hai ghi các phần tử của stack theo thứ tự tăng dần, các phần tử phân tách nhau bằng 1 dấu cách.

Ví dụ:

INPUT	OUTPUT
3 1 10 -2 5 24 7 4 5 3 5 2 5 1	Test 1: 1 Test 2: -2 4 5 7 10 24 Test 3: 1 2 3 5 5 5

Bài 6. Cho hai stack đã sắp xếp theo thứ tự tăng dần. Hãy trộn hai stack lại sao cho kết quả là một stack cũng được sắp xếp theo thứ tự tăng dần.

- Input gồm nhiều dòng:

- Dòng đầu là số bộ test t thỏa mãn 0 < t <= 100.
- Mỗi bộ test cho trên 2 dòng. Dòng đầu là các phần tử của stack thứ nhất.
- Dòng còn lại là các phần tử của stack thứ hai.
- Các phần tử của hai stack phân tách nhau bằng 1 vài khoảng trắng.

- Output: kết quả mỗi bộ test in ra trên 2 dòng:
  - Dòng đầu là thứ tự bộ test dạng Test k: trong đó k tăng từ 1.
  - Dòng thứ hai ghi các phần tử của stack kết quả theo thứ tự tăng dần. Các phần tử phân tách nhau bằng 1 khoảng trắng.

Ví dụ:

INPUT	OUTPUT
3 1 1 1 2 3 4 5 6 7 5 5 5 5 -1 0 2 9 15	Test 1: 1 1 Test 2: 1 2 3 4 5 6 7 Test 3: -1 0 2 5 5 5 5 9 15

Trang chủ: https://braniumacademy.net

Bài giải mẫu: click vào đây.