Seven Lakes HS Kickoff 2023

Pirates of Stroudonia

Table of Contents

Coasting on the High Seas..................................3

The Pirate Jimmy...4

Mutiny on the High Seas..................................5

X Marks The Spot...6

Hit the Rack!..9

Message in a Bottle.....................................10

We're In For Some Chop!.................................12

Cannonball Run..13

Infinite Pirate Theorem.................................14

Crew Morale...15

Polly Wanna Cracker.....................................16

Shipbuilding..17

Send for the Armada.....................................18

Quartermaster...20

Navigating the Stroudonian Seas.........................22

Why Did I Take German?..................................24

Rough Seas..25

Hoist the Mizzenmast!...................................27

Coasting on the High Seas

Problem Description: Our Glorious Leader loves sailing on his ship, the SS High Seas! Print out the following ASCII art of his ship:

```
     |    |    |
    )_)  )_)  )_)
   )___))___))___)\
  )____)____)_____)\
_____|____|____|____\
\ @ @ @  SS HighSeas/
 -------------------
```

Input Description: None

Output Description: Output the ASCII art. Trailing spaces and trailing newlines will be ignored.

The Pirate Jimmy

Problem Description: Jimmy is a pirate. As he is a pirate, there are only two kinds of objects to him: gold coins and not gold coins. Help Jimmy distinguish gold coins from not gold coins.

Input Description: The input will consist of a single line, containing a string. This may be randomly capitalized, ignore the capitalization.

Output Description: The output should be a single line, either "yes" if the input is a gold coin or "no" if the input is not.

Sample Input 1:
gold coin

Sample Output 1:
yes

Sample Input 2:
banana

Sample Output 2:
no

Sample Input 3:
GOLD CoIN

Sample Output 3:
yes

Sample Input 4:
goldcoin

Sample Output 4:
no

Mutiny on the High Seas

Problem Description: Oh no! The pirates want to start a rebellion to overthrow the Black Angus. There are N pirates where the ith pirate has an attack value of a_i. The Black Angus is only strong enough to defeat some set of pirates that all together have no more than M strength points. Additionally, each pirate has either 1 or 2 swords (it is possible for a pirate to have a higher attack value but fewer swords). The Black Angus has the ability to collect these swords after each pirate is defeated. Help the Black Angus find out the maximum number of swords he can collect without fighting more than M strength points.

Input Description: The first line contains N (1 ⩽ N ⩽ 2 * 10^5), the number of pirates, and M(1 ⩽ M ⩽ 10^9), the maximum number of strength points the Black Angus can fight off without being overthrown. The next line contains N-separated integers where the ith integer is a_i (1 ⩽ a_i ⩽ 10^9), the attack value for the ith pirate. The line following contains N separated integers where the ith integer is s_i (1 ⩽ s_i ⩽ 2), the number of swords the ith pirate has.

Output Description: Output the maximum number of swords the Black Angus can collect without fighting more than M strength points.

Sample Input:

```
3 6
2 3 4
1 2 1
```

Sample Output:

```
3
```

Sample Explanation:

Killing off the first two pirates we have a total strength value of 5, which is less than 6. This maximizes the amount of swords we can collect at 1 + 2 = 3.

X Marks The Spot

Problem Description: There's nothing a pirate loves more than good old-fashioned buried treasure! Recently, your crew have found some maps pointing to ancient treasure and you are eager to get rich!

However, there's an issue. Due to practical limitations, you can only stay on land for a certain number of days, L. Similarly, you can only move at sea for a certain number of days, S. Each day, regardless of whether you are at sea or land, you can only move 1 tile (this includes diagonals!). The type of movement you carry is based on the current tile you are on. For instance, if you move from land to sea, you consume a land resource.

Furthermore, there are certain areas that are neither land nor sea that you can never step on. To resolve this issue, your pirate scientist, Pirate Nutt, has planted some bombs in the grid to destroy these barriers which will reveal either land or sea underneath. Each bomb has a radius R and will destroy all walls within an R radius. To clarify, a wall at location (a, b) will be destroyed by a bomb at (c, d) with radius R if $|a - c| + |b - d| \leq R$.

