1 A simple calculation

Task: You are given $k, n \in \mathbb{N}$, where $0 \leq k \leq 15$ and $1 \leq n \leq 145$. Compute the value of $\sum_{j=1}^{n} j \cdot k^j$.

Input: Each line specifies a test case. It consists of two integers, the first is n, the second is k.

Output: In the i-th line print the result of the i-th test case.

Sample Input:

```
3  5
10  4
```

Sample Output:

```
430
13514980
```

2 New Numbers

Task: For an integer $1 \leq x \leq 2^{31} - 1$, let $x[j, i] = (x_j, \ldots, x_i)$ denote the bits j to i of the binary representation of x, where the least significant bit has index 0. For example, if $x = 11$, then $x[3, 0] = (1, 0, 1, 1)$ and $x[2, 0] = (0, 1, 1)$. Your task is to find two integers a and b such that the following properties hold:

- If $x[i, i] = 0$, then so are $a[i, i]$ and $b[i, i]$.
- If $x[i, i]$ is 1, then so is either $a[i, i]$ or $b[i, i]$.
- For every range $i, \ldots, j$, the part $a[j, i]$ contains at most one "1" more and at most one "1" less than $b[j, i]$.
- a is larger than b if and only if $x[30, 0]$ contains an odd number of "1"s.

Input: The input contains several test cases. Each test case consists of a single line containing an integer $x \in [1 : 2^{31} - 1]$. The input is terminated by a single line containing a zero.

Output: For each test case output in a separate line the numbers a and b.

Sample Input:

```
57399
123
0
```

Sample Output:

```
16421  40978
41  82
```

3 Attractiveness

Task: You are developer of an online role-playing game. In this game each player generates a character choosing six (not necessarily different) attributes from the set $\{a, \ldots, z\}^3$. The *attractiveness* of a set of six attributes is given by the number of players who choose this set. Furthermore, we say that a set of six attributes is most attractive if there is no different set of six attributes that is chosen by a larger number of players. You are interested in the number of players who chooe a set of six attributes that is most attractive.

Input: The input consists of several test cases and is terminated by s single line containing a zero. The first line of a test case contains the number of players m, where $m \in [1 : 15000]$. Then, for each player a single line specifies the chosen six attributes.

Output: For each test case output the number of players who chose a set of six attributes that is most attractive.

Sample Input:

```
3
abc abd abe abf abg abh
xyz xxx zxy zxy uvw wuv
abd abe abf abg abh abc
3
abc def ghi jkl mno pqr
bcd efg hij klm nop qrs
cde fgh ijk lmn opq rst
0
```

Sample Output:

```
2
3
```

4 Floating median

Task: You are given online a sequence of non-negative integers $(a_i)_{i\in[1:n]}$. Compute for each $k \in [1 : n]$ the median of the subset $\{a_i : i \in [1 : k]\}$. The median m of a set $\{a_1, \ldots, a_k\}$ is defined as follows: Let $\tilde{a}_1 \leq \ldots \leq \tilde{a}_k$ denote the numbers $(a_i)_{i\in[k]}$ in non-decreasing order. Then, the median is defined as follows

$$m(\{a_1, \ldots, a_k\}) = \begin{cases} \tilde{a}_{(k+1)/2}, & \text{if } k \text{ is odd,} \\ \lfloor(\tilde{a}_{k/2} + \tilde{a}_{k/2+1})/2\rfloor, & \text{otherwise.} \end{cases}$$

Note that we simulated integer division in the previous line.

Input: The input consists of at most 11000 integers. a_1 is given in line 1, and so on.

Output: For line k, print the median of the set $\{a_1, \ldots, a_k\}$.

Sample Input:

```
1
2
3
5
8
```

Sample Output:

```
1
1
2
2
3
```

5 Battle

Task: In the stone age two hostile tribes are fighting each other. Such fights follow strict rules: There is a fix number a of arenas. In each round the best a warriors of each tribe fight against each other in the arenas. (The best fighters fight in arena 1, the second best in arena 2, and so on.)

Each warrior has a strength s, which is an integer between 1 and 100. The result of a fight is rather predictable: Assume fighter F_a has strength s_a and fighter F_b has strength s_b. W.l.o.g. assume that $s_a \geq s_b$. Then, after the fight, fighter F_b died, and fighter F_a has strength $s_a - s_b$. In case that both warriors have equal strength, both die.

