



## BÀI TẬP THỰC HÀNH BÀI HỌC SỐ 6.4

**Bài 1.** Tính tổng các phần tử trong danh sách gồm  $n$  phần tử.

- Input: cho trong file input1.txt gồm nhiều dòng
  - o Dòng đầu là số bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 2 dòng, dòng đầu là số  $n$ .
  - o Dòng thứ hai là  $n$  phần tử của danh sách phân tách nhau bằng 1 vài khoảng trắng
- Output: lưu vào file output1.txt mỗi bộ test ghi ra trên 2 dòng
  - o Dòng đầu ghi ra thứ tự bộ test dạng Test  $i$ : với  $i$  chạy từ 1.
  - o Dòng còn lại ghi ra tổng các phần tử của danh sách hoặc N INVALID nếu  $n \leq 0$ .

Ví dụ:

INPUT	OUTPUT
3	Test 1:
0	N INVALID
1 2 3	Test 2:
3	6
1 2 3	Test 3:
5	26
5 4 2 6 9	

**Bài 2.** Tính trung bình cộng các phần tử trong danh sách gồm  $n$  phần tử.

- Input: cho trong file input2.txt gồm nhiều dòng
  - o Dòng đầu là số bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 2 dòng, dòng đầu là số  $n$ .
  - o Dòng thứ hai là  $n$  phần tử là các số thực của danh sách phân tách nhau bằng 1 vài khoảng trắng
- Output: ghi vào file output2.txt, mỗi bộ test ghi trên 2 dòng
  - o Dòng đầu ghi ra thứ tự bộ test dạng Test  $i$ : với  $i$  chạy từ 1.
  - o Dòng còn lại ghi ra TBC các phần tử của danh sách hoặc N INVALID nếu  $n \leq 0$ .

Ví dụ:

INPUT	OUTPUT
3	Test 1:
-6	N INVALID
1 2 3	Test 2:
3	2
1 2 3	Test 3:
5	5.2
5 4 2 6 9	



**Bài 3.** Cho số nguyên  $n$  và danh sách gồm  $n$  số thực. Hãy tính trung bình cộng các phần tử ở vị trí chẵn trong danh sách. Lưu ý vị trí trong danh sách bắt đầu từ 0.

- Input: cho trong file input3.txt gồm nhiều dòng.
  - o Dòng đầu là số bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 2 dòng, dòng đầu là số  $n$ .
  - o Dòng thứ hai là  $n$  phần tử của danh sách phân tách nhau bằng 1 vài khoảng trắng.
- Output: ghi vào file output3.txt, kết quả mỗi bộ test ghi trên 2 dòng
  - o Dòng đầu là thứ tự bộ test dạng Test  $k$ : với  $k$  tính từ 1.
  - o Dòng còn lại ghi ra TBC các phần tử ở vị trí chẵn của danh sách hoặc N INVALID nếu  $n \leq 0$ .

Ví dụ:

INPUT	OUTPUT
3 -3 1 2 3 3 1 2 3 5 5 4 2 6 2	Test 1: N INVALID Test 2: 2 Test 3: 3

**Bài 4.** Cho số nguyên  $n$  và danh sách gồm  $n$  số nguyên. Liệt kê tất cả các phần tử là số nguyên tố trong danh sách theo cặp(chỉ số, giá trị).

- Input: cho trong file input4.txt gồm nhiều dòng
  - o Dòng đầu là số bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 2 dòng, dòng đầu chứa số nguyên  $n$ .
  - o Dòng thứ hai là  $n$  phần tử của danh sách phân tách nhau bằng 1 vài khoảng trắng.
- Output: lưu vào file output4.txt mỗi bộ test ghi kết quả trên 2 dòng
  - o Dòng đầu là số thứ tự bộ test dạng Test  $k$ : với  $k$  bắt đầu từ 1.
  - o Dòng còn lại ghi ra các phần tử là số nguyên tố trong danh sách theo cặp(chỉ số, giá trị) cách nhau bởi 1 dấu cách hoặc N INVALID nếu  $n \leq 0$ .

Ví dụ:

INPUT	OUTPUT
3 -9 1 2 3 3	Test 1: N INVALID Test 2: (1, 2) (2, 3)



1 2 3 7 0 4 2 6 2 9 11	Test 3: (2, 2) (4, 2) (6, 11)
------------------------------	----------------------------------

**Bài 5.** Cho số nguyên  $n$  và danh sách gồm  $n$  số nguyên. Liệt kê các số chính phương trong danh sách theo cặp (chỉ số, giá trị).

- Input: cho trong file input5.txt gồm nhiều dòng
  - o Dòng đầu là số lượng bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 2 dòng, dòng đầu là số  $n$ .
  - o Dòng thứ hai là  $n$  phần tử của danh sách phân tách nhau bằng 1 vài khoảng trắng.
- Output: ghi vào file output5.txt, mỗi bộ test ghi kết quả trên 2 dòng
  - o Dòng đầu là số thứ tự của bộ test dạng Test  $k$ : với  $k$  chạy từ 1.
  - o Dòng còn lại ghi ra các phần tử là số chính phương trong danh sách theo cặp (chỉ số, giá trị) cách nhau bởi 1 dấu cách hoặc N INVALID nếu  $n \leq 0$ .

Ví dụ:

INPUT	OUTPUT
3 -9 1 2 3 4 1 2 3 4 7 0 4 2 6 2 -9 -11	Test 1: N INVALID Test 2: (0, 1) (3, 4) Test 3: (0, 0) (1, 4)

**Bài 6.** Cho số nguyên  $n$  và danh sách gồm  $n$  số nguyên. Liệt kê các số thuận nghịch trong danh sách. Giả định rằng số thuận nghịch là các số có từ 2 chữ số trở lên sao cho khi ta đọc các chữ số của số đó từ trái sang phải hay từ phải sang trái thì đều nhận được một giá trị không đổi. Không xét dấu của giá trị.