Detonating these bombs is expensive, thus, you must find the minimum number of bombs that must be detonated to pave a way to the treasure. Note that the bombs **MUST** be detonated in increasing order of explosion radius. Ties are settled by considering the detonations in row-major order, moving across the first row and repeating the process for each successive row. Being a smart pirate, the Black Angus only goes for the treasure that he can get to. It's your job to find out whether you can get to the treasure or not and the minimum number of bombs needed.

Input Description: The first line of input contains the number, T (0 < T < 100), of test cases. For each test case, the first line contains 4 numbers: X, Y, L, S, separated by spaces (0 < X, Y < 500). The next Y lines will have a string of X characters each, creating a matrix.

On this matrix, there will be a single character, \`o\`, representing the current location of your ship. There will be another character,

`x` representing the treasure. For the rest of the matrix, `,` characters represent land, ‘ ‘ represents water, ‘#’ represents a wall with land underneath, and ‘*’ represents a wall with water underneath. The next Y lines will contain an integer matrix, each row having X space-separated integers. An element will be 0 if there is no bomb on that space and a positive integer representing the radius of the bomb otherwise.
Note: Your first move away from the starting point WILL NOT consume a land/sea resource but WILL count towards your total steps.

Output Description: If it is possible to reach the treasure given the limitations, print the least number of steps it takes and the number of bombs that must be blown up in this format: “We can make it in X steps with Y bombs!” where X is the number of steps and Y is the number of bombs. Additionally, make sure to check for singulars. For example, it is “1 step” and not “1 steps”. Remember, minimizing bombs is the priority! If not possible, output a dialog (“We cannot make it… ”).

Sample Input:

```
2
4 4 2 0
o,,,
,,##
, #x
,,,,
0 0 0 1
0 0 1 0
0 0 0 0
0 0 0 0
4 4 1 0
*,,x
,,,,
,*,*
o,,,
0 0 0 0
0 5 5 0
0 5 5 0
0 0 0 0
```

Sample Output:

```
We can make it in 3 steps with 2 bombs!
We cannot make it...
```

Sample Explanation:

For the first sample case:
Blowing up the first bomb will yield the following matrix:

```
o,,,
,,#B
, #x
,,,,
```

Where B represents a land cell revealed by the bomb. We cannot reach the treasure after blowing up this bomb.

Blowing the second bomb will yield the following matrix:

```
O,,,,
,,BB
, Bx
,,,,
```

The strategy for getting to x is detailed as follows:

We can start off on the right side of the tile (1, 0) taking 1 step but without using any resources. We use both of our land moves by going down and to the right diagonally to reach the x, each one taking a step for a total of 3 steps.

For the second sample case, we only have 1 land move, so no matter where we start, we can't make it to the end.

Hit the Rack!

Problem Description: The pirates aboard the Black Angus's ship really care about how much time they spend asleep. Pirates Nutt, Pittman, Castillo, and Panter strive to reach at least 8 hours of good sleep every night.

However, they are too busy to calculate how many hours they've slept. It's your job to help them determine whether they've slept enough.

Input Description: The first line of input contains the number N (0 < N < 10000), the number of cases. The next N test cases each contain 2 times on a single line: the time they sleep and the time they wake up the next day. Assume the 1st input time is PM and the second one is AM.

The input times will also never have an hour hand of 12. Furthermore, you do not have to consider daylight savings or any other time complexities.

Output Description: Print out “getting the ZZZs” if they sleep at least 8 hours and “gonna take a nap” if they don't.

Sample Input:

```
3
8:30 5:30
9:20 6:20
11:30 1:20
```

Sample Output:

```
getting the ZZZs
getting the ZZZs
gonna take a nap
```

Message in a Bottle

Problem Description: Our Glorious Leader loves leaving his mark on the world. To do this, he writes messages, places them inside bottles, and throws them into the ocean. Naturally, he cannot be bothered to write these messages, so he wants to find the best writer among his crew to appoint as his scribe. The best writer will have the highest **pirate words per minute**, a statistic that accounts for both speed and accuracy.

To calculate **pirate words per minute**, you must take the number of correct characters written and divide by 5 to get the number of words written correctly. Then you must divide the number of words by the number of minutes spent writing.

