Abstract. The May 2011 ACM Programming Contest. Expect all inputs from standard in. All expected outputs go to standard out. Do not print any additional prompts.

1 Double Double Factorial?

Alice and Bob are arguing about factorials and double factorials. Bob swears that since double the double factorial of 2 is more than 2 factorial, then double the double factorial of any number must be greater than the factorial of that number. Alice disagrees, and she wants to give him several counter-examples to prove him wrong.

Most people versed in mathmatics are familiar with or factiorial, or !, operator. Where:

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1 \tag{1}$$

with the aditional definition that $0! = 1$.

The double factorial operator, or !!, is very similar:

$$n!! = \begin{cases} n > 0 \text{ odd} & : n \cdot (n-2) \dots 5 \cdot 3 \cdot 1 \\ n > 0 \text{ even} & : n \cdot (n-2) \dots 6 \cdot 4 \cdot 2 \\ n = -1, 0 & : 1 \end{cases} \tag{2}$$

1.1 Input

Except as input, a single integer, X.

Constraint: $X > 2$

1.2 Output

Expected output is the difference between factorial of X and double of the double factorial of X, that is:

$$Output = X! - 2 \cdot X!! \tag{3}$$

1.3 Sample Input

5

1.4 Sample Output

90

2 Fibo and Tetra

Fibo likes playing with numbers, and he came up with a cool series that grows quickly, but not too quickly. His younger brother Tetra, always jealous of his older brother, wants to out do him by creating a more complicated series, that grows faster, but not too much faster.

Fibo defines his series as follows:

$$F_n = F_{n-1} + F_{n-2}, \text{ where } F_0 = 0, \text{ and } F_1 = 1. \tag{4}$$

Tetra's more complicated series is defined as follows:

$$T_n = T_{n-1} + T_{n-2} + T_{n-3} + T_{n-4}, \text{ where } T_0 = 0, T_1 = 1, T_2 = 1, \text{ and } T_3 = 2. \tag{5}$$

2.1 Input

Expect as input, a single integer, X.

Constraint: $X \geq 4$

2.2 Output

Expected output is the difference between T_X and F_X that is:

$$Output = T_X - F_X \tag{6}$$

2.3 Sample Input

4

2.4 Sample Output

1

3 The Cape Wrecker

A villian known as The Cape Wrecker has been wrecking capes all over Capetown and the authorities think they figured out a way to track him down. They leave a crate of capes on the coast, and depending on the count of the capes in the crate of capes on the coast the cops can calculate how long it will be until the The Cape Wrecker arrives.

The calculation is as follows:

Reorder the digits of a number in ascending and descending order. Take the difference of these two new numbers. Contuing this will lead to one of two phenomenon. Either it will converge to the number of seconds until the Cape Wrecker appears (a Cape Wrecker constant) or it will cycle infinitely (the Cape Wrecker is onto your ruse and will not show).

3.1 Input

Expect as input, a single integer X.

3.2 Output

Expected output is either the Cape Wrecker constant for X, or if X causes a cyle, -1.

3.3 Sample Input

18

3.4 Sample Output

0

4 Earl and Bernie

Earl and Bernie have been roomates since they were children (no explanation is given for the lack of adult supervision). They've decided to go their seperate ways, but they realize that they don't know what belongs to whom. They have decided the fairest thing is to come up with a partitioning of objects such that each of them end up with objects that sum to equal value. If there exists no partitioning such that they each end up with an equal sum value of objects, they will postpone their seperation until they have more stuff.

4.1 Input

Expect as input, an integer X, followed by X integers.

4.2 Output

Expected output is the string, ”YES” if there exists two partitions of equal value, or ”NO” if there does not.

4.3 Sample Input 1

6
2
2
2
2
2
2

4

4.4 Sample Output 1

YES

4.5 Sample Input 2

4
2
3
4
6

4.6 Sample Output 2

NO

5 Knockout Tournament

In a knockout tournament there are 2n players. One loss and a player is out of the tournament. Winners then play each other with the new winners advancing until there is only one winner left. If we number the players 1, 2, 3, . . . , 2n, with the first round pairings $2k - 1$ vs $2k$, for $k = 1, 2, \ldots, 2n - 1$, then we could give the results of the tournament in a complete binary tree. The winners are indicated in the interior nodes of the tree. See Fig. 1 as an example of a tournament with $n = 3$.

