



# Problem Set

## ACM-ICPC Programming Competition

### 2019

### Day 2

#### Instructions

1. Do not open the booklet unless you are explicitly told to do so. You can only read these Instructions below.
2. Do not create disturbance or move around unnecessarily in the arena.
3. If you have any question regarding the problems, send a clarification to judges using PC^2.
4. Before submitting a run, make sure that it's executable via command line. For Java, it must be executable via "javac" and for GNU C++ via "g++". In particular, Java programmers need to remove any "package" statements and source code's file name must be the same as of main class. **C++ programmers need to remove any `getch()` / `system("pause")` like statements.**
5. Do not attach input files while submitting a run, only submit/attach source code files i.e. \*.java or \*.cpp.
6. Source code file name should not contain white space or special characters.
7. You will have to make an Input file for every problem based on the format provided in the Sample Input. For example, **the input file for Problem 1 shall be "P1\_input.txt", for problem 2, it shall be "P2\_input.txt" and so on.** The **outputs of all the problems would have to be on the console** according to the sample output given in.
8. While programming, **do not open input files by providing absolute/complete paths** (absolute paths of your machine and judge's machine would be different). Just open them from the current directory from where your code is running.
9. Please, don't create/open any file for output. All outputs must be on console.
10. Please strictly meet the output format requirements as described in problem statements, because may be some auto judge will be evaluating your program which will compare your output with judge's output byte-by-byte and not tolerate even a difference of single byte. So, be aware! **Pay special attention to spaces, commas, dots, newlines, decimal places, case sensitivity etc.**
11. Unless mentioned in some problems, all your programs must meet the time constraint of 20 seconds.
12. The decision of judges will be absolutely final.



## PROBLEM 1: Chessboard painting

Time limit: <3 seconds

A standard chessboard has black and white painted square boxes organized as an 8x8 matrix of total 64 squares. Malik is setting up a new online business of custom painted chessboards. He has bought a large stock of completely white chessboard and black paint. Users will order arbitrary painted chessboard and Malik will paint the user-defined black boxes with black paint and leave the remaining boxes, as they are already white. There is only one restriction on Malik's website, that a user can paint an entire row or an entire column black.

To maximize Malik's profit, your job as a programmer is to analyze the ordered board and save black paint. That is, given an  $8 \times 8$  matrix with arbitrary rows and/or columns of black squares, you have to find out the minimum number of strokes that completes the chessboard. A stroke can only be either horizontal (row) or vertical (column) and it paints all concerned squares perfectly black with no need for a second stroke.

Can you help Malik?

## Input

The input file contains 8 lines, each of the lines contains 8 characters. The given matrix describes the user's requirements, W character stands for a white square, and B character for a square painted black.

## Output

Output “Case #:” where # is the test case being solved, and the only number, the minimum amount of rows and columns that Malik has to get painted on the white chessboard to meet the user's requirements.

Sample input	Sample Output
WWWBWWBW BBBB BBBB WWWBWWBW WWWBWWBW WWWBWWBW WWWBWWBW WWWBWWBW WWWBWWBW WWWBWWBW	Case 1: 3



## PROBLEM 2: The super healing of Markhors

Time limit: <10 seconds

According to the **Central Dogma of Molecular Biology**, a **gene** from a **genome** is first **transcribed** into a strand of **RNA**. Then, the **RNA** transcript is translated into an **amino acid** sequence of a **protein**.

**DNA** (deoxyribonucleic acid) is a hereditary material in humans and almost all other organisms. Nucleotides in **DNA** contain four different nitrogenous bases: adenine (**A**), guanine (**G**), cytosine (**C**), and thymine (**T**). A **gene** is a segment of **DNA**. A **Genome** is an organism's complete set of **DNA**.

The Markhor /'mɑ:rkɔ:r/ (Capra falconeri), is a large Capra species native to Central Asia, Karakoram and the Himalayas. It is listed on the IUCN Red List as Near Threatened since 2015. The Markhor is the national animal of Pakistan. It is also known as the screw horn goat, Pashto: مرغومی Marḡūmi and Persian/Urdu as مارخور.

Historically, in the ancient scriptures and folktales, Markhors are depicted to be miraculous holy animals



– their blood could be used to clean a wound and flesh consumed by injured persons to help cure the disease instantaneously. Scientists have identified the exact protein that is responsible for this behavior. However, they are not sure of the exact **DNA** sequence which was eventually translated into this specific protein. They know the DNA sequence which is approximately same across all the Markhor species. The life of a bioinformatics scientist would be easy if these sequences were completely conserved across all the Markhors. But the reality is more complex, as these DNA subsequences may mutate, i.e., vary at some positions.