- Input: cho trong file input6.txt gồm nhiều dòng
  - o Dòng đầu là số lượng bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 2 dòng, dòng đầu là số  $n$ .
  - o Dòng thứ hai là  $n$  phần tử của danh sách phân tách nhau bằng 1 vài khoảng trắng.
- Output: lưu vào file output6.txt, kết quả mỗi bộ test ghi ra trên 2 dòng
  - o Dòng đầu là số thứ tự bộ test dạng Test  $k$ : trong đó  $k$  bắt đầu từ 1.
  - o Dòng còn lại ghi ra các phần tử là số thuận nghịch cách nhau bởi 1 dấu cách hoặc N INVALID nếu  $n \leq 0$ .



Ví dụ:

INPUT	OUTPUT
3 -9 1 2 3 4 1 22 39 47 8 0 44 2 6 2 -9 -11 99	Test 1: N INVALID Test 2: 1 22 Test 3: 0 44 2 6 2 -9 -11 99

**Bài 7.** Cho số nguyên  $n$  và danh sách gồm  $n$  số nguyên. Hãy tìm giá trị lớn nhất và nhỏ nhất trong danh sách. Giả sử rằng nếu tất cả các phần tử trong danh sách có cùng 1 giá trị thì không tồn tại giá trị lớn nhất, nhỏ nhất.

- Input: cho trong file input7.txt gồm nhiều dòng
  - o Dòng đầu là số lượng bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 2 dòng, dòng đầu là số  $n$ .
  - o Dòng thứ hai là  $n$  phần tử của danh sách phân tách nhau bằng 1 vài khoảng trắng
- Output: lưu vào file output7.txt, kết quả mỗi bộ test ghi ra trên 2 dòng
  - o Dòng đầu là thứ tự bộ test dạng Test  $k$ : trong đó  $k$  tính từ 1.
  - o Dòng còn lại ghi ra theo thứ tự lần lượt giá trị nhỏ nhất và giá trị lớn nhất cách nhau bởi 1 dấu cách hoặc N INVALID nếu  $n \leq 0$ . Trường hợp không có giá trị lớn nhất nhỏ nhất, in ra NOT AVAILABLE.

Ví dụ:

INPUT	OUTPUT
3 -9 1 2 3 4 1 32 29 47 9 10 4 12 6 32 -7 -11 99 3	Test 1: N INVALID Test 2: 1 47 Test 3: -11 99

**Bài 8.** Cho số nguyên  $n$  và danh sách gồm  $n$  số nguyên. Hãy tìm giá trị lớn thứ hai và nhỏ thứ hai trong danh sách. Giả sử rằng nếu tất cả các phần tử danh sách có giá trị bằng nhau thì không tồn tại giá trị lớn nhất, lớn thứ hai và nhỏ nhất, nhỏ thứ hai.

- Input: cho trong file input8.txt gồm nhiều dòng
  - o Dòng đầu là số lượng bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 2 dòng, dòng đầu là số  $n$ .



- Dòng thứ hai là n phần tử của danh sách phân tách nhau bằng 1 vài khoảng trắng
- Output: lưu vào file output8.txt, kết quả mỗi bộ test ghi ra trên hai dòng.
  - Dòng đầu ghi ra thứ tự bộ test dạng Test i: trong đó i chạy từ 1.
  - Dòng còn lại ghi ra theo thứ tự lần lượt giá trị nhỏ thứ hai và giá trị lớn thứ hai cách nhau bởi 1 dấu cách hoặc N INVALID nếu  $n \leq 0$ . Nếu không có giá trị thỏa mãn thì in ra NOT AVAILABLE

Ví dụ:

INPUT	OUTPUT
4 -9 1 2 3 4 1 32 29 47 9 10 4 12 6 32 -7 -11 99 3 5 9 9 9 9	Test 1: N INVALID Test 2: 29 32 Test 3: -7 32 Test 4: NOT AVAILABLE

**Bài 9.** Cho số nguyên n, x và danh sách các số nguyên. Hãy tìm số lần xuất hiện của x trong danh sách.

- Input: cho trong file input9.txt gồm nhiều dòng
  - Dòng đầu là số lượng bộ test t thỏa mãn  $0 < t \leq 100$ .
  - Mỗi bộ test gồm 2 dòng, dòng đầu là 2 số n, x cách nhau bằng dấu cách.
  - Dòng thứ hai là n phần tử của danh sách phân tách nhau bằng 1 vài khoảng trắng.
- Output: lưu vào file output9.txt, kết quả mỗi bộ test gồm 2 dòng
  - Dòng đầu là thứ tự bộ test dạng Test k: với k chạy từ 1.
  - Dòng còn lại ghi ra số lần xuất hiện của x hoặc N INVALID nếu  $n \leq 0$ .

Ví dụ:

INPUT	OUTPUT
4 -9 3 1 2 3 4 25 1 32 29 47 9 3 3 4 5 6 2 3 -11 9 3 5 9 9 9 9 9	Test 1: N INVALID Test 2: 0 Test 3: 3 Test 4: 5



**Bài 10.** Cho số nguyên  $n$ ,  $k \neq 0$  và danh sách gồm  $n$  số nguyên. Đếm số phần tử của danh sách chia hết cho  $k$ .

- Input: cho trong file input10.txt gồm nhiều dòng
  - o Dòng đầu là số lượng các bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 2 dòng, dòng đầu là 2 số  $n, k$  cách nhau bằng dấu cách.
  - o Dòng thứ hai là  $n$  phần tử của danh sách phân tách nhau bằng 1 vài khoảng trắng.
- Output: ghi vào file output10.txt. Mỗi bộ test ghi trên 2 dòng.
  - o Dòng đầu là số thứ tự bộ test dạng Test  $k$ : với  $k$  tăng từ 1.
  - o Dòng còn lại là số phần tử chia hết cho  $k$  hoặc N INVALID nếu  $n \leq 0$ . K INVALID nếu  $k = 0$ .

