

Problem 5 – Three in One

Solving one task is too mainstream. Solving two task is also too mainstream.
You are given three tasks. Solve them all.

First task

A round of Blackjack has ended and it's time to reveal the winner. Use the following two rules to determine the outcome of the round:

- If a player has more than **21** points, he loses.
- Of all the players who have **21** or fewer points, the ones with the maximum number of points are the winners.

You are given a **zero-based** list of integers called **points**, where **points[i]** is the number of points received by the i-th player.

If there is exactly one winner, your program should print the 0-based index of the winning player. Otherwise, it should print **-1**.

Second task

You have a birthday party with your friends! On the table are a variety of cakes of different sizes, with size being measured in "bites". You've decided to take turns picking cakes until all of the cakes are picked by either you or one of your friends. You all like big cakes, so each person on their turn will pick the biggest remaining cake. If there's a tie for the biggest cake, it doesn't matter which of the tied cakes is picked. Since it's your birthday, you get to pick the first cake.

For example, suppose you have 2 friends and there are 5 cakes with sizes of 6, 7, 9, 6, and 4. You will pick the 9 bite cake, your first friend will pick the 7 bite cake, your other friend will get one of the 6 bite cakes, you will get the other 6 cake, and your 1st friend will get the 4. You will get a total of 15 bites.

F will be the number of friends you have with you (not including you). **sizes** will be a list of integers, each of which is the size of one cake - not necessarily in order of size.

Your program should find the total bites of cake you will get.

Third task

Your character in an RPG game has **G1** gold, **S1** silver and **B1** bronze coins. You need **G2** gold, **S2** silver and **B2** bronze coins to buy a new beer. A bank in the game supports four types of exchange operations:

- the bank will give you **9 silver** coins in exchange for **1 gold** coin
- the bank will give you **1 gold** coin in exchange for **11 silver** coins
- the bank will give you **9 bronze** coins in exchange for **1 silver** coin
- the bank will give you **1 silver** coin in exchange for **11 bronze** coins

Find the **minimal number of exchange operations** required for your character to hold at least **G2** gold, at least **S2** silver and at least **B2** bronze coins. Print **-1** if the character doesn't have enough money for beer.

Input

The input data should be read from the console.

The elements of the **points** list are given on the first input line, separated by a single comma.

The elements of the **sizes** list will be given on the second input line, separated by a single comma. The number **F** will be given on the third input line.

Numbers **G1**, **S1**, **B1**, **G2**, **S2** and **B2** will be given on the fourth input line separated by a single space. The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

The output data should be printed on the console.

On the first output line print the solution of the first task (*the index of the Blackjack winner*).

On the second output line print the solution of the second task (*the total bites of cake you will get*).

On the third output line print the solution of the third task (*minimal number of exchange operations*).

Constraints

- **points** will contain between 1 and 50 elements, inclusive. Each element of **points** will be between 1 and 35, inclusive.
- **sizes** will contain between 1 and 10 elements, inclusive. Each element of **sizes** will be between 1 and 10, inclusive. **F** will be between 1 and 5, inclusive.
- **G1**, **S1**, **B1**, **G2**, **S2** and **B2** will each be between 0 and 1 000 000, inclusive.
- Allowed working time for your program: 0.2 seconds. Allowed memory: 16 MB.

Examples

Example input	Example output
21, 20, 19 6, 7, 9, 6, 4 2 1 0 0 0 0 81	0 15 10

Example input	Example output
1, 2, 3, 3, 2 1, 1, 1, 5, 1 3 1 100 12 5 53 33	-1 6 7