You are required to solve a much simpler version of this problem: you are given the DNA sequence corresponding to this protein, a chromosomal DNA sequence from Capra falconeri genome, and are asked to find the **best matching regions** between the genomes of a group of Markhors, **with exactly one mismatch**. You are required to discover these similar regions without any prior knowledge of how the regions look like.

So, the problem you are going to solve would be as follows: Given a random sample of  $t$  **DNA** sequences, find the **pattern** that is implanted in each of the individual **DNA** sequences, with exactly one **mutation** (variation).

As an example, consider the following randomly generated **DNA strings** with implanted pattern **TTTTCCC** of length 6.

```
GAGGATTCGTTTGCCGAGG
AAATTTACCTAGAATGTCA
AGTTCCCCCCTGGACACAAT
GAATTATCCCAAGTCCTAA
AGACGAAGGTTCCCGTTGA
```

Notice that each of the implanted pattern has one mutation, as underlined.

### Input

The input consists of multiple test cases. The first line of input is the number of test cases ( $1 \leq N \leq 100$ ). Each of the following  $N$  lines contains an integer  $t$  followed by an integer  $k$ , followed by a collection of  $t$  **DNA** strings. Note that both the  $t$  and  $k$  can have any value between 2 and 9, inclusive.

### Output



Output basically *returns* all the **substrings**, each of which is of length  $k$ , and has exactly **one mutation**.

For each test configuration in the input first, output “Case  $n$ : ” where  $n$  is the test configuration number, followed by all the **substrings** of length  $k$ , separated by spaces.

Sample input	Sample Output
4	
4 2 TAT GGG CTC AGA	Case 1: GT TG
3 3 CGTTT GGTGG CCCCT	Case 2: CGT GCT
4 3 TAGACCG ACGGAAT GCCATAG CTTTTAC	Case 3: GAC TAA TAT
4 4 AAATACTG TACCAGTT GTCAAGGG AAATACGT	Case 4: AAGT CAAT



### PROBLEM 3: Property tycoon

Time limit: <5 seconds

As a property tycoon, your best investment is to buy property while its price is low, invest money on that property to renovate it, and sell it for a profit. Ideally, you would like to buy a run-down property whose current price is low but whose potential is quite high. To achieve this, you need to calculate the current cost (in millions of Rupees) of the property ( $C$ ), estimated cost of renovation ( $R$ ), and its potential selling price ( $S$ ). You net profit will then be  $S - (C + R)$ .

Ideally, you would only invest in properties that will result in a net profit. To maximize your profit, you should choose and buy all those properties that result in a profit. This, however, will result in the city administration selling only profitable properties while the rest of the neighborhood blocks would remain. The smart mayor of Jinkalala, a city devastated by recent economic recession and hoping to make an economic comeback, has come up with a smart plan.

His plan makes for selling entire blocks instead of individual properties. This way a block will sell both properties with a positive net profit as well as negative profit. Blocks are made up of contiguous rectangular properties and can be of any size but not trivial (consisting of only one property). There is also another catch – you can only make ONE purchase. Therefore, you must choose the block with the largest cumulative net profit.

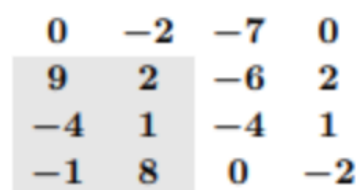


Figure 1

For instance, in the collection of the properties (a neighborhood) given in figure 1, the block corresponding to the maximal profit is highlighted (profit is 15).

#### Input:

The input consists of multiple test cases. The first line of the input is the number of test cases,  $N$ . Each of the following  $N$  lines contains (i) the number of rows,  $r$  (ii) the number of columns,  $c$  (iii) a row-wise entry of the data.

#### Output:

For each test case, print a single line that says “Case  $i$ ”, where  $i$  is the test case number, followed by the starting point (top-left), the ending point (bottom-right), and the profit of the investment packet.

**Note:** the first sample input/output corresponds to the data in figure.

Sample input	Sample Output
2	
4 4 0 -2 -7 0 9 2 -6 2 -4 1 -4 1 -1 8 0 -2	Case 1: (2,1), (4,2), 15
4 5 1 2 -1 -4 -20 -8 -3 4 2 1 3 8 10 1 3 -4 -1 1 7 -6	Case 2: (2,2), (4,4), 29



## PROBLEM 4: The secrets of starfish

Starfish are not fish, but rather marine invertebrates or "echinoderms" in the class Asteroidea. Therefore, scientists prefer to call them sea stars. There are approximately 1,800 different species of Asteroidea. They are known for their ability to regenerate amputated limbs, and are characterized by flexible arms attached to a central disc. The central disc is essential to regeneration in nearly all sea star species.



A sea star that has the ability to regenerate amputated limbs must first undergo a repair phase to heal the exposed wound. Once the wound is healed, the sea star can begin to generate new cells, which in turn, sparks new growth. Regeneration can take anywhere from several months to years. Cell proliferation that results in the growth of the new limb occurs in the final phase. If the final phase is interrupted, the resulting new limb may be deformed.