Ví dụ:

INPUT	OUTPUT
4	Test 1:
-9 3	N INVALID
1 2 3	Test 2:
4 0	K INVALID
1 32 29 47	Test 3:
9 3	5
3 4 5 6 2 3 -11 -9 -3	Test 4:
5 20	0
9 9 9 9 9	

**Bài 11.** Cho số nguyên  $n$  và danh sách gồm  $n$  số nguyên. Kiểm tra xem danh sách bạn vừa nhập có phải là danh sách đối xứng hay không. Danh sách đối xứng là danh sách có các cặp phần tử đối xứng qua phần tử ở chính giữa.

- Input: cho trong file input11.txt gồm nhiều dòng
  - o Dòng đầu là số lượng các bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 2 dòng, dòng đầu là số nguyên  $n$ .
  - o Dòng thứ hai là  $n$  phần tử của danh sách phân tách nhau bằng 1 vài khoảng trắng.
- Output: lưu vào file output11.txt, kết quả mỗi bộ test gồm 2 dòng.
  - o Dòng đầu ghi thứ tự bộ test dạng Test  $k$ : trong đó  $k$  chạy từ 1.
  - o Dòng còn lại ghi ra YES, NO tương ứng hoặc N INVALID nếu  $n \leq 0$ .

Ví dụ:

INPUT	OUTPUT
-------	--------



3 -9 1 2 3 4 1 32 29 47 4 1 2 2 1	Test 1: N INVALID Test 2: NO Test 3: YES
---	---

**Bài 12.** Cho số nguyên dương  $n$  và danh sách gồm  $n$  số nguyên. Sắp xếp các phần tử trong danh sách theo thứ tự tăng dần.

- Input: cho trong file input12.txt gồm nhiều dòng
  - o Dòng đầu là số lượng các bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 2 dòng, dòng đầu là số nguyên  $n$ .
  - o Dòng thứ hai là  $n$  phần tử của danh sách phân tách nhau bằng 1 vài khoảng trắng.
- Output: ghi vào file output12.txt, kết quả mỗi bộ test gồm 2 dòng.
  - o Dòng đầu ghi thứ tự bộ test dạng Test  $k$ : trong đó  $k$  chạy từ 1.
  - o Dòng còn lại ghi ra các phần tử sau khi sắp xếp tăng dần.

Ví dụ:

INPUT	OUTPUT
2 5 5 2 1 4 3 6 1 2 1 2 1 2	Test 1: 1 2 3 4 5 Test 2: 1 1 1 2 2 2

**Bài 13.** Cho số nguyên dương  $n$  và danh sách gồm  $n$  số thực. Sắp xếp các phần tử trong danh sách theo thứ tự giảm dần.

- Input: cho trong file input13.txt, gồm nhiều dòng
  - o Dòng đầu là số lượng các bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 2 dòng, dòng đầu là số nguyên  $n$ .
  - o Dòng thứ hai là  $n$  phần tử của danh sách phân tách nhau bằng 1 vài khoảng trắng.
- Output: ghi ra file output13.txt, kết quả mỗi bộ test gồm 2 dòng.
  - o Dòng đầu ghi thứ tự bộ test dạng Test  $k$ : trong đó  $k$  chạy từ 1.
  - o Dòng còn lại ghi ra các phần tử sau khi sắp xếp.

Ví dụ:



INPUT	OUTPUT
2 5 5.2 2.4 1 4 3.1 6 1 2 1 2 1 2	Test 1: 5.2 4 3.1 2.4 1 Test 2: 2 2 2 1 1 1

**Bài 14.** Cho chuỗi kí tự gồm nhiều từ cách nhau bằng một vài khoảng trắng. Sắp xếp các từ trong chuỗi theo thứ tự từ điển.

- Input: cho trong file input14.txt gồm nhiều dòng
  - o Dòng đầu là số lượng các bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 1 dòng chứa chuỗi kí tự đầu vào tùy ý.
- Output: ghi ra file output14.txt, kết quả mỗi bộ test gồm 2 dòng.
  - o Dòng đầu ghi thứ tự bộ test dạng Test k: trong đó k chạy từ 1.
  - o Dòng còn lại ghi ra các phần tử sau khi sắp xếp.

Ví dụ:

INPUT	OUTPUT
2 How are you today I Love Learning Python At Branium Academy	Test 1: How are today you Test 2: Academy At Branium I Learning Love Python

**Bài 15.** Cho chuỗi kí tự gồm nhiều từ cách nhau bằng một vài khoảng trắng. Sắp xếp các từ trong chuỗi theo thứ tự tăng dần độ dài của từ.

- Input: cho trong file input15.txt gồm nhiều dòng
  - o Dòng đầu là số lượng các bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 1 dòng chứa chuỗi kí tự đầu vào tùy ý.
- Output: ghi ra file output15.txt, kết quả mỗi bộ test gồm 2 dòng.
  - o Dòng đầu ghi thứ tự bộ test dạng Test k: trong đó k chạy từ 1.
  - o Dòng còn lại ghi ra các phần tử sau khi sắp xếp.

Ví dụ:

INPUT	OUTPUT
-------	--------





2 How are you today I Love Learning Python At Branium Academy	Test 1: How are you today Test 2: I At Love Python Branium Academy Learning
---	--

**Bài 16.** Cho chuỗi kí tự gồm nhiều từ cách nhau bằng một vài khoảng trắng. Sắp xếp các từ trong chuỗi theo thứ tự giảm dần độ dài của từ.

- Input: cho trong file input16.txt gồm nhiều dòng
  - o Dòng đầu là số lượng các bộ test  $t$  thỏa mãn  $0 < t \leq 100$ .
  - o Mỗi bộ test gồm 1 dòng chứa chuỗi kí tự đầu vào tùy ý.
- Output: ghi ra file output16.txt, kết quả mỗi bộ test gồm 2 dòng.
  - o Dòng đầu ghi thứ tự bộ test dạng Test k: trong đó k chạy từ 1.
  - o Dòng còn lại ghi ra các phần tử sau khi sắp xếp.