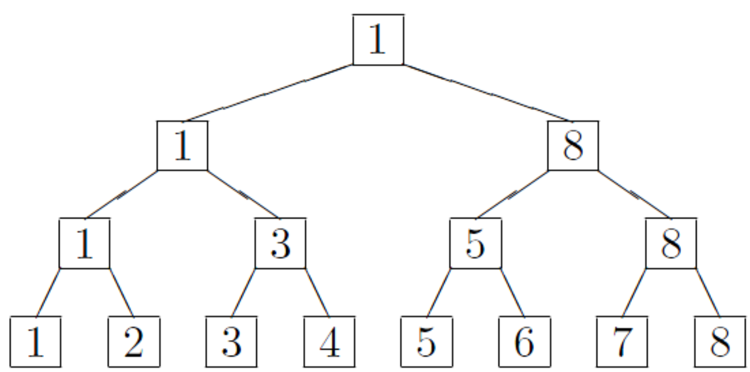

Fig. 1. The Luminosity based metric plots for each frame for 1,2,4, and 64 mesh updates per frame using the least recently updated update metric.

After the tournament, some reporters were arguing about the relative ranking of the players, as determined by the tournament results. Its assumed that if player A beats player B who in turn beats player C, that player A will also beat player C; that is, winning is transitive. Now there is no doubt who the best player is. The question is what is the highest ranking a player can reasonably claim as a result of the tournament and what is the worst ranking a player can have, as a result of the tournament? For example, in the above tournament player 2, having lost to the eventual winner, could claim to be the 2nd best player in the field, but could well be the worst (ranked 8th). Player 5 could claim to be as high as 3rd (having lost to someone who could be 2nd) but no worse than 7th (having beaten one player in the 1st round).

You are to determine the highest and lowest possible rankings of a set of players in the field, given the results of the tournament.

5.1 Input

There will be multiple input instances. The input for each instance consists of three lines. The first line will contain a positive integer $n < 8$, indicating there are 2n players in the tournament, numbered 1 through 2n, paired in the manner indicated above. A value of n = 0 indicates end of input. The next line will contain the results of each round of the tournament (listed left-to-right) starting with the 1st round. For example, the tournament above would be given by:

```
1 3 5 8 1 8 1
```

The final line of input for each instance will be a positive integer m followed by integers k1, ..., km, where each ki is a player in the field.

5.2 Output

For each ki, issue one line of output of the form: Player ki can be ranked as high as h or as low as l. where you supply the appropriate numbers. These lines should appear in the same order as the ki did in the input. Output for problem instances should be separated with a blank line.

5.3 Sample Input

```
3
1 3 5 8 1 8 1
2 2 5
4
2 3 6 7 9 11 14 15 3 6 9 15 6 9 6
4 1 15 7 6
0
```

5.4 Sample Output

```
Player 2 can be ranked as high as 2 or as low as 8.
Player 5 can be ranked as high as 3 or as low as 7.
```

Player 1 can be ranked as high as 4 or as low as 16.
Player 15 can be ranked as high as 3 or as low as 13.
Player 7 can be ranked as high as 2 or as low as 15.
Player 6 can be ranked as high as 1 or as low as 1.

6 One Person "The Price is Right"

In the game show The Price is Right, a number of players (typically 4) compete to get on stage by guessing the price of an item. The winner is the person whose guess is the closest one not exceeding the actual price. Because of the popularity of the one-person game show Who Wants to be a Millionaire, the American Contest Management (ACM) would like to introduce a one-person version of the The Price is Right. In this version, each contestant is allowed G $(1 \leq G \leq 30)$ guesses and L $(0 \leq L \leq 30)$ lifelines. The contestant makes a number of guesses for the actual price. After each guess, the contestant is told whether it is correct, too low, or too high. If the guess is correct, the contestant wins. Otherwise, he uses up a guess. Additionally, if his guess is too high, a lifeline is also lost. The contestant loses when all his guesses are used up or if his guess is too high and he has no lifelines left. All prices are positive integers.

It turns out that for a particular pair of values for G and L, it is possible to obtain a guessing strategy such that if the price is between 1 and N (inclusive) for some N, then the player can guarantee a win. The ACM does not want every contestant to win, so it must ensure that the actual price exceeds N. At the same time, it does not want the game to be too difficult or there will not be enough winners to attract audience. Thus, it wishes to adjust the values of G and L depending on the actual price. To help them decide the correct values of G and L, the ACM has asked you to solve the following problem. Given G and L, what is the largest value of N such that there is a strategy to win as long as the price is between 1 and N (inclusive)?

6.1 Input

The input consists of a number of cases. Each case is specified by one line containing two integers G and L, separated by one space. The end of input is specified by a line in which G = L = 0.

6.2 Output

For each case, print a line of the form:

Case c: N

where c is the case number (starting from 1) and N is the number computed.

6.3 Sample Input

3 0
3 1
10 5
7 7
0 0

6.4 Sample Output

Case 1: 3
Case 2: 6
Case 3: 847
Case 4: 127