Input Description: The first line will contain the message that the Glorious Leader wants his scribes to write. This message is guaranteed to be alphanumeric with spaces (only letters, numbers, and spaces). The second line contains a single integer **N**, the number of scribes in the Glorious Leader's Crew. Following the second line are **N** groups of 2 lines, the first containing the scribe's writing record and the second containing the integer number of seconds the scribe took to finish writing. Writing record is guaranteed to be alphanumeric except for spaces and the # symbol, which means that the scribe erased a character. For example, "abc#" is "ab" and "abc##" is "a".

Output Description: Output the pirate words per minute of each scribe in the order they were given rounded to two decimal places, one on each line. On the final line, print out the highest pirate words per minute rounded to two decimal places.

Sample Input:

```
Thar she blows
2
Tharr shee## blowss
15
Thar she blows
20
```

Sample Output:

```
8.00
8.40
The highest pirate words per minute is 8.40
```

Sample Explanation:

The first scribe wrote out "Tharr sh blowss" after accounting for the backspaces. Compared to "Thar she blows" (the prompt), we can see that the first scribe wrote 10 characters correctly in 15 seconds, resulting in 8 pirate words per minute.

The second scribe wrote out "Thar she blows". Compared to the prompt, the second scribe wrote all 14 characters correctly in 20 seconds, resulting in 8.4 pirate words per minute.

We’re In For Some Chop!

Problem Description: Our Glorious Leader needs to plan out the structure of the ocean ahead. He has mapped out the waters ahead as a 1-dimensional array. Each cell has a certain height describing how tall the wave at that point is. He wants to find out where the peaks are in his map. A peak will be defined as a cell where the height of the cell is strictly greater than the cell to the immediate left and right. Note that this means the first and last cells can never be peaks.

Input Description: The first line of input contains T, the number of test cases.
The following test cases each contain a single integer N on the first line followed by N integers, the heights of the waves on the second line.

Output Description: Output the number of peaks in the water.

Sample Input:

```
3
5
1 2 1 2 1
5
1 2 2 2 1
5
4 2 6 5 1
```

Sample Output:

```
2
0
1
```

Sample Explanation:
TC 1: Waves 2 and 4 are peaks.
TC 2: No waves are peaks. None of the 2s are peaks because they aren’t strictly greater.

Cannonball Run

Problem Description: The Black Angus owns a powerful battleship equipped with many terrifying cannons, but these powerful weapons often need restocking. Each cannon on his ship has the same maximum capacity, but some cannons have used more cannonballs than others. Additionally, the Black Angus can only order cannon balls in bulk, in a set amount with a set cost. Your job as the accountant is to find out how much it will cost the Black Angus to buy sufficient cannonballs to fully restock the ship.

Input Description: The first line of input contains the number N (0<N<50) of test cases. For each test case, the first line contains a number C indicating how many cannons there are, followed by a number B, the maximum capacity for each cannon. The next line will contain C integers representing the number of cannonballs currently present in each box. The last line will contain two integers representing the cost in coins of a bulk shipment of cannonballs and how many cannonballs are in each shipment.

Output Description: Output the total cost to fully restock.

Sample Input:

```
3
5 6
1 2 3 4 5
200 5
3 10
10 9 3
50 7
4 5
0 0 0 0
199 3
```

Sample Output:

```
600 coins
100 coins
1393 coins
```

Infinite Pirate Theorem

Problem Description: The Infinite Monkey Theorem states that if a monkey hits keys on a typewriter at random for an infinite amount of time, it will eventually type out any text, including the complete works of William Shakespeare (yes, this is an actual theorem). The Infinite Pirate Theorem is very similar, except they have a more limited vocabulary. The pirate's alphabet is limited to the characters in the phrase "argh matey" so they won't type out the complete works of William Shakespeare. Find out which phrases can be generated given an infinite amount of time. Note that pirates don't know any punctuation either.

Input Description: The first line of input contains T, the number of test cases.
The following T test cases each contain a string, the phrase the pirates must type out.

Output Description: Output "Yes" if the pirates can type it and "No" if they can't.

