Perse Coding Team Challenge

2019

Round 2 Trios

There are 12 questions with 60 minutes allowed.

Three students will work together on three adjacent computers

A maximum of two year 10/11 students are allowed in any trio

This is meant to be fun – do make sure you take the odd pause to enjoy it! If you are in a younger year group then remember you can always use what you learn this year for another year to come.

SUBMISSION ESSENTIALS:

- Ensure you change the language of your coding editor on hackerrank once you start Q1 (top right of coding area) to the language you wish to use.
- Ensure your inputs contain NO PROMPTS; the only screen output should be as requested in the question.

INSTRUCTIONS:

- You must work in threes for Round 2 wherever possible (solo/duo if necessary)
- Your teacher will invigilate you for precisely 60 minutes and cannot discuss the problems.
- Each trio may use up to three computers for Round 2 and you may discuss problems/solutions quietly within your trio and so should be sitting next to each other, separated from other trios wherever possible.
- All code submissions must be made within the time allowed. Any team that submits code after their time has been ended by the invigilator will be automatically disqualified from the competition.
- You may write code directly into each hackerrank question page or you may write it in a development environment and copy/paste it across so long as these submissions are made within the time allowed.
- You may use the formal language reference online documentation for your language and you may also bring up to 10 A4 pages (20 sides) of notes/snippets into the competition. You should not have access to any other applications or resources (in particular no calculators are allowed either physical or digital).
- Questions may be attempted in any order, all questions are worth 10 points with each code submission tested against up to 10 test cases.
- You should have some rough paper and a pen handy.

NOTES:

- Ensure you are logged in to hackerrank on your machines before the time starts.
- The invigilating teacher will ask you for your team name and your hackerrank usernames (top right once logged in) for submitting your team details as soon as possible after the event. Your hackerrank usernames used should be confidential and not include your name for data protection: you can change your hackerrank username easily and should do so before the competition starts under settings → account settings.

Level 1 - Question 1

BOLT HOLE

As a nervous one-dimensional rabbit, you always stay on a straight line somewhere between two of your bolt hole positions.

Input three positive whole numbers, the location of your first bolt hole (0-1000), the location of your second bolt hole (0-1000) and your CURRENT location (0-1000). Output your minimum distance to safety were a fox attack to occur!

Input Format

- three lines of input, each line specifying a single whole number

Output Format

- a single positive number; the distance to your nearest bolt hole

Constraints

- each input will be a positive whole number n with $0 \leq n \leq 1000$

Example Input

```
64
12
21
```

Example Output

```
9
```

Example Explanation

You are currently at location 21 which is distance of 43 from one of your bolt holes (64 - 21) and a distance of 9 from your other bolt hole (21 - 12). So your nearest bolt hole is a distance of 9 away which is your output.

Level 1 - Question 2

MOONSHOT

It is 384,400 km from the Earth to the moon. The moon has a circumference of 10,921 km. A space ship is going to travel from Earth to the moon, make one full orbit of the moon and come straight back to Earth at a constant speed.

Input one positive whole number, a speed in meters per second, and output a time as a whole number of hours, always rounded UP, that the space ship takes for its journey.

Input Format

- one line of input, specifying a single whole number

Output Format

- a single positive number; the number of hours taken for the journey (rounded up)

Constraints

- all inputs are whole numbers between 0 and 10,000 inclusive

Example Input

```
1000
```

Example Output

```
217
```

Example Explanation

Moving at 1000 meters per second takes 216.59 hours, which is rounded up to 217.

Level 1 - Question 3

DINNER TIME

A packet of spaghetti feeds 5 people. Your saucepan is big enough to hold enough spaghetti for 7 people. Each batch of spaghetti takes 11 minutes to cook.

Input one positive whole number, which is the number of dinner guests.

Output two numbers on the same line, separated by a space. The first is the number of packets of spaghetti to buy and the second is the number of minutes of cooking time that will be needed.

Input Format

- one line of input, specifying a single whole number

Output Format

- two single positive whole numbers separated by a space

Constraints

- all inputs are whole numbers between 0 and 1000 inclusive

Example Input

```
12
```

Example Output

```
3 22
```

Example Explanation

For 12 guests, I will need 3 packets of spaghetti since each packet can serve 5 people. I will also need to cook 2 batches, each batch taking 11 minutes, giving 22 minutes in total.

Level 2 - Question 1

NAME PLAQUE

The office manager needs a tool to print name plates for new employees. They would like it to cater for people with different length names. They would also like it to have some nice formatting above and below that is formatted as shown here. For the new employee 'Joe Bloggs', their name plate should be printed using three lines of output as:

```
------------------
-=:|Joe Bloggs|:=-
------------------
```

There should be the correct pagination of hyphens on the row above and below the name: always matching the exact number of hyphens required for that name. The name should be title cased regardless of how it is entered.

