Practice Midterm

This exam is:

- open notes/books/any printed resources
- open laptop (anything that you have on your own computer, not online storage)

Online resources that you are allowed to access:

- Gradescope
- course website
- language documentation:
    - Java: https://docs.oracle.com/javase/10/docs/api/
    - C++: https://www.cplusplus.com/reference/ and/or https://en.cppreference.com/w/
- anything else you want since we cannot restrict it, but keep in mind that this will take the time away from the time you have to solve the problems

Instructions:

Solve three out of the four problems given on the next pages. You will **not** get extra credit for solving all four problems.

Grading:

Every exam problem is graded out of 10 points. The total exam grade is the weighted sum computed as follows (assume *scoreN* is a score for a particular problem with *score1 >= score2 >= score3*):

*exam = 5 * score1 + 3 * score2 + 2 * score3*

The total score for a problem is determined by the maximum between zero and the sum of scores for individual tests based on their results. The maximum score for each test is determined by *max_score = 10/number_of_tests*.

test outcome	test score
passed test	*max_score*
wrong answer	*- 0.5 max_score*
runtime error	*- 0.5 max_score*
timeout error	*- 0.5 max_score*
presentation error	*0.75 max_score*

Threeprimes

Michelle is working on her thesis about prime numbers. She proposed a new categorization of primes into groups as follows:

- oneprimes are all the primes whose last digit is 1
- twoprimes are all the primes whose last digit is 2
- threeprimes are all the primes whose last digit is 3
- fourprimes are all … (well, you get it)

She asked all of her friends to count the primes less than some number N that belong to one of the above groups. You were honored with counting the *threeprimes*.

You are going to write a program for which the input and output are as described below.

Input The input consists of a single integer N (1 <= N <= 20,000,000).

Output The program should print a single integer (the number of *threeprimes* that are smaller than N) followed by a newline.

Example 1

```
Input:
6

Output:
1
```

Example 2

```
Input:
15

Output:
2
```

Because 3 and 13 are *threeprimes*.

Example 3

```
Input:
100

Output:
7
```

Because 3, 13, 23, 43, 53, 73, and 83 are all *threeprimes*.

Network

Alice is a network administrator. She is keeping a log of all connections between the computers in the network. Each connection is bi–directional. Two computers are interconnected if they are directly connected or if they are interconnected with the same computer (of course, any computer is directly connected with itself). Alice is wondering if two given computers are interconnected at a particular moment, according to the log.

Write a program to count the number of successful and the number of unsuccessful answers to the questions of the kind:

is *computer* c_i interconnected with *computer* c_j?

Input

The 1st line contains a single integer n, 1 <= n <= 10^6 , the number of computers in the network. At the beginning, there is no direct connection between any two computers. The following lines contains commands in the form:

1. c c_i c_j, which means a new direct link between c_i and c_j is built, c_i and c_j are interconnected now. (a pair of computers can be connected multiple times)
2. q c_i c_j, stands for the question:

   is *computer* c_i interconnected with *computer* c_j now?

You can assume 1 <= c_i, c_j <= n. And there will be no more than 10^6 commands.

Output

Prints two integer numbers to the standard output on the same line, in the order: ‘successful answers, unsuccessful answers’, as shown in the sample output.

Example 1

```
Input:
10
c 1 3
c 2 4
q 6 1
c 3 4
c 9 3
q 2 9

Output:
1,1
```

Example 2

```
Input:
1
q 1 1
c 1 1
q 1 1

Output:
2,0
```

Game Enthusiast

Kou loves playing video games and wants to share her games with her classmates. She has a USB stick but it is too small to hold all of her games. Kou decided to copy only some of her games to that USB stick but struggled to determine which games to select and turned to you for help.

You are given the size of N games and the capacity of the USB stick. You need to find subset of games to store on the memory stick such that the leftover unused space is minimum.

Input

The first line of the input contains two integers V (1 <= V <= 10,000) and N (1 <= N <= 20), representing the capacity of the USB stick and the number of games.

The second line contains N integers, indicating the size of Kou’s games. All of these integers are between 1 and 10,000.

Output

Print one line containing a single integer W, the total space occupied by the games copied onto the USB stick such that the USB stick has the least space unused.

Example 1

```
Input:
5 3
1 3 4

Output:
5
```

Example 2

```
Input:
10 4
9 8 4 2

Output:
10
```

Example 3

```
Input:
20 4
10 5 7 4

Output:
19
```

Example 4

```
Input:
90 8
10 23 1 2 3 4 5 7

Output:
```

```
55
```

Example 5

```
Input:
45 8
4 10 44 43 12 9 8 2

Output:
45
```

Time Master

Eto manages all timers in the Light Kingdom. People usually need her to repeatedly alert them at a fixed interval. They can register a reminder at her using the following instruction:

Register *id interval*

where *id* is the ID of that reminder and *interval* is the interval between two consecutive alerts. Eto will first alert the user after *interval* hours of register request and then alert that user every *interval* hours.

There are a bunch of different reminders requests registered at once. Your task is to process the first N alerts for Eto. If there are more than one alerts occurring at the same time, you should complete them in the ascending order of *id*.

Input

The input consists of two parts. Each line of the first part contains a single register instruction described above where 1 <= #Instructions <= 1000, 1 <= *id* <= 3000 and 1 <= *interval* <= 3000. It is guaranteed that there is no duplicated *id*.

The first part is followed by a single line consisting of a single `#` character, which leads to the second part. The second part consists of only one line, containing a single integer N (1 <= N <= 10000), representing the number of alerts you have to do for Eto.

Output

You should print the *id* of the first N alerts. One *id* per line.

Example 1

```
Input:
Register 2004 200
Register 2005 300
#
5

Output:
2004
2005
2004
2004
2005
```