A hotel needs a program to assist any tourist who wants to visit N different places ($N \leq 20$). Every day the tourist leaves the hotel, visits M places (may be fewer on the last day) and comes back to the hotel.

Let us consider that the hotel is identified by 0, the places to visit by consecutive positive integers and the available roads by triples (i, j, d), where d represents the road length (in km) and $0 \leq i < j \leq N$.

Input

The input file includes data about several tourists. For each tourist the available roads (at least 2) and a negative value ($v < -1$) which designates the number of visits to be scheduled for one day ($M = -v$) are given, as in the sample input below.

The program must schedule the visits according to the following constraints:

- each place is visited only once;
- a place must be visited when reached for the first time (afterwards it can be included in any tour as a passing by point);
- the total travel must be as short as possible;
- ties are solved by applying the 'shorter one day tour' rule, from the last to the first day ([... 180 km 50 km] is preferred to [... 200 km 50 km]);
- if the tie is not solved by the above rule, the 'preferred place first' rule is applied (the schedule 1 3 5 ... is preferred to 1 3 7 ...).

Output

The resulting schedules are stored in the output file in the following format:

```
== Tourist  i  -- M visits a day --
 Day  1: [dfh] - fvp - [dpp] - ..... - lvp - [dth]
    . . . . . .
===
```

where, for each day tour

- dfh - distance from hotel to the first visited place
- fvp - index of the first visited place
- dpp - distance between consecutive visited places
- lvp - index of the last visited place
- dfh - distance from last visited place to the hotel

Sample Input

```
0 1 10
0 2 10
0 3 10
0 4 10
1 2 10
3 4 15
4 5 10
-3
0 1 2
1 2 2
2 3 2
3 1 2
0 4 2
0 5 2
-2
```

Sample Output

```
== Tourist  1  --  3 visits a day --
 Day  1: [10] - 3 - [15] - 4 - [10] - 5 - [20]
 Day  2: [10] - 1 - [10] - 2 - [10]
===
== Tourist  2  --  2 visits a day --
 Day  1: [2] - 1 - [4] - 4 - [2]
 Day  2: [4] - 2 - [2] - 3 - [4]
 Day  3: [2] - 5 - [2]
===
```