While working on its ability to regenerate, in 1980s, Swedish molecular scientists were amazed by particular unique characteristics of its genome – instead of 20 amino acids, it contains 26 amino acids, all represented by alphabets in capital letters. In addition, like typical Eukaryotes (animals), the sequence (and identity) of starfish's amino acid sequences matters but, unlike other reptiles and animals, some of their amino acids can morph into other amino acids at will but retain the functional of the original amino acid. Extensive experimentation by the scientists has led them to decoding the amino acid conversions and they are able to identify the equivalent amino acids. In the early wet lab experiments, Swedish scientists at Karolinska Institute were able to decipher the functions performed by some of the most commonly occurring proteins in starfish but determining the alternate protein remains elusive.



You are given a list of equivalent amino acids and a list of pairs of proteins that may (not) be alternates. Your task is to verify whether the proteins in each pair match. Two proteins match if they have the same length and if each amino acid of the first protein can be turned into the corresponding amino acid of the second protein by using the available equivalence rule zero or more times.

### Input

The first line contains space-separated amino acid letters ( $X_1 X_2 X_3 \dots X_n$ ) and the second line contains the equivalent of each amino acid of line 1 separated by space ( $Y_1 Y_2 Y_3 \dots Y_n$ ). Then each of the next line contains exactly two space separated proteins to check for equivalence. Equivalence amino acids and protein pairs use only uppercase letters 'A'–'Z', and each protein contains **at least 3** and **at most 20** letters.

### Output

For each pair of words, display "Case #:" where # is the test case being processed, and "Yes" if the two words match, or "No" otherwise.

Sample Input	Sample Output
CIKORTTSWF TRPCOEFHPS CORK CORK CAN CON RACKTTS ROCKETS OFF THS FSTIOOKS SHEROCKS	Case 1: Yes Case 2: No Case 3: No Case 4: Yes Case 5: Yes





## PROBLEM 5: Programming Talent

Sam is competing in the preliminary round of a regional programming talent show and wants to advance to the next (global) round. In the talent show, each contestant gets 10 minutes to impress the judges. After all the contestants have performed, each of the judges will cast two distinct votes. A vote can be either in favour of a contestant (meaning this contestant should advance) or against a contestant (meaning this contestant should not advance). The number of contestants that advance to the next round is not known in advance; if there are only very bad contestants, then it is possible that nobody will advance, or if everybody is amazing, then everybody may advance.

Sam is afraid that the judges will not appreciate his programming talents and wants to use his other talent to advance to the next round: hacking. Having gained access to the jury system, Sam is capable of overriding the regular process of counting votes, and instead selecting exactly which contestants advance to the next round. The only problem is that he has to be careful not to arouse suspicion.

Each judge expects that at least one of his (or her) own two votes corresponds to the outcome of the contest. If the outcome contradicts both votes, the judge becomes alarmed. As an example, assume Judge Harry casts a vote in favour of Pete and a vote against Sally. If Sally advances and Pete does not, Judge Harry will be alarmed and may discover Sam's tampering with the system.

Since Sam's programming talents are limited (otherwise, he would not have needed his hacking talents), he needs you to make a program that finds out if there is a set of contestants, which includes himself, that he can select to advance to the next round by hacking the jury system, such that it does not alarm any of the judges.

### Input

For each test case, the input is as follows:

- One line containing two positive integers: the number of contestants  $n$  ( $2 \leq n < 1000$ ) and the number of judges  $m$  ( $1 \leq m < 2000$ ).
- $m$  lines containing the votes of each judge.

Each of these line contains two integers: the numbers  $a$  ( $1 \leq |a| \leq n$ ), and  $b$  ( $1 \leq |b| \leq n$ ), the two votes of this judge ( $|a| \neq |b|$ ).

A vote  $x < 0$  means that the vote is against advancement of contestant  $|x|$ .

A vote  $x > 0$  means that the vote is in favour of contestant  $|x|$ .

Contestants are numbered  $1 \dots n$ . Sam is contestant 1.

