The German Collegiate Programming Contest 2017 (Practice Session)

GCPC 2017

Problems

A Relatively great day
B Catch them all!
C Advertising

Do not open before the contest has started.

This page is intentionally left (almost) blank.

Problem A
Relatively great day

It is a good day. Gregor wakes up full of energy, gets out of bed and fixes himself his favorite breakfast: **G**reen Tea, **C**arrots, **P**ears, and **C**ookies together with some toast. Afterwards, he relaxes a bit and then heads out to meet two of his friends, Paul and Jhonny.

Jhonny notices immediately how Gregor is in a good mood and asks him about it. Gregor tells him, that today he just feels so full of energy. After a while Paul remembers that there is even a way to tell just how much energy Gregor has today — using the very well-known formula $E = mc^2$.

Can you tell the trio how much energy Gregor contains?

Input

The input consists of an integer m ($1 \leq m \leq 100$), the mass of Gregor.

Output

Output a single line containing the energy equivalent of Gregor. You may assume that the input is given in the same order of magnitude as c ($c = 299792458$), the speed of light. In particular, this means that you may disregard all units and need not to convert the input.

Sample Input 1

```
5
```

Sample Output 1

```
449377589368408820
```

Sample Input 2

```
85
```

Sample Output 2

```
7639419019262949940
```

This page is intentionally left (almost) blank.

Problem B
Catch them all!

One of Jhonny's favorite things to do on the weekend is to get out his old console, the GameGirl, and play his favorite game: Mainzelmon! The goal of the game is to catch little monsters, so-called Mainzelmons, and train them by letting them fight against the ones from other players. If a Mainzelmon reaches a certain number of experience points, it turns into another Mainzelmon which is stronger and looks slightly different. Jhonny was already able to catch the cute but rather weak Mainzelmon FAUbsi. After some intense fights against his enemy player, team RocKIT, his FAUbsi turned into a FAUboga, and eventually even became a FAUboss. One thing that's bugging Jhonny is that he has never managed to catch the strongest and most seldom Mainzelmon of the game: MewTUM!

Recently his friend Paul told him that MewTUM can only be caught if one owns at least one Mainzelmon of every available Mainzelmon type in the game. Unfortunately, Jhonny has lost track of which Mainzelmon types he already has. Luckily, the game provides an option to print out a list of all the Mainzelmons that a player has caught so far. However, this list is not sorted and also contains some Mainzelmon types more than once, as Jhonny has caught Mainzelmons of his favorite types multiple times. Since Jhonny is a particularly lousy programmer, he asks you to write a program which outputs a sorted list containing each type of Mainzelmon he owns only once. Further, he would like to compute some values for statistical purposes. The exact formats of the input and output lists are described below.

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 10^4$) which is the total number of Mainzelmons that Jhonny owns.
- The following n lines describe Jhonny's Mainzelmons further in detail. Each line consists of an integer x_i, a string s_i, an integer k_i, and a double c_i, where
  - x_i ($1 \leq x_i \leq 10^4$) is the ID of the i-th Mainzelmon,
  - s_i ($1 \leq |s| \leq 20$) is the type of the i-th Mainzelmon,
  - k_i ($1 \leq k \leq 10^9$ is the number of experience points of the i-th Mainzelmon)
  - c_i ($1 \leq c_i \leq 10^3$) is monetary value of the i-th Mainzelmon.

Each Mainzelmon type is identified with a distinct ID, and Mainzelmons of the same type share the same ID. The Mainzelmon types solely consist of small and capitalized letters of the English alphabet.

Output

The output consists of:

- One line with an integer m, an integer ℓ, and a double d, where
  - m is the number of distinct Mainzelmon types that Jhonny owns.
  - ℓ is the sum of experience points of all Mainzelmons Jhonny owns.

  - d is the average monetary value of all Mainzelmons Jhonny owns. The average should be correct up to an absolute or relative error of at most 10^{-2}.
- m lines each containing an integer x_i and a string s_i, where
  - x_i is the ID of the i-th Mainzelmon type, and
  - s_i is the name of the corresponding type.

  The m lines should be sorted in ascending order by the ID.

Sample Input 1

```
3
16 FAUbsi 110 10.00
25 Peecrapchew 20 15.00
16 FAUbsi 70 5.11
```

Sample Output 1

```
2 200 10.03
16 FAUbsi
25 Peecrapchew
```

Sample Input 2

```
4
18 FAUboss 200 564.86
9 TUMtok 350 653.72
17 FAUboga 150 58.59
10 FAUpy 30 1.01
```

Sample Output 2

```
4 730 319.54
9 TUMtok
10 FAUpy
17 FAUboga
18 FAUboss
```

Problem C
Advertising

Gregor was appointed to be the marketing manager of the upcoming GCPC. The problem is that Gregor has absolutely no experience in making advertisement for such an event and, unfortunately, he is also very little talented. Desperately he remembers that big companies used to make advertisement by putting up posters - so he simply does the same. But when he asked people if they were coming to the GCPC, most of them did not even remember seeing the posters. So in the end he resorted to the best ad strategy he knew of - he just told everybody.

However, advertising is fickle — to be actually effective for somebody highly depends on who told the respective person about the event. If your best friend told you to participate with him in the GCPC, you would be a lot more likely to follow his advice than if some stranger told you on the way from one lecture to the next.

To measure this, Gregor assigns each person a "participation likelihood" (from 0 to 10), measuring how likely it is that the person participates in the GCPC. Naturally, the "participation likelihood" of a person is the maximum "friendship value" among the people that told him to participate. Friendship values range from 0 ("Don't even know who this is.") to 10 ("He/She is my favorite person to bake cookies with."), but do not have to be symmetric. You can assume that as long as their own participation likelihood is greater than 0, everyone will tell all people they know to participate as well. However, if the participation likelihood of a person is 0, then he does not tell anybody else about the event. As Gregor started the campaign, his participation likelihood is already 10.

Can you help Gregor predict the total influence his advertising campaign will have, i.e. the sum of participation likelihoods of all people?

Input

The input starts with a line containing n ($1 \leq n \leq 400$), the number of people Gregor knows. n lines follow, with the i-th line containing n integers denoting the friendship values values $friendship_{i,1}, \ldots, friendship_{i,n}$ where $friendship_{i,j}$ ($0 \leq friendship_{i,j} \leq 10$, $friendship_{i,i} = 0$) is the friendship value person i assigns to person j. Gregor is person 1.

Output

Output the sum of all participation likelihood values of all people.

Sample Input 1

```
4
0 0 0 1
5 0 0 0
1 5 0 5
1 10 5 0
```

Sample Output 1

```
30
```

Sample Input 2

```
5
0 0 1 6 0
0 0 10 9 0
0 9 0 0 0
0 2 0 0 0
4 0 0 0 0
```

Sample Output 2

```
14
```