Input Format

- two lines of input
- the first is the forename of the user, followed by the surname which may have been entered in any sequence of lower or upper case letters

Output Format

- three lines of text to create the intended name plate

Constraints

- all forenames/surnames will be a maximum length of 15 characters each

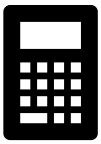
Example Input

```
jane
DOH
```

Example Output

```
----------------
-=:|Jane Doh|:=-
----------------
```

Level 2 - Question 2



SMALLEST SQUARE

Find the smallest positive square number that ends in a given sequence of digits. It is guaranteed that at least one perfect square ends in this sequence of digits.

Input Format

- one line of input consisting of a string of at least one and at most five digits

Output Format

- a single positive whole number: the smallest square number ending in the digits from the input

Example Input

```
0481
```

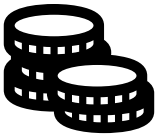
Example Output

```
2430481
```

Example Explanation

2430481 = 1559 × 1559 and is the smallest square number ending in 0481.

Level 2 - Question 3



COUNTING THE COFFERS

A bank keeps track of transactions. Your job is to write a program that prints out the final balance of an account after a series of transactions in whole pounds.

The first line will be the starting balance of the account. The second line of input will be an integer, *n*, that represents the number of transactions to keep track of.

You will then receive *n* transactions, each of which will either be a credit i.e. money coming into the account (CR) or a debit i.e. money leaving the account (DR). The account has a very large overdraft available (which won't be exceeded) so can go into as much debt as required. The program should then display the final balance as a credit or debit using the same notation.

Input Format

- the first line will start with CR or DR followed by a NON-NEGATIVE INTEGER separated by a space
- the second line will be an integer representing the number of transactions to follow
- all transactions will be of the same format as the first line

Output Format

- if the account is 0 then the account is considered in credit
- display CR or DR followed by a space and then a NON-NEGATIVE INTEGER for the account balance

Example Input

```
CR 350
3
DR 12
DR 150
CR 1200
```

Example Output

```
CR 1388
```

Level 3 - Question 1

WARSHIPS

A salvo of shots will be fired on to your game map (6x6) shown below which contains the following ships: Aircraft carrier (5 units length) marked A, Destroyer (4 units length) marked D, Submarine (3 units length) marked S and Patrol Boat (2 units length) marked P.

```
      x
    123456
  1 -A----
  2 -ADDDD
y 3 -A----
  4 -A---S
  5 -A---S
  6 PP---S
```

You must output a status report after the end of these shots in the exact form:
UNHARMED:<x>
HIT BUT AFLOAT:<y>
HIT AND SUNK:<z>
...where <x>, <y> and <z> represent the number of ships in that category.

Input Format

- the first line of the input contains a single integer $1 \leq n \leq 36$ representing the number of shots that will be made at your game board.
- each of the following n lines describes one shot: the x coordinate followed by a space, followed by the y coordinate of the shot.

Output Format

- A three-line status report detailing the number of your ships in each of the three categories in the format shown above.

Constraints

- All shots will be of the form $x\ y$ where x and y are whole numbers with $1 \leq x, y \leq 6$

Example Input

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

Example Output

```
UNHARMED:1
HIT BUT AFLOAT:2
HIT AND SUNK:1
```

Level 3 - Question 2

UPGRADED PARKING METER

The parking meter from the first round of this competition has been upgraded. Now, if you overpay for your parking it will give you change. It still charges 60p per half-hour and will give you as many half-hours parking as possible for your money.

For example, if you pay £3.17, you will get 5 half-hours of parking (so 150 minutes) and 17p change. The meter gives you your change in the most efficient way, i.e. using the smallest number of coins possible. Write a program to determine how many coins you will receive as change.

You can assume that the parking meter has an infinite supply of each of the £2, £1, 50p, 20p, 10p, 5p, 2p and 1p coins.

Input Format

- each line of coin input contains one positive whole number denoting the value of a coin in pence, which will be one of 200, 100, 50, 20, 10, 5, 2, 1
- the input is terminated by a line containing -1

Output Format

- one whole number: the minimum number of coins you will receive as change

Constraints

- if no change is needed, output 0

Example Input

```
200
50
50
5
1
1
10
-1
```

Example Output

```
3
```

Example Explanation

You can give 17p change in three coins (one 10p, one 5p and one 2p), but not in two or fewer coins.

Level 3 - Question 3

HOMEWORK PLANNER

Rebecca woke up early in the morning and she realised that she still needed to complete her homework in *n* different subjects before going to school! She has little time left and she knows she is going to get a sanction for each individual missing homework. Rebecca wants to make the best use of her limited time and minimise the number of sanctions she gets.

