

Perse Coding Team Challenge

2019



Round 1 Pairs

There are 10 questions with 40 minutes allowed.

Two students will share one computer

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 pairs for Round 1 wherever possible (solo in exceptional cases)
- Your teacher will invigilate you for precisely 40 minutes and cannot discuss the problems.
- You may only use one computer for Round 1 and you may discuss problems/solutions quietly within your pair and so should be sitting next to each other, separated from other pairs 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; Questions 1-6 are worth three points with three test cases, Q7-10 questions are worth five points with five 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 username 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.



Question 1

NAME CHANGE

What name would you like to have been given? In this challenge you will input two names (on separate lines) which will represent your current name and what you would like to be called. You will then output an official name-changing statement of the form:



Hello <name1>, your name is now <name2>.

Input Format

2 lines of input, each line specifying a name.

Output Format

Output a single line name changing greeting in the form given above.

Constraints

- All inputs are strings between 1 and 30 characters inclusive.
- You must pay careful attention to capitalisation and punctuation in your output to exactly match that requested.

Example Input

```
Maeve  
Elise
```

Example Output

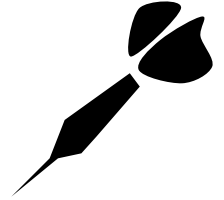
```
Hello Maeve, your name is now Elise.
```



Question 2

DARTS

The maximum score you can achieve with three darts is 180. Input three whole numbers each representing a dart score from a throw and output how much short of the maximum the total of the darts thrown was.



Input Format

Three lines of input as positive whole numbers between 1 and 60 inclusive.

Output Format

A single positive whole number

Example Input

```
60
57
5
```

Example Output

```
58
```

Example Explanation:

The score from the three darts was $60 + 57 + 5 = 122$ which is 58 short of the maximum 180 so the output is 58.



Question 3

HELP YOURSELF

If you are offered a slice of cake and you are hungry then chances are you might pick the one that looks largest. In this challenge you will be given the volume (in cubic centimetres) of two slices of cake, SLICE 1 and SLICE 2 in that order.



Input Format

2 lines of input, each line specifying a single whole number representing the volume of that slice of cake.

Output Format

Output either SLICE 1 or SLICE 2 to indicate which is the largest or output SAME if they are the same size.

Constraints

- all inputs are whole numbers between 1 and 1000 inclusive

Example Input

```
417
```

```
428
```

Example Output

```
SLICE 2
```



Question 4

EQUALLY SPACED?

A simple sequence can be spotted because it changes by the same number each time. For example 12 16 20 24 changes by 4 each term and 17 14 11 8 changes by -3 each term.



Input four whole numbers on four lines and output NO (uppercase) if it is not a simple sequence or otherwise output the number that it changes by each time.

Input Format

Four lines of input giving four whole numbers.

Output Format

Output NO (all uppercase) or a single whole number.

Example Input

```
7
2
-3
-8
```

Example Output

```
-5
```

Constraints

- all inputs are whole numbers between -1000 and 1000 inclusive

Example Explanation:

The sequence goes down in fives each time so the output is -5.



Question 5

FOX IN A HOLE

There are 7 holes labelled from 1 to 7. A fox is hiding in one of the holes. Your job is to report the whereabouts of the fox which is shown as an F. An empty hole is represented by a capital letter O.



Input Format

A seven-letter sequence of characters consisting of six Os and 1 F (all upper-case).

Output Format

A positive integer from 1 to 7 inclusive to indicate the fox's hiding position where hole 1 is on the left and hole 7 on the right.

Example Input

```
OOFOOOO
```

Example Output

```
3
```



Question 6

LEMONADE STAND



John went shopping to buy lemons and sugar to make lemonade but he forgot to check his ingredients. He NEEDS at least 20 lemons and 3kg of sugar.

Input the number of lemons he bought followed by the number of kgs of sugar he bought (both will be whole numbers) and then display one of the following options:

- A if John has enough ingredients
- B if he needs more lemons
- C if he needs more sugar
- D if he needs both more sugar and lemons.

