Universidad del Valle

Super Computing and Distributed Systems Camp Programming Contest

2016

PROBLEM A
BOMBING FIELD RELOADED

Statement

In a very far, far away country, the Military Command (MC) is planning how to destroy an enemy battle field. However, in the same field is located a populated area, soy they must be very careful.

In a simulated scenario, the battle field is described as square area of N x N yards, where 100 <= N <= 10000. Yards are numbered 0 to N-1.

In this simulated scenario, Military Targets (MT) are marked as negative numbers, where the number describes the “strength” of the target. So, a -8 target is stronger than an -2 one. Civilian Targets (CT) are marked in the same fashion, but using positive integers. So, a CT of 8 is more important than a CT of 2. Yards with no interest, are marked as 0.

The MC experts program a series of attacks, with special bombs that can destroy square areas. Each of such bombs has two (2) parameters: the size of the area to destroy and the power itself. Power is an integer number, so its effect on the field is to decrease the number of each yard affected by the bomb in “power” units, if the yard is a CT, and to increase the damage, in the case of MT.

After a number of attacks, MC wants to know:

- How many MT where totally destroyed
- How many MT where partially destroyed, i.e., final number is lower than 0
- How many MT were not touched
- Hoh many CT where affected partially by the bombing
- How many CT where totally destroyed
- How many CT where not touched

Your mission is to write a parallel program that helps MC to take such information

Input Format

Simulated scenarios values are to be read from standard input. The first number describes the size N of battle field. In a second line, the number T of targets, including both CT and MT. Then, T lines, with three integers: coordinates X, Y (0<= X,Y <= N-1) and value of the target (positive for CT, negative if a MT).

After the T lines, a number B describing the number of attacks planned by the MC. Each attack is described by four numbers: coordinates X, Y (0<= X,Y <= N-1), the size R of the square radius of the bomb, and the power (P) of the bomb. Square radius means: given X and Y coordinates of the bomb, each yard within (X-R,Y-R) and (X+R,Y+R) is affected by the bomb in P units. A ratio of 0 means a direct attack.

The number B of attacks has no limits, and certain X,Y coordinates could be repeated. Bombs has effect only in the battle field, so coordinates less than 0 or greater than N-1 must not be considered.

It is guaranteed there are at least one CT and one MT, and neither CT’s nor MT’s has the same coordinates.

Output Format

After B attacks, you must write on standard output:

```
Military Targets totally destroyed: A
Military Targets partially destroyed: B
Military Targets not affected: C
Civilian Targets totally destroyed: D
Civilian Targets partially destroyed: E
Civilian Targets not affected: F
```

Example

Input:

```
10
4
0 0 8
5 5 100
1 1 -2
7 7 -6
5
2 1 2 3
1 1 1 4
7 7 0 3
6 6 4 8
9 9 8 1
```

Output for the input example:

```
Military Targets totally destroyed: 2
Military Targets partially destroyed: 0
Military Targets not affected: 0
Civilian Targets totally destroyed: 0
Civilian Targets partially destroyed: 2
Civilian Targets not affected: 0
```

REMARKS

1) How different would be your solution… if you have to *decide* how to destroy the most MT with a given amount of bombs?

2) … if you have to maximize destruction without casualties, with a limited amount of bombs?

3) … if bombs that touch CT can't be used, but bombs that destroy MT are too slow to calculate? i.e. military bombs delays 1 minute calculating destruction

PROBLEM B
WEIRD PUZZLE

Statement

A popular cross-word game, is to find words inside an square matrix, in any of vertical or horizontal directions. A weird version of this puzzle, is when we must find the words even if they are sparse along a line. i.e., find all the letters of the word in the right order in the same vertical or horizontal line, but there could be additional letters in between.

Moreover, words can be in different directions (right to left, upper to lower and vice-versa) and they can wrap around the matrix, it means that a word beginning at the right most side of the matrix, can finalize at the left side of the matrix, and the same for vertical direction.

Your challenge if to find at most different words as you can, given a square matrix of letters, and a list or words. You must simply have to show the number of words found.

Input Format

Matrixes and words are to be read from standard input. The first number describes the size N of square matrix of letters, 1 <= N <= 50000. In the next N lines comes the contents of each row of the puzzle, containing N letters of the English alphabet, lowercase. Note that there can be large lines!

After the N lines, it is an integer representing the number W of words. Then, W lines, one word per line, lowercase. No entire words will be repeated, and it is guaranteed no words are sub-strings of another one.

Output Format

The output is simply an integer, representing the number of different words you find inside the puzzle. Note that a word could appear more than once, in such case, you have to count *only the first time* the word is found.

Example

Input:
```
5
aocde
adrft
wkdig
wabgt
ooooo
4
ooo
ar
bw
zs
```

Output for the Input example:

```
3
```

REMARKS

1) How different would be your solution… if you have to count every word found?
2) … if you have to find words on different puzzles?
3) … if you have to count until a given condition, for example, until you reach N appearances of an specific word?

PROBLEM C

YET MORE PRIMES

Statement

Prime test is a common task in many applications, specially for security, cryptography, and other interest areas. There are a number of techniques used to verify if a number is prime, and the common problem is that all techniques consumes lots of time.

The problem is easy to solve, giving the number to be tested and time enough to calculate. But a weird scientist took our numbers, and cut them in two halves, so we have first to re-arrange the numbers, looking for the primes. In this case, we have the certainty that *all* the numbers we have are primes, but the problem is how to re-order them

Input Format

Numbers will be read from standard input. In the first line, there is the number P of prime numbers (1 <= P <= 100). Then, P lines with the *first half* of the numbers to be tested, with no order. Next, P lines with the *second half* of the numbers with no order. Prime numbers in this problem are taken as strings, so *first half* and *second half* simply means an arbitrary cut of such string. For instance, prime number 504155039 can be cut in 5041 and 55039, or 504155 and 039

Note that second half numbers could begin with a lead zero, so you must take it into account. Of course, this doesn't happens with the first half.

Output Format

You have to print the list of the P prime numbers, in ascendant order

Example

Input:

```
4
50415
100
5041
55750
01
0213
5039
55713
```

Output for the Input example:

```
1000213
5575001
504155039
504155713
```

REMARKS

1) How different would be your solution… if you are not sure ALL numbers are actually prime numbers?
2) … if there are combinations that produce more than ONE single prime number?
3) … if you have to test not just TWO substrings but three? An arbitrary number?
4) … if you have to deal with an arbitrary increasing number of strings UNTIL find a given number of prime numbers?