The funny stone game is coming. There are n piles of stones, numbered with $0, 1, 2, ..., n-1$. Two persons pick stones in turn. In every turn, each person selects three piles of stones numbered i, j, k ($i < j$, $j \le k$ and at least one stone left in pile i). Then, the person gets one stone out of pile i, and put one stone into pile j and pile k respectively. (Note: if $j = k$, it will be the same as putting two stones into pile j). One will fail if he can't pick stones according to the rule.

David is the player who first picks stones and he hopes to win the game. Can you write a program to help him?

The number of piles, n, does not exceed 23. The number of stones in each pile does not exceed 1000. Suppose the opponent player is very smart and he will follow the optimized strategy to pick stones.

Input

Input contains several cases. Each case has two lines. The first line contains a positive integer n ($1 \le n \le 23$) indicating the number of piles of stones. The second line contains n non-negative integers separated by blanks, $S_0, \ldots, S_{n-1}$ ($0 \le S_i \le 1000$), indicating the number of stones in pile 0 to pile $n-1$ respectively.

The last case is followed by a line containing a zero.

Output

For each case, output a line in the format '`Game` t`:` i j k'. t is the case number. i, j and k indicates which three piles David shall select at the first step if he wants to win. If there are multiple groups of i, j and k, output the group with the minimized lexicographic order. If there are no strategies to win the game, i, j and k are equal to '`-1`'.

Sample Input

```
4
1 0 1 100
3
1 0 5
2
2 1
0
```

Sample Output

```
Game 1: 0 2 3
Game 2: 0 1 1
Game 3: -1 -1 -1
```