Ví dụ:

INPUT	OUTPUT
2 How are you today I Love Learning Python At Branium Academy	Test 1: today How are you Test 2: Learning Academy Branium Python Love At I

**Bài 17.** Cho một ma trận gồm các phần tử là số nguyên cấp  $[m \times n]$  với  $m, n > 0$ . Hãy nhập vào các phần tử của ma trận sau đó hiển thị ma trận ra màn hình.

- Input: cho trong file input17.txt gồm 2 dòng
  - o Dòng đầu chứa số nguyên  $m, n$  cách nhau bằng 1 vài khoảng trắng.
  - o Dòng thứ hai là  $m \times n$  phần tử của mảng
- Output: ghi ra file output17.txt gồm  $m + 1$  dòng
  - o Dòng đầu là cấp của ma trận  $m, n$
  - o  $M$  dòng còn lại, mỗi dòng là một hàng các phần tử của ma trận
  - o Nếu  $m$  hoặc  $n$  âm, báo ERROR

Ví dụ:

INPUT	OUTPUT
-3 3 0 3 6 2 6 5	ERROR
3 4	3 4



2 3 6 5 9 5 4 7 1 0 2 3	2 3 6 5 9 5 4 7 1 0 2 3
-------------------------	-------------------------------

**Bài 18.** Cho ma trận các số thực cấp  $[m \times n]$ . Hãy viết chương trình thực hiện các chức năng sau:

- Nhập vào hai số  $m, n$  và các phần tử của ma trận
  - Tìm ma trận chuyển vị của ma trận này
  - Hiển thị ma trận gốc và ma trận chuyển vị ra màn hình
- Input: cho trong file input18.txt gồm nhiều dòng
    - Dòng đầu là hai số  $m, n$  theo thứ tự cách nhau bằng 1 vài khoảng trắng.
    - M dòng kế tiếp mỗi dòng chứa  $n$  phần tử của từng hàng trong ma trận phân tách nhau bằng một vài khoảng trắng.
  - Output: ghi ra file output18.txt, nếu  $m, n$  không hợp lệ, in ra ERROR. Ngược lại, kết quả gồm nhiều dòng
    - Dòng đầu là cấp của ma trận cách nhau bằng 1 dấu cách
    - Các dòng kế tiếp là ma trận gốc
    - Tiếp theo là một dòng trống
    - Sau đó là cấp của ma trận chuyển vị
    - Các dòng sau đó là ma trận chuyển vị
    - Ma trận cấp  $[m \times n]$  được in ra sao cho đủ  $m$  hàng, mỗi hàng phải in đủ  $n$  phần tử

Ví dụ:

INPUT	OUTPUT
-2 2 2 3 4 5  3 4 1 2 3 4 5 6 7 8 9 0 5 7	ERROR    3 4 1 2 3 4 5 6 7 8 9 0 5 7  4 3 1 5 9 2 6 0 3 7 5 4 8 7

**Bài 19.** Cho hai ma trận  $A$  cấp  $[m \times n]$  và  $B$   $[l \times k]$ . Tìm ma trận tổng  $A + B$ .

- Input: cho trong file input19.txt gồm nhiều dòng



- Dòng đầu là 4 số m, n, l, k theo thứ tự cách nhau bằng một vài khoảng trắng.
- m dòng tiếp theo, mỗi dòng là n phần tử của ma trận A
- l dòng kế tiếp, mỗi dòng gồm k phần tử của ma trận B
- Output: ghi ra file output19.txt, gồm nhiều dòng
  - Dòng đầu là cấp của ma trận tổng.
  - Các dòng còn lại, mỗi dòng là các phần tử của một hàng trong ma trận kết quả. Các phần tử phân tách nhau bằng 1 khoảng trắng.
  - Nếu hai ma trận không cộng được thì in ra INVALID ACTION

Ví dụ:

INPUT	OUTPUT
2 2 2 3 2 3 5 4 2 3 5 6 4 5 8	INVALID ACTION
3 3 3 3 1 2 3 4 5 6 7 8 9 1 2 3 5 2 1 1 0 2	3 3 2 4 6 9 7 7 8 8 11

**Bài 20.** Viết chương trình vẽ hình dấu cộng như sau, lưu vào file output20.txt:

```

      * * * * *
      *       *
      *       *
      * * * * *   * * * * *
      *                   *
      * * * * *   * * * * *
      *       *
      *       *
      * * * * *
  
```

**Bài 21.** Đọc vào hai số nguyên m, n sau đó vẽ hình chữ nhật đặc m hàng n cột bằng các dấu \*. Nếu m hoặc n <= 0 in ra ERROR.

- Input: cho trong file input20.txt, gồm nhiều dòng.
  - Mỗi dòng là một bộ test chứa hai số nguyên m, n.



- Output: ghi ra file output21.txt kết quả hình chữ nhật đặc kích thước  $m \times n$  tương ứng hoặc ERROR nếu cạnh không hợp lệ.

Ví dụ minh họa:

INPUT	OUTPUT
-9 5	ERROR
4 5	<pre>* * * * * * * * * * * * * * * * * * * *</pre>

**Bài 22.** Đọc vào hai số nguyên  $m, n$  sau đó vẽ hình chữ nhật rỗng  $m$  hàng  $n$  cột bằng các dấu  $*$ . Nếu  $m$  hoặc  $n \leq 0$  in ra ERROR.

- Input: cho trong file input22.txt gồm nhiều dòng.
  - Mỗi dòng là 1 bộ test chứa hai số nguyên  $m, n$ .
- Output: ghi ra file output22.txt hình chữ nhật rỗng kích thước  $m \times n$  tương ứng hoặc ERROR nếu cạnh không hợp lệ.

Ví dụ minh họa:

INPUT	OUTPUT
-9 5	ERROR
4 5	<pre>* * * * * *       * *       * * * * * *</pre>

**Bài 23.** Đọc số nguyên  $h$  sau đó vẽ tam giác vuông góc trái dưới có chiều cao  $h$  bằng các dấu  $*$ . Nếu  $h \leq 0$  thì ghi ERROR.