### Output

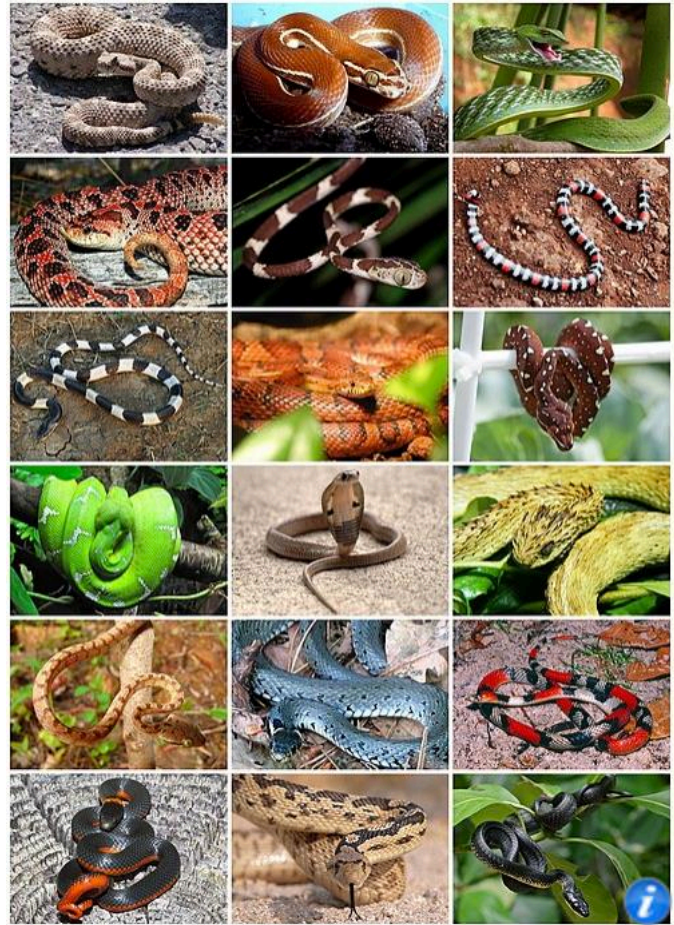
For each test case  $i$ , print one line of output containing "Case  $i$ : " and the word 'yes' if there is a set of contestants that advances to the next round that includes Sam, and does not alarm any of the judges. If there is no such set of contestants, the line should contain 'no'.

Sample input	Sample Output
4 3 1 2 -2 -3 2 4 2 4 1 2 1 -2 -1 2 -1 -2	Case 1: Yes Case 2: No



## PROBLEM 6: THE VENOMOUS SNAKES OF AUSSIE DESERTS

Living snakes are found on every continent except Antarctica, and on most smaller land masses; exceptions include some large islands, such as Ireland, Iceland, Greenland, the Hawaiian archipelago, and the islands of New Zealand, and many small islands of the Atlantic and central Pacific oceans.[4] Additionally, sea snakes are widespread throughout the Indian and Pacific Oceans. More than 20 families are currently recognized, comprising about 520 genera and about 3,600 species.[5][6] They range in size from the tiny, 10.4 cm (4.1 in)-long Barbados thread snake[7] to the reticulated python of 6.95 meters (22.8 ft) in length.[8] The fossil species *Titanoboa cerrejonensis* was 12.8 meters (42 ft) long.[9] Snakes are thought to have evolved from either burrowing or aquatic lizards, perhaps during the Jurassic period, with the earliest known fossils dating to between 143 and 167 Ma ago.[10] The diversity of modern snakes appeared during the Paleocene epoch (c 66 to 56 Ma ago, after the Cretaceous–Paleogene extinction event). The oldest preserved descriptions of snakes can be found in the Brooklyn Papyrus.



Most species are nonvenomous and those that have venom use it primarily to kill and subdue prey rather than for self-defense. Some possess venom potent enough to cause painful injury or death to humans. Nonvenomous snakes either swallow prey alive or kill by constriction.

In Australia, bite of several species is extremely venomous for small mammals and animals. Mainly, the scientists have found a number of proteins (character strings) in these snake species that are venomous. The common thing in all these proteins (character strings) is that regardless of position, the group contains a specific character sequence, "TPEXXXRYIE", where T, P, E, R, Y, and I are essential amino acids (characters in the string) and XXX represents 1, 2 or 3 amino acids (any of 26 letters of English alphabet in capital).

You are given a list of proteins (character sequences) extracted from various snakes in Pakistan, and your job is to determine if these character sequences contain the poisonous character sequence.

### Input

Each line in the input contains a single protein sequence (string) that may (not) contain the requisite sequence of characters. The proteins are of various lengths where length of a protein must be greater than 10 but less than 100.

### Output

For each pair of words  $i$ , display "Case  $i$ :" followed by **Yes** if the string contains the required domain, and **No** otherwise. The valid protein must have a length greater than 10 but less than 100.





Sample Input	Sample Output
RUNTPRQWTPELRYIETYQPR	Case 1: Yes
KPQLZMHTPEATEKRYIEPLOQ	Case 2: No
MKLAOPZTPATERYIETPERYIE	Case 3: No
LNNNPQZTPECTKRYIE	Case 4: Yes
TPETPETPEKKRYIERIERYIE	Case 5: Yes
TPERYIETPERYIE	Case 6: No
TPEATERYIELLLKPL	Case 7: Yes
TPEARYIE	Case 8: No