1 Less than, greater than or equal to

Task: You are given pairs of numbers. Check, which relationship holds.

Input: The first line specifies the number of test cases t, where $t \leq 100$. The following t lines contain two integers x and y, where $|x|, |y| < 1000$.

Output: Print "less than" if $x < y$, "greater than" if $x > y$, or "equal to" if $x = y$.

Sample Input:

```
3
1 1
-1 1
1 -1
```

Sample Output:

```
equal to
less than
greater than
```

2 Chess

Task: We define the *independence problem* as follows: Given a chess board of size $m \times n$ and a certain chess piece (king, queen, rook or knight), how many pieces of a kind can you place on the board so that the pieces do not attack each other? We abbreviate king with 'K', queen with 'Q', rook with 'r', and knight with 'k'.

Input: The first line specifies the number of test cases $t \leq 100$. The following t lines describe the test cases. The first character specifies the piece, the following two integers m, n the size of the board. You can assume, that $4 \leq m, n \leq 100$.

Output: Print the maximum number of non attacking pieces for each test case.

Sample Input:

```
2
k 6 7
Q 8 10
```

Sample Output:

```
21
8
```

3 Company excursion

Task: You have to organize the next company excursion. You are given the number of participants, some possible dates and a limit of money you are allowed to spend. Now you have to find an accommodation, which meets the requirements, i.e., there must be sufficiently many free rooms and they should not be too expensive.

Input: There are several concatenated test cases. The first line of a test case is specified as follows. There are four integers: The number n of possible participants, your limit ℓ, the number a of possible accommodations, and the number d of possible dates. Then, for each accommodation there are two lines: The first line states the price for one person, and the next line the number of free rooms at the dates.

Output: For each test case print the minimum cost for the company excursion, and print "impossible" if there is no date which meets all the requirements.
Sample Input:

```
3 1000 2 3
200
0 2 2
300
27 3 20
5 2000 2 4
300
4 3 0 4
450
7 8 0 13
```

Sample Output:

```
900
impossible
```

4 Summation

Task: You are given a multiset A of positive integers $\{a_1, \ldots, a_n\}$. Now you want add all numbers to compute $\sum_{i=1}^{n} a_i$, but nothing is for free: For each addition of two numbers b, c you have to pay $b + c$. Find a smart way to add the numbers, so that the cost you have to pay is minimized.

We give an example: Assume we are given the integers $2, 6, 3$. If we add at first 2 and 6 we have to pay 8. Then, we add 8 and 3 and have to pay another 11. Therefore, the total cost of the summation $((2 + 6) + 3)$ is 19. A cheaper way would be to add at first 2, and 3 (cost 5) and then 5 and 6 (cost 11). So, the total cost would be 16.

$(0,0)$	$(0,1)$	$(0,2)$	$(0,3)$	$(0,4)$
$(1,0)$	$(1,1)$	$(1,2)$	$(1,3)$	$(1,4)$
$(2,0)$	$(2,1)$	$(2,2)$	$(2,3)$	$(2,4)$
$(3,0)$	$(3,1)$	$(3,2)$	$(3,3)$	$(3,4)$
$(4,0)$	$(4,1)$	$(4,2)$	$(4,3)$	$(4,4)$

Figure 1: The second floor.

Input: There are several test cases specified. Each test case is described by two lines: The first line contains the size of the multiset n with $2 \leq n \leq 20000$. In the next line the numbers $a_1, a_2 \ldots$ will be specified. You can assume that $1 \leq a_i \leq 500000$. The last line of the input contains a single 0.

Output: In each line print the number of the minimum possible payment.

Sample Input:

```
2
56 91
9
66 85 52 22 44 1 59 88 67
0
```

Sample Output:

```
147
1454
```

5 Fire prevention

Task: The computer science department moved to the new Campus Poppelsdorf. You work on the concept of fire prevention. Assume that the second floor is a square and has been divided in 25 parts (see Figure 1): In each of the 25 parts a number of people n_i works. Now you want to establish five assembly points in the case of fire. You want to establish the assembly points in such a way that the time people need to arrive at them is minimized. You can assume that the time to move between different parts is given by the Manhattan distance.

Input: There will be several test cases specified. The first line contains the number of test cases n. Each test case is described as follows: The first number

k denotes the number of parts, where people work. In the following k lines a triple (x, y, z) is described where x and y denote the coordinates of the part and z denotes the number of people working in this part.

Output: Output the total time people need to arrive at the assembly point. (So we look for $\sum_{p \in \text{Persons}} t_p$, where t_p denotes the time person p needs to reach the nearest assembly point.)

Sample Input:

```
2
8
0 0 20
0 1 26
1 2 13
2 0 5
3 0 16
3 2 19
4 1 28
4 2 8
10
0 2 10
0 3 5
0 4 5
1 1 7
2 0 12
2 2 28
2 3 20
2 4 25
4 3 11
4 4 24
```

Sample Output:

```
33
60
```

6 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
```

7 Basic Math

Task: You are given $U, V, W \in \{1, \ldots, 12000\}$. We define

$$S = \Big\{(x, y, z) \in \mathbb{Z}^3 \,:\, (x \neq y) \wedge (x \neq z) \wedge (y \neq z) \\ \wedge (x + y + z = U) \wedge (xyz = V) \wedge (x^2 + y^2 + z^2 = W\Big\}\,.$$

Output the lexicographic smallest element in S. In case that S is empty print "empty set".

Input: There will be several test cases concatenated. The first line contains a positive integer n that specifies the number of test cases. In each of the following n lines a test case will be described. A test case will be described by U, V, W.

Output: For each test case print in a single line x, y and z. If S is empty, print "empty set".

Sample Input:

```
3
1 1 1
88 1136 5298
42 420 842
```

Sample Output:

```
empty set
1 16 71
1 20 21
```

8 Basic Math 2

Task: You are given $U, V, W \in \{1, \ldots, 5 \cdot 10^{18}\}$. We define

$$S = \Big\{(x, y, z) \in \mathbb{Z}^3 \,:\, (x \neq y) \wedge (x \neq z) \wedge (y \neq z) \\ \wedge (x + y + z = U) \wedge (xyz = V) \wedge (x^2 + y^2 + z^2 = W\Big\}\,.$$

Output the lexicographic smallest element in S. In case that S is empty print "empty set".

Input: There will be several test cases concatenated. The first line contains a positive integer n that specifies the number of test cases. In each of the following n lines a test case will be described. A test case will be described by U, V, W.

Output: For each test case print in a single line x, y and z. If S is empty, print "empty set".

Sample Input:

```
3
1 1 1
88 1136 5298
42 420 842
```

Sample Output:

```
empty set
1 16 71
1 20 21
```