- Input: cho trong file input23.txt gồm nhiều dòng.
  - Mỗi dòng là một bộ test chứa chiều cao  $h$ .
- Output: ghi ra file output23.txt hình tam giác vuông góc trái dưới tương ứng hoặc ERROR nếu giá trị  $h$  không hợp lệ.

Ví dụ minh họa:

INPUT	OUTPUT
-8	ERROR



4	<pre>* * * * * * * * * *</pre>
---	--------------------------------

**Bài 24.** Đọc số nguyên  $h$  sau đó vẽ tam giác cân với chiều cao  $h$ . Nếu  $h \leq 0$  thì ghi ERROR.

- Input: cho trong file input24.txt gồm nhiều dòng.
  - Mỗi dòng là một bộ test chứa số nguyên  $h$ .
- Output: ghi ra file output.txt hình tam giác cân tương ứng hoặc ERROR nếu  $h$  không hợp lệ.

Ví dụ minh họa:

INPUT	OUTPUT
-8	ERROR
5	<pre>      *      * * *     * * * * *    * * * * * *   * * * * * * *  * * * * * * * *</pre>

**Bài 25.** Đọc số nguyên  $h$  sau đó vẽ tam giác số đối xứng chiều cao  $h$ . Nếu  $h \leq 0$  thì ghi ERROR.

- Input: cho trong file input25.txt gồm nhiều dòng.
  - Mỗi dòng là một bộ test chứa số nguyên  $h$ .
- Output: ghi ra file output25.txt hình tam giác cân bằng các con số tương ứng hoặc ERROR nếu  $h$  không hợp lệ.

Ví dụ minh họa:

INPUT	OUTPUT
-8	ERROR
5	<pre>          1         1 2 1       1 2 3 2 1     1 2 3 4 3 2 1   1 2 3 4 5 4 3 2 1</pre>



**Bài 26.** Đọc số nguyên  $h$  sau đó vẽ tam giác cân rỗng chiều cao  $h$ . Nếu  $h \leq 0$  thì ghi ERROR.

- Input: cho trong file input26.txt gồm nhiều dòng.
  - Mỗi dòng là một bộ test chứa số nguyên  $h$ .
- Output: ghi kết quả ra file output26.txt hình tam giác cân rỗng bằng số 1 tương ứng hoặc ERROR.

Ví dụ minh họa:

INPUT	OUTPUT
-8	ERROR
5	<pre>      1      1 1     1 1 1    1 1 1 1   1 1 1 1 1</pre>

**Bài 27.** Cho hình chữ nhật kích thước  $6 \times 7$ . Vẽ hình tim, lưu kết quả ra file output27.txt. Ví dụ:

```
  * * * *
 * * * * *
 * * * * *
 * * * *
  * * *
    * *
```

**Bài 28.** Vẽ hình vuông rỗng chiều dài cạnh  $h$  và các đường chéo của nó.

- Input: cho trong file input28.txt gồm nhiều dòng.
  - Mỗi dòng là một bộ test chứa số nguyên  $h$ .
- Output: ghi ra file output28.txt hình vuông rỗng và hai đường chéo bằng các dấu  $*$  của bộ test tương ứng. Nếu  $h$  không hợp lệ, in ra ERROR.



Ví dụ minh họa:

INPUT	OUTPUT
-8	ERROR
6	<pre>* * * * * * *   * * *   * * * *   * * * * *   * * * * * * *</pre>

**Bài 29.** Người gồm các thông tin về số CMND/CCCD, họ và tên, ngày sinh. Hành động của người gồm hiển thị thông tin bản thân, làm việc. Sinh viên là người và có thêm thông tin về mã sinh viên, điểm TB tích lũy, chuyên ngành. Mã sinh viên gồm cả chữ và số tự động tăng từ SV1000, SV1001, ... Sinh viên có thêm hành động làm bài thi cuối kỳ, đăng ký môn học. Môn học gồm các thông tin về mã môn học là một số nguyên tự tăng từ 1000, tên môn học, số tín chỉ. Lớp đăng ký môn học chứa thông tin mã đăng ký là một số nguyên có 3 chữ số tăng từ 100, môn học và sinh viên tương ứng kèm theo thời gian đăng ký lấy từ hệ thống.

Cho biết thông tin sinh viên lưu trong file STUDENT.DAT, thông tin môn học lưu trong file SUBJECT.DAT, thông tin đăng ký môn học trong file REGISTER.DAT. Mỗi file có nhiều bản ghi, mỗi bản ghi là thông tin của một đối tượng hoàn chỉnh. Mỗi đối tượng gồm nhiều trường thông tin và mỗi thông tin đó được ghi trên 1 dòng theo thứ tự các thuộc tính được mô tả ở trên. Ví dụ:

STUDENT.DAT	
1	012305999990 ← CMND/CCCD
2	Trần Tuấn Anh ← Họ tên
3	25/03/2005 ← Ngày sinh
4	SV1000 ← Mã SV
5	3.25 ← Điểm TB
6	CNTT ← Chuyên ngành
7	012305999993
8	Ngô Quang Khải
9	15/07/2005
10	SV1001
11	3.55
12	CNTT
13	012305999991
14	Nguyễn Thanh Thúy
15	10/12/2005
16	SV1002
17	3.75
18	CNTT



Riêng bản ghi trong file REGISTER.DAT có định dạng: mã đăng ký, mã môn, mã sinh viên, thời gian đăng ký.