Afterwards, the surviving fighters join their tribes, and the next round starts. There are new rounds, until all warriors of one of the tribes are dead. In case, that a tribe has only m fighters, where $m < a$, only the best m warriors of each tribe fight against each other.

Which tribe will survive the fight?

Input: The first line contains the number t of test cases. Afterwards we specify the test cases. A test case starts with one line containing the number of arenas a, the number of warriors w_1 of the first tribe, and the number of warriors w_2 of the second tribe. The following w_1 lines contain the strength of each warrior of the first tribe, and then there are w_2 lines, which contain the strengths of the second tribe. (There is no guarantee that the strengths are given in an order!) We have $1 \leq t \leq 100$, and $1 \leq w_1, w_2 \leq 50000$.

Output: For each test print: In case that both tribes die, print "Both tribes died". Otherwise, if the first tribe survives, print "The first tribe survived". Then, print the strength of the surviving warriors in descending order. If the second tribe survives, print "The second tribe survived" and again print the the strength of the surviving warriors in descending order.

Sample Input:

```
2
3 1 1
97
97
2 2 4
5
5
25
5
5
25
```

Sample Output:

```
Both tribes died
The second tribe survived
20
20
5
5
```

6 Contest Scores

Task: The team ranking on this server is computed in the following way. Teams are first ranked by the number of correctly solved problems (in descending order). Ties are broken by the total time needed for the correctly solved (!) problems. If two teams are tied in both these parameters, they are ordered by increasing team numbers.

A team has correctly solved a problem if any of the submissions for that problem was judged as correct. The time needed for this problem is the time at which the first correct solution was submitted plus 20 minutes for every incorrect submission received before the first correct solution.

Input: The input consists of a single test case. There are 50 teams and 20 problems, both numbered $1, 2, \ldots$. Each line of the input except the last one (which is empty) describes one submission to the server. It contains the team number, the problem number, the submission time, and an additional character; this character can be "C" (for a correct submission), "I" (for an incorrect submission), or "R" (for a clarification request). Clarification requests do not affect the time score of a team. You can assume that the input lines are sorted by increasing timestamps.

Output: The output is the scoreboard sorted by the above-mentioned criteria. Only teams that made a submission shall be printed. Each line of the ouput

contains the team number, the number of solved problems, and the total time needed for the solved problems.

Sample Input:

```
2  4  0  I
5  4  5  C
2  4  7  C
2  5  9  R
2  5  10  C
```

Sample Output:

```
2  2  37
5  1  5
```

7 An online graph problem

Task: You are given an undirected graph on the set of vertices $\{1, \ldots, n\}$. In the beginning, the set of edges E is empty. Then, over the time, events of the following two types occur: A new edge is added to the graph, or you have to answer a query if there exists a path between two vertices. We say that a query is *successful* if there exists a path between the corresponding two vertices, otherwise we say it is *not successful*. You have to compute the number of successful and not successful queries.

Input: The first line contains the numbers of test cases. A test case is specified as follows: The first line contains the number n of vertices and the number of events. Then the events arrive online. The event that an edge is added to the graph is given by a line containing the character "n" and then two integers in $[1 : n]$ that describe the endpoints of the edge. The query event is given by the character "q" and then two integers in $[1 : n]$ that give you the vertices you are interested in.

Output: For each test case print in a single line the number of successful and then the number of not successful queries.

Sample Input:

```
2
4  6
q  1  2
n  1  2
q  2  1
n  1  3
n  2  4
q  3  4
2  2
```

q 1 1
q 1 2

Sample Output:

2 1
1 1

8 Most Frequent Value

Task: Let $a_1, \ldots, a_n$ be a sequence of non-decreasing integers. A query consists of two indices i and j, where $1 \leq i \leq j \leq n$. The task is to compute the number of occurrences of the most frequent value in the subsequence $a_i, \ldots, a_j$.

Input: The first line of every test case contains two integers $n \leq 100000$ and $q \leq 100000$. The next line contains n integers $a_1, \ldots, a_n$ in non-decreasing order separated by spaces. The following q lines each contain one query, i.e., two integers i and j with $1 \leq i \leq j \leq n$.

The last test case is followed by a line containing "0".

Output: For each query print the number of occurrences of the most frequent value in the subsequence.

Sample Input:

12 3
−7 −6 −5 −5 0 0 0 0 2 9 9 9
4 5
3 12
7 12
0

Sample Output:

1
4
3