Input Format

2 lines of input giving 2 positive whole numbers specifying the number of lemons and the total kgs of sugar purchased.

Output Format

A single uppercase letter: A, B, C or D

Constraints

- Both inputs will be a positive integers between 1 and 100 inclusive

Example Input

```
22  
2
```

Example Output

```
C
```



Question 7

FOLLOW MY SISTER



Jack has an older sister Amy and he often likes to be a bit annoying. Their mum is coming home with four cakes, one lemon, one chocolate, one caramel and one coffee. If Jack gets asked first he will say he wants chocolate but if Amy gets asked first then he will say he wants the same cake that Amy picks unless Amy picks coffee (which he hates) in which case he will say chocolate.

The first input will be 1 or 2 where 1 indicates that Jack gets asked first and 2 indicates that Jack gets asked second.

If the first input is 2 then (and only then), there will be a second input which indicates Amy's response which will be either the word lemon or the word chocolate or the word caramel or the word coffee.

Output what Jack will ask for in this circumstance again as a single word either lemon/chocolate/caramel/coffee.

Input Format

Up to two lines of input:

- The first line will be the digit 1 or the digit 2 to indicate whether Jack gets asked first or second.
- The second line (if present) will be a single word to indicate Amy's preference: lemon/chocolate/caramel/coffee (all lower-case)

Output Format

Output a single word to indicate Jack's request: lemon/chocolate/caramel/coffee

Example Input

```
2
lemon
```

Example Output

```
lemon
```

Example Explanation:

Jack is being asked second (because of the 2) so Amy has gone first and has requested lemon. Jack will copy Amy and say he also wants lemon.



Question 8

PRINT A RUG

Kardeesh wants to 'print' out some rugs using her new 3D printer. Each rug will be made out of stitched squares. A rug will be either 3, 4 or 5 squares wide and the length of the rug will also be entered by the user. In this simulation she will use the character @ to represent one square of the rug.



Input Format

2 lines of input giving 2 positive whole numbers specifying the width and the height of the rug.

Output Format

A 2D version of the rug using @ characters only over multiple lines.

Constraints

- the first input will be between 3 and 5 inclusive
- the second input will be a positive integer between 1 and 100 inclusive

Example Input

```
5
3
```

Example Output

```
@@@@@
@@@@@
@@@@@
```



Question 9

SEASON JUMPING

We all know, of course, that our seasons go in the cycle spring - summer - autumn - winter and then back to spring again. Input a current season and then the number of seasons to jump and output the season that it will then be.



Input Format

2 lines of input, the first line specifying one of the four seasons spring / summer / autumn / winter and the other a positive integer for the number of seasons to jump.

Output Format

A single word indicating the season that it will then be in lowercase (spring/summer/autumn/winter).

Constraints

- the second input will be a positive integer less than 100

Example Input

```
summer
```

```
3
```

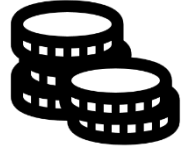
Example Output

```
spring
```



Question 10

CAR PARKING METER



You rummage through the coins in your pocket looking for the cash you need for the parking meter which costs 60p per half-hour. Input the value of your coins in pennies, line by line, with a -1 indicating your pocket is empty. Output the number of minutes that you will be able to park for. [Note: you can only park in units of half-hours, nothing less.]

Input Format

Multiple lines of positive whole numbers (pennies), terminated by a -1 as an end-of-data signal.

Output Format

A whole number representing the number of minutes you can park for.

Constraints

- coin inputs will consist of 1, 2, 5, 10, 20, 50, 100, 200
- a maximum of 10 coins will be provided
- an input of -1 will be used to indicate the end of input

Example Input

```
50
20
100
10
2
-1
```

Example Output

```
90
```

Example Explanation:

You have £1.82 (182p) in total from $50+20+100+10+2$ so that is three half-hours i.e. 90 minutes of parking.