Viết chương trình đọc vào danh sách sinh viên, môn học, lớp đăng ký từ file. Chương trình cần có menu chọn cho phép thực hiện các chức năng sau:

- 1) Thêm mới sinh viên vào danh sách sinh viên.
- 2) Thêm mới môn học vào danh sách môn học.
- 3) Thêm mới một bản đăng ký môn học vào danh sách bảng đăng ký. Lưu ý với mỗi sinh viên với 1 môn học chỉ được xuất hiện 1 lần trong danh sách bảng đăng ký.
- 4) Sắp xếp danh sách sinh viên theo tên tăng dần, họ tăng dần.
- 5) Sắp xếp danh sách môn học theo tên tăng dần.
- 6) Sắp xếp bảng đăng ký theo thứ tự từ đăng ký sớm nhất đến muộn nhất.
- 7) Hiển thị danh sách sinh viên dạng bảng gồm các hàng, cột ngay ngắn.
- 8) Hiển thị danh sách môn học dạng bảng gồm các hàng, cột.
- 9) Hiển thị danh sách đăng ký gồm mã bản đăng ký, mã SV, tên SV, mã môn, tên môn và thời gian đăng ký dạng dd/MM/yyyy ví dụ 22/12/2025 18:24:36.
- 10) Liệt kê danh sách môn học mà sinh viên đã đăng ký trong danh sách đăng ký theo mã sinh viên. Thông tin cần hiển thị gồm thông tin môn học và thời gian đăng ký sắp xếp theo thời gian đăng ký từ sớm đến muộn.
- 11) Liệt kê danh sách sinh viên đã đăng ký môn học nào đó trong danh sách đăng ký môn học theo mã môn học.
- 12) Thống kê số lượng sinh viên đăng ký theo từng môn học từ nhiều nhất đến ít nhất.
- 13) Cho biết thông tin bản ghi của bản đăng ký sớm nhất.
- 14) Cho biết thông tin bản ghi của bản đăng ký muộn nhất.
- 15) Lưu các danh sách vào file với định dạng đã cho trong từng file tương ứng. Các bản ghi phải là duy nhất không trùng lặp.
- 16) Kết thúc chương trình.

**Bài 30.** Người gồm các thông tin về số CMND/CCCD, họ và tên, ngày sinh. Hành động của người gồm hiển thị thông tin bản thân, làm việc. Sinh viên là người và có thêm thông tin về mã sinh viên, điểm TB tích lũy, chuyên ngành. Mã sinh viên gồm cả chữ và số tự động tăng từ SV1000, SV1001, ... Sinh viên có thêm hành động làm bài thi cuối kỳ, đăng ký môn học. Giảng viên kế thừa từ người có thêm mã giảng viên gồm cả chữ và số tự tăng từ GV001, mức lương, chuyên môn. Môn học gồm các thông tin về mã môn học là một số nguyên tự tăng từ 1000, tên môn học, số tín chỉ. Bảng điểm gồm có thông tin về mã bảng điểm là số nguyên tự tăng từ 100, sinh viên, điểm TB môn học và xếp loại. Mỗi lớp học gồm có mã lớp tự tăng từ C100, môn học, giảng viên, phòng học và danh sách bảng điểm.

Trong đó dữ liệu về sinh viên lưu trong file STUDENT.DAT. Dữ liệu môn học lưu trong file SUBJECT.DAT. Dữ liệu giảng viên lưu trong file LECTURER.DAT. Dữ liệu về Bảng điểm lưu trong file TRANSCRIPT.DAT. Dữ liệu về các lớp học lưu trong file COURSE.DAT. Trong mỗi file có thể có nhiều bản ghi. Mỗi bản ghi gồm dữ liệu hoàn chỉnh của một đối tượng. Mỗi trường dữ liệu của





đối tượng ghi trên một dòng theo thứ tự các thuộc tính mô tả phía trên. Hãy load dữ liệu từ các file vào danh sách đối tượng tương ứng khi chương trình được kích hoạt. Chương trình cần có menu cho phép lựa chọn thực hiện các chức năng sau:

- 1) Thêm mới một sinh viên vào danh sách sinh viên.
- 2) Thêm mới một giảng viên vào danh sách giảng viên.
- 3) Thêm mới một môn học vào danh sách môn học.
- 4) Thêm mới một lớp học vào danh sách lớp bằng cách nhập mã môn học, mã giảng viên cùng thông tin liên quan.
- 5) Lập danh sách lớp bằng cách nhập mã lớp, mã sinh viên và điểm TB môn của sinh viên. Xác định học lực cho sinh viên từ điểm TB môn. Lưu ý mỗi sinh viên trong danh sách một lớp học không xuất hiện quá 1 lần. Quy ước điểm TB môn vs phân loại học lực:
  - a. Dưới 4.0: Yếu.
  - b. Từ 4.0 đến dưới 6.5: Trung bình.
  - c. Từ 6.5 đến dưới 8.0: Khá.
  - d. Từ 8.0 đến dưới 9.0: Giỏi.
  - e. Từ 9.0 đến 10.0: Xuất sắc.
- 6) Hiển thị danh sách sinh viên ở dạng bảng.
- 7) Hiển thị danh sách môn học ở dạng bảng.
- 8) Hiển thị danh sách các lớp học ở dạng bảng.
- 9) Hiển thị danh sách bảng điểm của sinh viên trong lớp. Thông tin cần hiển thị gồm mã lớp, mã môn, tên môn, mã sv, tên sv, điểm TB và xếp loại.
- 10) Sắp xếp danh sách sinh viên theo tên tăng dần.
- 11) Sắp xếp danh sách sinh viên theo ngày sinh từ già đến trẻ.
- 12) Sắp xếp môn học theo tên môn học.
- 13) Sắp xếp danh sách các lớp học theo tên phòng học.
- 14) Sắp xếp danh sách sinh viên trong lớp theo điểm giảm dần, tìm lớp theo mã lớp.
- 15) Liệt kê các sinh viên có điểm cao nhất trong một lớp theo mã lớp.
- 16) Liệt kê các sinh viên đạt điểm cao nhất theo từng môn học.
- 17) Tìm các sinh viên trong lớp theo điểm nhập vào từ bàn phím. Lớp cần tìm xác định qua mã lớp.
- 18) Thống kê số lượng các sinh viên trong một lớp theo học lực giảm dần. Chọn lớp cần thống kê theo mã lớp.
- 19) Thống kê số sinh viên có học lực giỏi và xuất sắc theo từng môn học.
- 20) Lưu các danh sách vào file tương ứng. Các bản ghi phải là duy nhất không trùng lặp.
- 21) Kết thúc chương trình.

