



# Event Organizer

Link submit: <https://www.codechef.com/problems/MAXCOMP>

Solution:

C++	<a href="https://ideone.com/vSTREm">https://ideone.com/vSTREm</a>
Java	<a href="https://ideone.com/rNUexa">https://ideone.com/rNUexa</a>
Python	<a href="https://ideone.com/5qWuPh">https://ideone.com/5qWuPh</a>

Tóm tắt đề:

Đầu bếp Po đã đưa ra một quảng cáo trực tuyến nhằm cung cấp dịch vụ tổ chức sự kiện. Po nhận được một số lượng lớn phản hồi cho dịch vụ của anh ấy. Po nhận được nhiều đơn đặt sự kiện từ nhiều tổ chức. Po sẽ nhận được số tiền tương ứng với đơn đặt. Po nhận được  $N$  đơn đặt cho cuối tuần này. Cuối tuần gồm có 2 ngày, tất cả các sự kiện được tổ chức trong 48h. Đối với mỗi đơn có thời gian bắt đầu  $S$ , kết thúc là  $E$ , và số tiền nhận được tương ứng là  $C$ . Trong toàn bộ thời gian tổ chức một sự kiện thì không được tổ chức sự kiện khác. Điều đó có nghĩa không có 2 sự kiện diễn ra tại một thời điểm. Tuy nhiên, thời gian bắt đầu sự kiện này trùng với thời gian kết thúc sự kiện khác thì có thể tổ chức.

Hãy giúp đầu bếp Po sắp xếp để tổ chức các sự kiện sao cho thu được nhiều tiền nhất.

Input:

Dòng đầu chứa một số nguyên  $T$  ( $1 \leq T \leq 10$ ) là số lượng testcase. Mỗi test có dạng như sau:

- Dòng đầu gồm một số nguyên  $N$  ( $1 \leq N \leq 2000$ ) là số đơn đặt mà Po nhận được.
- $N$  dòng tiếp theo chứa ba số nguyên  $S_i$   $E_i$   $C_i$  ( $0 \leq S_i < E_i \leq 48$ ,  $C_i \leq 10^6$ ), tương ứng thời gian bắt đầu, kết thúc và số tiền.

Output:

Đối với mỗi testcase in ra số nguyên duy nhất, là số tiền nhiều nhất mà Po có thể nhận

Ví dụ:

2	3700
4	2000
1 2 100	
2 3 200	
3 4 1600	
1 3 2100	

3	
1 10 2000	
2 5 100	
6 9 400	

### Giải thích ví dụ:

Ví dụ gồm hai bộ test.

**Bộ 1:** chọn tổ chức sự kiện 3 và 4 sẽ thu được lợi nhuận lớn nhất và thời gian của chúng không chồng chéo lên nhau.

**Bộ 2:** các chọn tốt nhất là chỉ thực hiện sự kiện 1 với lợi nhuận là 2000.

### Hướng dẫn giải:

Dùng giải thuật Floyd. Đầu tiên phải lập ma trận kề  $dist$ ,  $dist[i][j]$  với  $i \leq j$  tương ứng với số tiền nhận được khi tổ chức sự kiện từ giờ thứ  $i$  tới giờ thứ  $j$ .

Lúc lập ma trận, nếu có nhiều sự kiện cùng tổ chức vào cùng một khoảng thời gian  $i \rightarrow j$  thì chọn sự kiện có số tiền thu được là nhiều nhất.

Dùng giải thuật Floyd để tìm số tiền thu được lớn nhất với mọi khoảng thời gian  $i$  đến  $j$ .

Kết quả là số tiền thu được khi tổ chức sự kiện trong khoảng thời gian từ 0 đến 48 chính là  $dist[0][48]$ .

**Độ phức tạp:**  $O(T * N^3)$  với  $T$  là số lượng bộ test,  $N$  là số lượng thời gian khác nhau (tức  $N = 49$  giờ khác nhau từ 0 đến 48).