**Giải thuật đề thi thử lần 2:**

**Bài 1:**

+ Nếu a chia hết cho b thì ta in a/b

+ Ngược lại thì ta ép kiểu a ra số thực rồi chia (/) cho b để kết quả ra số thập phân. Rồi ta nhân kết quả đó với 100. Lấy phần nguyên của kết quả đó. Để lấy phần nguyên thì ta cứ gán số thực cho 1 biến nguyên là ok. Nếu kết quả này chia hết cho 100 thì ta in ra kết quả/100 (phép chia lấy phần nguyên, vì đề bài yêu cầu nếu phần thập phân là 2 số 0 thì ta chỉ in phần nguyên). Ngược lại điều trên thì lúc này ta chia số đó cho 100 ra kết quả số thực. rồi dùng setprecision để in 2 phần thập phân

**Bài 2:**

Trong bài này, ta sử dụng thuật toán tham lam.

Ta sẽ duyệt lần lượt các kí tự trong các xâu từ trái qua phải. Gọi kí tự đang duyệt là s[i].

Nếu s[i] = ‘0’ thì :

* Nếu i ở vị trí cuối cùng của xâu, ta sẽ không làm gì nữa
* Ngược lại, nếu s[i + 1] = ‘1’ thì ta sẽ tăng vị trí i thành i = i + 2 vì xâu ‘01’ thõa mãn trong S
* Nếu s[i + 1] = ‘0’ thì ta sẽ tăng i = i + 1 vì xâu ‘0’ thỏa mãn trong S

Nếu s[i] = ‘1’ thì:

* Nếu i ở vị trí cuối cùng của xâu, ta tăng số lượng kí tự cần loại bỏ lên 1.
* Ngược lại, nếu s[i + 1] = ‘1’ thì ta sẽ tăng vị trí i thành i = i + 1 và tăng kết quả lên 1 ( vì xâu ‘1’không có trong S).
* Nếu s[i + 1] = ‘0’ thì ta sẽ tăng i = i + 2 vì xâu ‘10’ thỏa mãn trong S

Độ phức tạp O(N).

**Bài 3:**

* Ta sẽ chặt nhị phân số ngày tối thiểu để làm hết M cái nem, gọi sum là tổng các Ai, ta sẽ chặt từ 1 đến 2 \* (M / sum + 1) để tính toán không phải xử lí số lớn. M / sum + 1 là số ngày làm việc tối đa và nhân 2 để trường hợp xấu nhất là tất cả Bi = 1.
* Với giá trị mid đang chặt, ta duyệt qua toàn bộ N nhân viên, tính được số ngày nghỉ của nhân viên thứ i là mid / (Bi + 1) = X, từ đó số nem làm được trong mid ngày là Ai \* (mid – X), ta tính tổng số nem làm được nếu ≥ M thì trả về true và chặt xuống, ngược lại < M trả về false và chặt lên.
* Độ phức tạp: O(Nlog21015)

**Bài 4:**

Ta thấy N chia hết cho x và x + 1, mà x và x + 1 là 2 số nguyên tố cùng nhau

* N chia hết cho x\*(x+ 1).

Đặt N = k \* x \* (x+1).

* Xét trường hợp : k <= 106 . Ta chỉ cần sử dụng vòng for để tìm k, sau đó với mỗi giá trị của k sẽ tìm xem tồn tại x thỏa mãn hay không. Độ phức tạp O(106).
* Xét trường hợp : k > 106. Vì N <= 1018 nên ta suy ra x \* (x + 1) <= 1012.

⬄ x <= 106. Đến đây, ta chỉ cần sử dụng vòng for để tìm x xem có thỏa mãn không. Độ phức tạp O(106).

Vậy độ phức tạp của thuật toán là O(106).

**Bài 5: SHARE**

+ Để ăn được 1 điểm bài này cực kỳ đơn giản:

Vì đề cho n <= 3 và ai <= 100. Ta chỉ cần xét tất cả khả năng xảy ra (cho n vòng for chạy từ 0 đến 100) để tìm khả năng nào thoả mãn thì lấy. (Đại loại là duyệt tất cả các trường hợp). Độ phức tạp là O(100n)

+ Để ăn full điểm ta làm như sau:

Sort khả năng trả tiền tăng dần.

Nhận thấy nếu chú mèo trả được tiền thì chú mèo cũng trả được tiền do khả năng trả tăng dần. Vậy nên ta sẽ để các chú mèo trả tiền tăng dần.

Cách trả tiền tối ưu là dàn đều số tiền phải trả, vậy nên số tiền chú mèo cần trả là với là số tiền còn lại.

Để cài đặt bài này cho đơn giản ta dùng cấu trúc struct hoặc pair để lưu số tiền ban đầu của các chú mèo kèm theo vị trí ban đầu của các chú mèo. Để ta thuận tiện cho việc in kết quả. (vi ta đã sort làm thay đổi vị trí ban đầu).