**Bài 31.** Nhân viên gồm thông tin về mã nhân viên, họ và tên, email, số điện thoại, mức lương. Hành động của nhân viên gồm: checkin, checkout, tính lương, làm việc. Giám đốc là nhân viên có thêm chức vụ, nhiệm kỳ, tổng lương thực lĩnh theo quý. Giám đốc thường đi họp với đối tác, ký duyệt các quyết định quan trọng của công ty. Lập trình viên là nhân viên có thêm chuyên môn, số ngôn ngữ lập trình đã thành thạo, số dự án đã tham gia tại công ty, KPI theo tháng. Lập trình



viên phải nhận task, code, fix lỗi, báo cáo tiến độ công việc cho sếp. Tester là nhân viên có thêm chuyên môn, công cụ sử dụng, số lỗi đã phát hiện trong tháng, số testcase đã hoàn thành trong tháng. Tester phải nhận các dự án, viết testcase, test, báo cáo lỗi cho lập trình viên sửa đổi cập nhật. Trong đó mã Giám đốc có dạng Mxxx với xxx là 3 chữ số tự động tăng từ 100, ví dụ M100. Mã lập trình viên có dạng Dxxx với xxx là 3 chữ số tự tăng từ 100, ví dụ D105. Mã tester có dạng Txxx trong đó xxx là 3 số tự động tăng bắt đầu từ 100, ví dụ T101. Mỗi khi thêm mới 1 nhân viên thì giá trị xxx tự tăng thêm 1 đơn vị và dùng chung cho cả giám đốc, lập trình viên cũng như tester. Mỗi công việc có mã công việc là 1 số nguyên tự tăng bắt đầu từ 100, tên việc, thời gian cần thiết để hoàn thành công việc đó trên lý thuyết. Bảng phân công công việc có mã bảng phân công là một số nguyên tự tăng bắt đầu từ 1000, nhân viên được phân công, công việc được phân công, thời gian bắt đầu thực hiện, deadline và kết quả thực hiện công việc. Bảng lương của từng nhân viên gồm mã bảng lương là một số nguyên tự tăng bắt đầu từ 10000, danh sách bảng phân công, nhân viên, tổng số công việc được giao, số công việc hoàn thành, số công việc không hoàn thành, tổng tiền phạt, lương thực lĩnh. Mô tả các thông tin trên vào các lớp tương ứng cho phù hợp.

- Cách tính lương của nhân viên: số ngày đi làm x (mức lương/22).
- Cách tính lương của giám đốc: như lương nhân viên cộng thêm thưởng theo quý bằng 80% tổng lương thực lĩnh 3 tháng trong quý.
- Cách tính lương lập trình viên: lương nhân viên + 30% (lương thực tế trong tháng x KPI)/100. Mỗi công việc được phân công hoàn thành sẽ được 2KPI.
- Cách tính lương tester: lương nhân viên + 20% lương tháng x (số testcase(công việc được phân công) đã hoàn thành trong tháng/100) + (50k x số lỗi tester đã phát hiện được).
- Mỗi công việc được phân công mà nhân viên không hoàn thành trừ 100k vào lương tổng.

Dữ liệu về nhân viên lưu trong file EMP.DAT. Dữ liệu về công việc lưu trong file WORK.DAT. Dữ liệu bảng phân công công việc được lưu trong file ASS.DAT. Dữ liệu về bảng lương lưu trong file PAY.DAT. Trong mỗi file có nhiều bản ghi. Mỗi bản ghi gồm nhiều trường dữ liệu. Mỗi trường dữ liệu ghi trên 1 dòng theo thứ tự các thuộc tính của lớp được mô tả ở trên. Load dữ liệu từ các file vào các danh sách tương ứng. Viết chương trình có menu chọn cho phép thực hiện các chức năng sau:

- 1) Thêm mới một giám đốc vào danh sách nhân viên.
- 2) Thêm mới một lập trình viên vào danh sách nhân viên.
- 3) Thêm mới một tester vào danh sách nhân viên.
- 4) Thêm mới một công việc vào danh sách công việc.
- 5) Thêm mới bảng phân công vào danh sách bảng phân công công việc. Với mỗi công việc và một nhân viên không xuất hiện quá 1 lần trong bảng này.
- 6) Hiển thị danh sách giám đốc ra màn hình dưới dạng bảng.
- 7) Hiển thị danh sách lập trình viên ra màn hình dưới dạng bảng.
- 8) Hiển thị danh sách tester ra màn hình dưới dạng bảng.
- 9) Hiển thị danh sách công việc ra màn hình dưới dạng bảng.
- 10) Hiển thị danh sách bảng phân công ra màn hình dưới dạng bảng.



- 11) Sắp xếp bảng phân công công việc theo tên nhân viên, nếu trùng tên sắp xếp theo họ tăng dần.
- 12) Sắp xếp bảng phân công theo deadline giảm dần.
- 13) Lập danh sách bảng lương cho nhân viên, hiển thị bảng lương ra màn hình.
- 14) Sắp xếp bảng lương theo lương thực lĩnh giảm dần, tên tăng dần, họ tăng dần.
- 15) Sắp xếp bảng lương theo tên nhân viên tăng dần, họ tăng dần.
- 16) Sắp xếp bảng lương theo tổng tiền phạt giảm dần, tên nhân viên tăng dần.
- 17) Liệt kê các nhân viên có mức lương thực lĩnh cao nhất trong tháng.
- 18) Liệt kê các nhân viên có mức lương thực lĩnh bằng x nhập vào từ bàn phím.
- 19) Lưu các danh sách vào file tương ứng. Các bản ghi phải duy nhất không trùng lặp.
- 20) Kết thúc chương trình.