Sample Input:

```
3
ArGh mAteyyyy ARGHHHHH
TREASUREEEE
HAHAHHAHAHAHHA
```

Sample Output:

```
Yes
No
Yes
```

Sample Explanation:

Test case 1: All letters are in the pirate's vocabulary
Test case 2: U is not in the pirate's vocabulary
Test case 3: H and A are in their vocabulary

Crew Morale

Problem Description: The Glorious Leader believes that keeping the morale of his crew up at all times will let them work at maximum efficiency. In order to keep track of his crew's morale, he devised a formula to calculate their total happiness value. The formula is as follows:

Morale = $(Total\ Wealth)^2/\sqrt{\#\ Crew\ members} + Leadership\ quality$

Input Description: The first line of input contains T, the number of test cases.
The following test cases each contain 3 doubles. The total wealth of the crew, the number of crew members, and the Glorious Leader's leadership quality. Note that the Glorious Leader's leadership will always be above 100 because he's the Glorious Leader.

Output Description: Output the crew's morale rounded to 2 decimal places.

Sample Input:

```
4
35.44 7 200.0
10 10.0 1000
4 25.0 2000
2 4 100
```

Sample Output:

```
674.72
1031.62
2003.20
102.00
```

Polly Wanna Cracker

Problem Description: In his quest to explore the Stroudonian seas, the Glorious Leader has enlisted some of his men and a parrot, Polly, to form an adventure crew. The men quickly find out that whenever they say something, the parrot repeats it. Because the Glorious Leader is forgetful, he often says something to himself but forgets it, even if it is important. Luckily, the parrot repeats what he says too, although an arbitrary (random) number of times. Please help the Glorious Leader remember what he said!

Input Description: The first line includes $t (1 \leq t \leq 100)$, the number of test cases. The next t lines describe each test case, with s, denoting the string squawked by the parrot, being on each line.

Output Description: For each test case, output the **shortest** possible phrase such that s is equal to that phrase repeated k number of times, for any integer k.

Sample Input:

```
4
aba
abababab
stringstringstring
xxxxxxx
```

Sample Output:

```
aba
ab
string
x
```

Shipbuilding

Problem Description: Our Glorious Leader is debating on what material to build his ship out of. He knows that the waters ahead will require a ship of strength at least X. He also has 3 materials he can choose from: Plastic, Wood, and Metal, each with strengths A, B, and C respectively. Also, if the Glorious Leader can construct his ship out of multiple materials, he chooses the weakest one since it's cheaper. Help him find what material to construct his ship out of.

Input Description: The first line of input contains T, the number of test cases.
The following test cases each contain 4 space-separated integers, X, A, B, and C. $(A < B < C)$, $(X \leqslant C)$.

Output Description: Output the material that the Glorious Leader must use

Sample Input:
```
3
0 2 4 6
4 2 4 6
5 2 4 6
```

Sample Output:
```
Plastic
Wood
Metal
```

Send for the Armada

Problem Description: Your ship is under attack and you and your Glorious Leader are trying to call for backup. You must help the Glorious Leader write and encrypt a message using a Caesar cipher, which he will then launch out of his personal catapult to a nearby ship (don't worry about the logistics)!

Write a program that will read in an int **i** and a String **s** and print out a Caesar cipher by encoding **s** with a shift of **i**.

A Caesar cipher is a cipher created by shifting each letter in a message to a different letter that is **i** letters away in the sequence of the alphabet.

For example, if **s** is "PIRATE" and **i** is 1, your program will shift each letter in "PIRATE" to the right by 1 and print "QJSBUF".

If **i** > 0, it's an **i** shift to the right, and if **i** < 0, it's an **i** shift to the left.

Input Description: The first line contains a number, **n**, which denotes the number of test cases. The first line is followed by **n** lines, each containing **i** and **s** separated by a space.

Output Description: Print out each encoded string. Make sure to ignore spaces and punctuation and don't forget to capitalize every letter!

Sample Input:

```
3
-2 Hello World
1 Black Angus
28 Treasure!
```

Sample Output:

```
FCJJM UMPJB
CMBDL BOHVT
VTGCUWTG!
```

Sample Explanation:

For the first testcase, we shift each letter in “Hello World” left by 2 letters. ‘H’ → ‘G’ → ‘F’, ‘e’ → ‘d’ → ‘c’, and so on. After capitalizing we get the result FCJJM UMPJB. This process is repeated for the next two testcases.

