1 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
```

2 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
```

3 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
```

4 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
```

5 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
```

6 Maximal Independent Set

Task: You are given an undirected graph $G = (V, E)$. An *independent set* is a subset S of the vertices V, such that no two vertices u, v in V are adjacent, i.e., $\{u, v\} \notin E$. A maximal independent set is an independent set S' such that $|S'| \geq |S|$ for all independent sets S.

You have to find the size of a maximal independent set.

Input: The first line contains the number of test cases t. A test case is specified as follows: The first line contains the number of vertices $|V|$ and the number of edges $|E|$. You can assume that $|V| \leq 100$. The following $|V|$ lines describe the edges: Each line contains two integers $a, b \in \{1, \ldots, 100\}$ and this means that $\{a, b\} \in E$.

Output: Output the size of a maximal independent set.

Sample Input:

```
2
2 1
1 2
4 5
1 2
2 3
4 3
2 4
4 1
```

Sample Output:

1
2

7 2C-Coin

Task: The age of Bitcoin has already started. So your task is to reform the classical monetary system of the Euro, because it is a bit outdated. The idea is as follows: A new monetary system consists of k coins, where the i-th coin is composed of Bitcoins and Euros, i.e., the i-th coin is given by a tuple (b_i, e_i), where b_i denotes the number of Bitcoins and e_i denotes the number of Euros of the i-th coin. Since it is composed of two components we call this new monetary system the 2C-system. Assume you are given some coins, where m_i denotes the number of coins of type i you possess.

Then, the *2C-value* V is given as follows:

$$V = \sqrt{\sum_{i=1}^{k} m_i \cdot (b_i^2 + e_i^2)}.$$

The question is the following: Given a 2C-system, consisting of several coins, and a value V, what is the minimum number of coins you need to obtain a 2C-value of V?

Input: The first line contains the number of test cases t. A test case is specified as follows: The first line contains k and V (both of them integers). You can assume that $k \in [1 : 35]$ and $V \in [1 : 280]$. In the following k lines the coins are described: The first number denotes b_i and the second number denotes e_i. You can assume that b_i, e_i are integers in $[0 : 10]$.

Output: Output the minimum number of coins you need to obtain a 2C-value of V. If it is not possible to obtain a value of V print “impossible”.

Sample Input:

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

Sample Output:

1
5
impossible

8 Looking for oil

Task: You are looking for oil in a Native American reservation. In order to do this, you subdivide their quadratic land into $n \times n$ quadratic parcels. For each parcel you estimate the difference between buying it and the selling value of the pumped up oil (the profit).

Unfortunately, due to old conventions, the Native Americans are only willing to sell you a region of their land that forms an arbitrary rectangle in the $n \times n$ parcels.

What is the maximum profit you can achieve? Be fast, other companies are also interested.

Input: There will be only a single test case specified. The first line contains n, where you can assume that $1 \leq n \leq 300$. Then, the next line contains the individual profits of each parcel, i.e., there are n^2 many integer values with values in $\{-100, \ldots, 100\}$.

We go down from north to south and from west to east, i.e., the first n numbers describe the profits in the parcels, which are furthest in the north, from west to east.

Output: Output the maximum profit you could obtain.

Sample Input:

3
-1 4 -1 -2 1 2 -2 -1 -3

Sample Output:

6