**Bài 32.** Tài khoản thẻ ngân hàng gồm các thông tin: số tài khoản dạng VCBX trong đó x là số có 11 chữ số tự tăng từ  $10^{10}$ , tên chủ TK, ngày phát hành, ngày hết hạn, số dư, ngân hàng phát hành, số dư tối thiểu trung bình hàng tháng, phí duy trì thẻ, trạng thái hoạt động của tài khoản. Mỗi tài khoản có thể thực hiện các hành động kiểm tra số dư, chuyển tiền, nạp tiền vào tài khoản, rút tiền, gửi tiết kiệm, thanh toán hóa đơn, kích hoạt, khóa tài khoản, đổi mã PIN. Tài khoản thanh toán nội địa kế thừa tài khoản có thêm mức phí rút tiền nội ngân hàng, liên ngân hàng, có thể vay tiền dựa vào tổng số tiền đã giao dịch tính theo năm của TK loại này, hạn mức giao dịch tối đa trên 1 lần giao dịch. Tài khoản thanh toán quốc tế kế thừa tài khoản ngân hàng có thêm số định danh quốc tế của thẻ (8 chữ số), phí rút tiền nội ngân hàng, phí rút tiền liên ngân hàng và phí thanh toán quốc tế, hạn mức giao dịch tối đa trên một ngày. Cây ATM còn hoạt động chấp nhận tất cả các thẻ thanh toán và cho phép thực hiện các giao dịch theo yêu cầu của khách hàng chủ thẻ. Thông tin về cây ATM gồm mã số ATM, ngân hàng cung cấp dịch vụ, địa chỉ đặt ATM, tình trạng hoạt động. Mỗi lần giao dịch sẽ có thông tin giao dịch gồm mã giao dịch có 8 chữ số tự tăng từ  $10^7$ , mã ATM, số tài khoản, tên giao dịch, phí giao dịch, kết quả giao dịch, thời gian bắt đầu thực hiện giao dịch dạng dd/mm/yyyy HH:MM:SS ví dụ 25/03/2025 10:30:28.

- Thẻ bị khóa không thể thực hiện mọi giao dịch.
- Thẻ nội địa không thanh toán được hóa đơn và chuyển tiền quốc tế.
- Thẻ quốc tế sử dụng toàn cầu mọi giao dịch.
- Chi phí rút tiền tài khoản thẻ thanh toán nội địa: 1100đ/lần rút cùng ngân hàng và 3300đ/lần rút khác ngân hàng.
- Chi phí chuyển tiền, thanh toán hóa đơn, gửi tiết kiệm, kiểm tra số dư cả thẻ nội địa là 0đ.
- Chi phí rút tiền thẻ quốc tế: 1100đ nếu rút cùng ngân hàng, 9900đ nếu rút khác ngân hàng trong nước và 39900đ nếu rút ngân hàng nước ngoài.
- Chi phí thanh toán hóa đơn mua hàng quốc tế với thẻ quốc tế là 2.99\$ trong đó tỉ giá VND/USD lấy tại thời điểm thanh toán.
- Với thẻ thanh toán nội địa, chủ thẻ được vay tối đa 60% tổng số tiền giao dịch đã thực hiện trong 1 năm tính đến ngày vay.



- Khi rút tiền, chuyển tiền, thanh toán hóa đơn với cả hai loại thẻ bắt buộc tổng số dư khả dụng phải lớn hơn số tiền cần thanh toán 70,000đ và không vượt hạn mức giao dịch/lần hoặc trên 1 ngày.
- Nếu số dư tối thiểu trung bình hàng tháng > 2,000,000đ với thẻ nội địa và 5,000,000đ với thẻ quốc tế thì miễn phí rút tiền trong nước cả liên ngân hàng và nội ngân hàng phát hành thẻ.

Bạn được một ngân hàng kí hợp đồng phát triển phần mềm cho máy ATM của họ. Biết rằng dữ liệu thẻ nội địa lưu trong file DOMCARD.DAT. Dữ liệu thẻ quốc tế lưu trong file INTERCARD.DAT. Dữ liệu về máy ATM được lưu trong file ATM.DAT. Dữ liệu các lần giao dịch lưu trong file TRAN.DAT. Mỗi file có thể gồm nhiều bản ghi. Mỗi bản ghi có nhiều trường dữ liệu được ghi trên nhiều dòng theo thứ tự mô tả các thuộc tính trên. Yêu cầu nạp dữ liệu vào các danh sách tương ứng khi chương trình được kích hoạt. Hãy sử dụng kiến thức và kĩ năng của mình để viết phần mềm có menu chọn cho phép thực hiện các chức năng sau. Lưu ý thông báo kết quả rõ ràng khi thực hiện từng chức năng:

- 1) Tạo mới tài khoản thẻ thanh toán nội địa. Thêm vào danh sách tài khoản thẻ.
- 2) Tạo mới tài khoản thanh toán quốc tế. Thêm vào danh sách tài khoản thẻ.
- 3) Kiểm tra số dư theo số tài khoản.
- 4) Nạp tiền vào tài khoản theo số tài khoản.
- 5) Rút tiền nội ngân hàng theo số tài khoản.
- 6) Thanh toán hóa đơn nội địa.
- 7) Thanh toán hóa đơn quốc tế.
- 8) Chuyển tiền nội địa cùng ngân hàng.
- 9) Chuyển tiền nội địa khác ngân hàng.
- 10) Chuyển tiền quốc tế.
- 11) Kích hoạt tài khoản.
- 12) Đổi mã pin của tài khoản nào đó.
- 13) Liệt kê các giao dịch đã thực hiện trong một ngày.
- 14) In sao kê các lần giao dịch trong 1 tuần gần đây nhất tính đến thời điểm hiện tại.
- 15) Lưu các danh sách vào file tương ứng. Các bản ghi phải là duy nhất không trùng lặp.
- 16) Thoát chương trình.

**Trang chủ:** <https://braniumacademy.net/>

**Bài giải mẫu:** [click vào đây](#)