Quartermaster

Problem Description: The quartermaster is the person on a ship responsible for determining how much money each pirate gets. Our Glorious Leader's ship is no different, but with so many crew members, the quartermaster needs your help in calculating how much money each person gets.

The way that the loot is divided is based on shares, which each contain the same amount of money (think stock shares). Each person gets at least 1 share of the total, but some more important jobs get more.

Your job is to print a formatted table displaying how much money each person is to get, given the "roster" of crew members and the total amount of money to be split.

Input Description: The first line contains N and M, separated by spaces. N represents the amount of money you have in total, and M represents the number of crew members.
The next M lines each have a crew member, represented by a name, job, and number of shares separated by spaces. The number of shares is guaranteed to be an integer.

Output Description: Print a formatted table showing the total amount of money split. For each crew member, show their name, job, number of shares, percentage of the total, and the amount of money.

Money and percentages should be represented in 2 decimal places. Check sample output. Column width should be based on the longest text in that column.

Sample Input:

```
500 3
Stroud Captain 3
Nutt Navigator 1
Castillo Quartermaster 1
```

Sample Output:

```
| Name     | Role          | # of Shares | % of Total | Money   |
| -------- | ------------- | ----------- | ---------- | ------- |
| Stroud   | Captain       | 3           | 60.00%     | $300.00 |
| Nutt     | Navigator     | 1           | 20.00%     | $100.00 |
| Castillo | Quartermaster | 1           | 20.00%     | $100.00 |
| Total    |               |             |            | $500.00 |
```

Sample Explanation:

The names are printed in the order that they were given in the input. The role is printed after the name. The number of shares is printed in the third column. The percent total for Stroud is calculated by doing 100 * 3 / (3 + 1 + 1) = 60.00%. The Money column for Stroud is calculated by doing 60.00 * 500 / 100 = $300.00. Repeat this process for the other names. The text in every column is left-aligned.

Navigating the Stroudonian Seas

Problem Description: Pirate Castillo has developed a system of pirating networks that connects key ports to easily get in contact with the supreme pirate leader, the Black Angus. The ports are numbered 1 to N. Today, Castillo has been called to meet the Black Angus. Castillo is currently on port A and the Black Angus is on port B. Each minute, both Castillo and the Black Angus MUST sail to an adjacent port. Unfortunately, being pirates, they pillage any ports they visit so any ports visited by any other pirate cannot be visited again. Find out if Castillo and the Black Angus can meet.

Input Description: The first line will contain T, the number of test cases. The first line of each test case will contain 4 integers, N, M, A, and B ($1 \leqslant N, M \leqslant 20$): the number of ports and the number of known routes on the map, the port Castillo is on and the port the Black Angus is on. The next M lines will each contain two integers, u_i and v_i, which represent that there is a bidirectional path from port u_i and port v_i.

Output Description: For each test case, print “We meet at last.” if they can meet and “Lost at sea.” if they can’t.

Sample Input:

```
3
3 2 1 3
1 2
2 3
4 3 1 4
1 2
2 3
3 4
3 3 2 3
1 2
2 3
1 3
```

Sample Output:

```
We meet at last.
Lost at sea.
We meet at last.
```

Sample Explanation:

Test Case 1: Both the Black Angus and Castillo can travel to port 2 in 1 minute.

Test Case 2: Try it all you want, they can't meet.

Test Case 3: Both the Black Angus and Castillo can travel to port 2 in 1 minute.

Why Did I Take German?

Problem Description: While visiting Atlantis, Captain Strauss is struggling to communicate with the locals. Atlanteans hate decimal numbers, instead preferring to use fractions. Unfortunately for Captain Strauss, he can only use decimals. Be a translator and translate Captain Strauss's decimals into the Atlanteans' fractions.

There will be N numbers that Captain Strauss needs to translate. Each number, a_i, will be between 1 and 10. These decimals will be translated into fractions of the form m/n where m and n are relatively prime positive integers. A translation is considered good if the first 3 numbers to the right of the decimal point of a_i match the first 3 numbers to the right of the decimal point of the decimal notation of the translated fraction and n is as small as possible.