Input Format

- the first line of the input contains one whole number $1 \leq n \leq 30$, for the number of pieces of homework Rebecca needs to do
- the second line of the input contains one whole number, $1 \leq t \leq 10{,}000$, for the time (in seconds) that Rebecca has left before school starts
- each of the following *n* lines describes one of these homeworks; each containing one whole number integer, not exceeding 10,000 for the number of seconds this homework would take to complete

Output Format

- a single whole number: the smallest number of sanctions that Rebecca can get

Example Input

```
3
3600
2000
1700
1600
```

Example Output

```
1
```

Example Explanation

Rebecca can guarantee getting at most one sanction by completing the first and the third homework, or the second and third homework. She does not have time to complete all three pieces of work.

Level 4 - Question 1

RESTRICTED DIGITS

Given a set of non-zero digits (1-9), find the sum of all the numbers that can be written using all of these digits.

Input Format

- the input will contain a string of several (at least two) non-zero digits
- a digit may appear more than once

Output Format

- a single whole number – the sum of all the distinct numbers that can be made using all the digits in the input.

Constraints

- a minimum of two and a maximum of twelve digits will be given as input
- it is guaranteed that the answer does not exceed 4×10^{18}

Example 1 Input	Example 1 Output
317	2442

Example 1 Explanation

317 + 371 + 137 + 173 + 713 + 731 = 2442

Example 2 Input	Example 2 Output
388	2109

Example 2 Explanation

388 + 838 + 883 = 2109
Note that there are only three possibilities to sum here because we only sum the distinct possibilities.

Level 4 - Question 2

CHRISTMAS LIGHTS MEGA-TANGLE

Jason had a huge Christmas tree with many lights for the holiday season. The tree was lit by several chains of lights. Now, as the Christmas season is over, he needs to take these lights off the tree. Unfortunately, these got tangled and Jason doesn't even know how many chains he had or how long they were.

Write a program, which reads which lights are connected by cables and determines the length of the longest light chain.

The lights are numbered from 1 to n. The lights form several disjoint chains: there are no "branching" cables. Each light is directly connected to at least one and at most two other lights.

Input Format

- the first line of the input contains one integer $2 \leq n \leq 50{,}000$ for the number of lights
- the second line of the input contains one integer $1 \leq m \leq 50{,}000$ for the number of connections between lights
- each of the following m lines contains two integers $1 \leq i < j \leq n$; this means that lights #i and #j are directly connected by a cable

Output Format

- one positive whole number – the number of lights in the longest chain

Example Input

```
8
6
1 3
1 4
6 7
2 5
5 8
4 6
```

Example Output

```
5
```

Example Explanation

We have two chains here: 3 – 1 – 4 – 6 – 7 and 2 – 5 – 8. The longer chain is five lights long.

Level 4 - Question 3

VAULTS AND DRAGONS

A group of adventurers want to get their hands on a great treasure buried deep in a dungeon. The dungeon consists of many vaults, connected to each other with tunnels. Some of the vaults are guarded by dragons – the adventurers would obviously want to avoid passing through these as much as possible.

Write a program to determine the MINIMUM number of dragons the adventurers need to encounter on their quest to get the treasure. All tunnels can be traversed in both directions. It is guaranteed that it is possible to reach the treasure from the dungeon entrance through the tunnels provided.

Input Format

- the first line of the input contains one positive integer $2 \leq n \leq 10{,}000$ – the number of vaults in the dungeon
- the second line of the input contains a string of *n* characters; the *i*th character in this string describes vault *i* where 'D' means it contains a dragon, 'T' means it contains the treasure and 'E' means it is empty
- the third line of the input contains one positive integer $1 \leq m \leq 30{,}000$ – the number of tunnels in the dungeon
- each of the following *m* lines describes one of the tunnels, giving two positive integers (vaults it connects to each other) separated by a single space

Output Format

- a single whole number: the minimum number of dragons that need to be encountered when passing from the entrance to the treasure (inclusive)

Constraints

- vault 1 is always the entrance to the dungeon
- exactly one vault contains the treasure

Example Input

```
7
EDDDTDE
9
1 2
1 3
2 3
2 6
3 4
4 5
4 6
5 7
6 7
```

Example Output

```
2
```

Example Explanation

There are many ways to the treasure, for example
1 -> 2 -> 6 -> 7 -> 5
or 1 -> 3 -> 4 -> 5,
but all of them require passing at least two dragons.

Perse Coding Team Challenge

is run as a free cross-school community competition

and is written and maintained by teachers from several schools,

working in partnership with

The Perse School, Cambridge.

www.perse.co.uk

The competition and prize money is kindly sponsored by Dr David Braben OBE FREng.

after whom the winner's cup is named.

Please see our competition website https://persecoding.net to find out more.