TRƯỜNG ĐẠI HỌC SÀI GÒN

**KHOA CÔNG NGHỆ THÔNG TIN**

**Môn: Cấu trúc dữ liệu và giải thuật**

Tên: Đặng Hữu Đăng Tâm

MSSV: 3124411265

NGĂN XẾP HÀNG ĐỢI

Câu 1:

**1. Hàm Trung\_To\_Hau\_To(string s):**

* **Main Idea**: Chuyển đổi biểu thức toán học từ dạng trung tố (infix) sang dạng hậu tố (postfix) hay còn gọi là ký pháp nghịch đảo Ba Lan.
* **Chức năng**:
  + Sử dụng stack để lưu trữ toán tử và dấu ngoặc.
  + Duyệt qua từng ký tự của biểu thức trung tố.
  + Xử lý toán hạng, toán tử và dấu ngoặc theo quy tắc chuyển đổi trung tố sang hậu tố.
  + Trả về chuỗi biểu thức hậu tố.
* **Input**: Chuỗi s chứa biểu thức trung tố.
* **Output**: Chuỗi chứa biểu thức hậu tố, với các toán hạng và toán tử được phân tách bằng khoảng trắng.

**2. Hàm tinh\_gia\_tri(string hau\_to):**

* **Main Idea**: Tính giá trị của biểu thức hậu tố.
* **Chức năng**:
  + Sử dụng stack để lưu trữ toán hạng.
  + Duyệt qua từng token (toán hạng hoặc toán tử) của biểu thức hậu tố.
  + Khi gặp toán hạng, đẩy nó vào stack.
  + Khi gặp toán tử, lấy hai toán hạng từ stack, thực hiện phép toán và đẩy kết quả trở lại stack.
  + Trả về giá trị cuối cùng trên stack.
* **Input**: Chuỗi hau\_to chứa biểu thức hậu tố.
* **Output**: Giá trị số nguyên là kết quả của biểu thức.

**3. Hàm Dao\_Nguoc\_So(int n):**

* **Main Idea**: Đảo ngược thứ tự các chữ số của một số nguyên.
* **Chức năng**:
  + Sử dụng stack để lưu trữ từng chữ số của số nguyên.
  + Lấy từng chữ số từ số nguyên và đẩy vào stack.
  + In ra các chữ số từ stack theo thứ tự ngược lại.
* **Input**: Số nguyên n.
* **Output**: In ra số nguyên đảo ngược.

**4. Hàm Kiem\_Tra\_Doi\_Xung(string str):**

* **Main Idea**: Kiểm tra xem một chuỗi có phải là chuỗi đối xứng (palindrome) hay không.
* **Chức năng**:
  + Sử dụng stack để lưu trữ các ký tự của chuỗi.
  + Đẩy từng ký tự của chuỗi vào stack.
  + So sánh từng ký tự của chuỗi với ký tự lấy ra từ stack.
  + Trả về true nếu chuỗi đối xứng, false nếu không.
* **Input**: Chuỗi str.
* **Output**: Giá trị true hoặc false.

**5. Hàm Chuyen\_Doi\_Nhi\_Phan(int n):**

* **Main Idea**: Chuyển đổi một số nguyên thập phân sang dạng nhị phân.
* **Chức năng**:
  + Sử dụng stack để lưu trữ các bit nhị phân.
  + Chia số nguyên cho 2 và đẩy phần dư vào stack cho đến khi số nguyên bằng 0.
  + In ra các bit từ stack theo thứ tự ngược lại.
* **Input**: Số nguyên n.
* **Output**: In ra chuỗi bit nhị phân.

**6. Hàm main():**

* **Main Idea**: Điều khiển luồng chương trình và gọi các hàm chức năng.
* **Chức năng**:
  + Nhập liệu từ người dùng.
  + Gọi các hàm Trung\_To\_Hau\_To, tinh\_gia\_tri, Dao\_Nguoc\_So, Kiem\_Tra\_Doi\_Xung và Chuyen\_Doi\_Nhi\_Phan để thực hiện các chức năng tương ứng.
  + In kết quả ra màn hình.
* **Input**:
  + Biểu thức trung tố (chuỗi).
  + Số nguyên.
  + Chuỗi để kiểm tra đối xứng.
  + Số nguyên thập phân để chuyển đổi sang nhị phân.
* **Output**:
  + Biểu thức hậu tố (chuỗi).
  + Kết quả tính toán biểu thức hậu tố (số nguyên).
  + Số nguyên đảo ngược.
  + Thông báo chuỗi có đối xứng hay không.
  + Biểu diễn nhị phân của số nguyên thập phân.

**Testcase 1:**

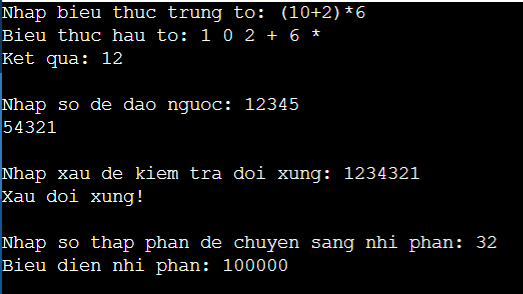
Biểu thức trung tố: (10 + 2) \* 6

Số đảo ngược: 12345

Kiểm tra xâu đối xứng: 1234321

Chuyển đổi nhị phân: 32

**Kết quả:**

****

**Testcase 2:**

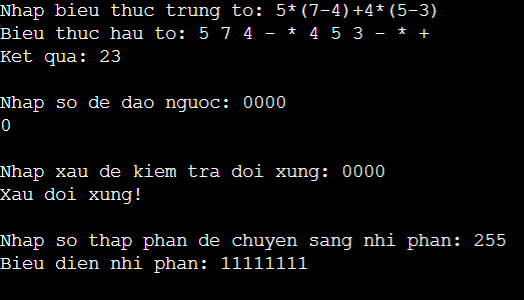
Biểu thức trung tố: 5\*(7-4)+4\*(5-3)

Số đảo ngược: 0000

Kiểm tra xâu đối xứng: 0000

Chuyển đổi nhị phân: 255

**Kết quả:**

****