Input Description: First line: N, the amount of numbers that need to be translated (1 ≤ N ≤ 1000)
Second line: a_i from i=1 to N, (1 ≤ a_i ≤ 10) the numbers to be translated separated by spaces

Output Description: N lines, 2 numbers per line: m and n, representing the fraction m/n

Sample Input:

```
5
1.47319283 2.33333 8.123456789 1.42587142587 5.5
```

Sample Output:

```
28 19
7 3
528 65
57 40
11 2
```

Sample Explanation:

1: 28/19 = 1.473684… matches the first 3 decimals of 1.47319283
2: 7/3 = 2.3333… matches the first 3 decimals of 2.33333

Rough Seas

Problem Description: While sailing at sea, Captain Nutt, the navigator on the High Seas, discovers an issue. To get to their destination, they must cross through an area of rough seas that will certainly slow them down.

The entire ocean can be represented as a coordinate system with a starting point of (x, y) and a destination of (m, n). The High Seas can travel in any direction. It is not limited to the lattice points. y=0 serves as the boundary between calm seas and rough seas: above y=0, the seas are calm, below y=0, the seas are rough. Captain Nutt knows that in calm seas, the ship travels at A mph while in rough seas, the ship travels at B mph. How fast can the High Seas reach its destination if it travels optimally (in hours)?

It is guaranteed that the starting point is above y=0 and the ending point is below y=0. Coordinates are written in miles.

Input Description:
First line: T, the number of test cases ($1 \leqslant T \leqslant 1000$)
Next T lines: x, y, m, n, A, B ($1 \leqslant x,y,m \leqslant 10^8$; $-10^8 \leqslant n \leqslant -1$; $1 \leqslant B<A \leqslant 10^8$, all are integers)

Output Description: T lines with the minimum time needed to reach the destination for each test case (in hours, rounded to 3 decimal points).

Sample Input:

```
3
3 4 7 -8 8 3
1 5 5 -2 2 1
0 5 0 -5 5 1
```

Sample Output:

```
3.303
5.111
6.000
```

Sample Explanation:

Testcase 1: If the Glorious Leader goes straight to (5.4180, 0), then to his destination, it will take 0.5843 hours to travel in the calm seas and 2.7183 hours to travel in the rough seas, which is a total of 3.303 hours.

Testcase 2: If the Glorious Leader goes straight to (4.4125, 0), then to his destination, it will take 3.0268 + 2.0845 = 5.111 hours.

Testcase 3: If the Glorious Leader goes straight to his destination, it will take 1.000 + 5.000 hours.

Hoist the Mizzenmast!

Problem Description: The Glorious Leader is preparing for his biggest mission yet. If he succeeds, then he will become the Glorious Leader of the entire world. The mission will consist of both members of the Glorious Leader's crew and Captain Pittman's crew.

Currently, the lineup contains N * K members of either the Glorious Leader's crew or Captain Pittman's crew, consisting of a lineup of length N repeated K times. Captain Pittman's crew will be represented by a 0 and the Glorious Leader's crew will be represented by a 1. For example, a lineup of "1100" repeated 3 times yields: "1100|1100|1100" (the | is only there to help you see the separation).

For the mission to succeed, the Glorious Leader will choose a contiguous substring of the lineup. Additionally, the chosen group must have an equal number of Pittman's and the Glorious Leader's crew members to ensure the power balance is equal. Find out how many ways the Glorious Leader can choose a lineup for the mission to succeed.

Input Description: The first line contains T, the number of test cases. The first line of each test case contains N (1 ⩽ N ⩽ 2 * 10^5), the number of pirates in the lineup, and K(1 ⩽ K ⩽ 10^9), the number of times the string will be repeated. The next line of each test case contains the original string of length N.

Output Description: For each test case, output the total number of ways the Glorious Leader can select a group modulo 10^9 + 7.

Sample Input:

```
2
5 3
01110
6 10
001101
```

Sample Output:

```
18
690
```

Sample Explanation:

The total lineup will be 011100111001110
Here are the substrings that cause the mission to succeed:
"01": Appears 3 times
"10": Appears 3 times
"1100": Appears 2 times
"0011": Appears 2 times
"1001": Appears 2 times
"001110": Appears 2 times
"011100": Appears 2 times
"00111001": Appears 1 time
"10011100": Appears 1 time