

## BÀI TẬP ÔN TẬP 2

**Bài 1:** Hãy cài đặt lớp `BigInteger` biểu diễn số nguyên lớn có những thuộc tính và phương thức cần thiết để đoạn mã nguồn ở trang sau biên dịch thành công và chạy đúng ngữ nghĩa:

```
void main()
{
    BigInteger n1("12545451515134567891011");
    BigInteger n2;
    cout << "Input an integer = ";
    cin >> n2;
    if ( n1 == n2 )
        cout << "Equal.\n";
    else
        cout << "Not equal.\n";
}
```

### Bài 3:

Cho một lớp `CThoiGian` gồm các thuộc tính sau:

`int gio, int phut, int giay`

- Hãy viết chương trình để hàm `main` chạy được đoạn code sau:

```
void main()
{
    CThoiGian t;
    cin>>t;
    cout<<t ;
    CThoiGian h(t) ;
    h--; //Giảm xuống một giây.
    cout<<h ;
    CThoiGian k(1,1,1) ;
}
```

# THE LAST BATTLE



Một đạo quân có  $N$  quân lính. Mỗi quân lính đều có hai khả năng là tấn công và phòng thủ, và có một số thông tin sau:

- **m\_Code (Mã số):** Kiểu chuỗi.
- **m\_Attack (Chỉ số tấn công):** kiểu số thực.
- **m\_Defend (Chỉ số phòng thủ):** kiểu số thực.
- **m\_Equipment (Trang bị):** Kiểu chuỗi (YES là có trang bị, NO là không có trang bị)
- **m\_Strength (Sức mạnh):** Kiểu số nguyên.

Trong đạo quân có các loại quân như sau:



**Valiant (Dũng sĩ):** thiên về tấn công, nếu được trang bị thêm khiên, khả năng phòng thủ sẽ tăng lên 2 lần.



**Archer (Cung thủ):** thiên về phòng thủ và hỗ trợ tấn công, nếu được trang bị thêm mũi tên lửa, khả năng tấn công sẽ tăng lên 1.5 lần.



**Knight (Hiệp sĩ):** dùng để đột phá hoặc tấn công chiến lược, nếu được trang bị thêm ngựa, khả năng tấn công và phòng thủ sẽ tăng lên 3 lần.

Ngoài ra, khả năng tấn công và phòng thủ cả đạo quân sẽ tăng lên hay giảm đi tùy thuộc vào hệ số chỉ huy của viên tướng cầm quân (**Army General**), thông tin của quân tướng chỉ huy tương tự như các loại quân khác nhưng có thêm chỉ số kinh nghiệm chỉ huy **m\_Experience** (kiểu số thực). Viên tướng cầm quân luôn luôn có trang bị, nhưng trang bị sẽ không ảnh hưởng đến khả năng tấn công hay phòng thủ. Mỗi đội quân chỉ có 1 viên tướng chỉ huy.



- **m\_Experience = 0**: khả năng tấn công và phòng thủ của đội quân sẽ giảm phân nửa.
- $0 < \mathbf{m\_Experience} \leq 2$ : khả năng tấn công và phòng thủ của đội quân sẽ giảm 1.2
- $2 < \mathbf{m\_Experience} \leq 5$ : khả năng tấn công và phòng thủ của đội quân sẽ tăng lên 1.5
- **m\_Experience > 5**: khả năng tấn công và phòng thủ của đội quân sẽ tăng lên 2.0

***Khi hai đạo quân giao tranh, cách thức chiến đấu như sau:***

- Quân lính đầu tiên của mỗi đội sẽ chiến đấu với nhau. Nếu bên nào thua quân lính đó sẽ bị xóa khỏi đội quân. Bên thắng khả năng tấn công và phòng thủ của quân lính đó sẽ giảm đi 20. Lần lược chiến đấu như vậy cho đến khi hết tất cả quân lính. Khi bên nào không còn quân lính nào thì viên tướng chỉ huy mới ra đánh trận.
- Một quân lính này chiến thắng quân lính kia khi và chỉ khi:
  - Ngăn được sức tấn công của đối phương (khả năng phòng thủ cao hơn khả năng tấn công của đối phương).
  - Phá được thế phòng thủ của đối phương (khả năng tấn công cao hơn hoặc bằng khả năng phòng thủ của đối phương).
  - Nếu 2 quân lính hòa nhau. Quân lính nào có chỉ số sức mạnh cao hơn sẽ giành chiến thắng.

Áp dụng phương pháp lập trình hướng đối tượng cho nhập vào tập tin INPUT (theo mẫu) và xuất ra tập tin OUTPUT (theo mẫu) là đội quân giành chiến thắng trong cuộc chiến và số quân còn lại với chỉ số sức mạnh còn lại.

- INPUT: Dòng đầu tiên là tên đội quân và số lượng quân lính trong đội quân đó.  
Các dòng tiếp theo là thông tin của từng quân lính.
- OUTPUT: Xuất ra thông tin đội quân chiến thắng (các chỉ số phải cập nhật).

INPUT . TXT	OUTPUT . TXT
Team A 4 AG AG1 70 70 YES 30 3.5 V V001 20 10 YES 15 A A001 10 5 NO 14 K K001 30 35 YES 20 Team B 3 AG AG001 100 100 YES 50 6 V V002 20 30 YES 20 K K001 50 50 YES 30	Team B Win AG AG001 100 100 YES 50 6 K K001 260 260 YES 30