You can view this report online at : https://www.hackerrank.com/x/tests/1757195/candidates/58287094/report

Full Name:	Instructor
Email:	aisha.batool@sse.habib.edu.pk
Test Name:	**CS101 - LW15 - Fall 23**
Taken On:	24 Nov 2023 23:05:09 PKT
Time Taken:	6 min 3 sec/ 180 min
Student Roll Number:	02163
Section:	T1
Invited by:	Aisha
Skills Score:	
Tags Score:	CS101 10/10 Lists 10/10 NestedLists 10/10 Tuples 10/10

100%

50/50

scored in **CS101 - LW15 - Fall 23** in 6 min 3 sec on 24 Nov 2023 23:05:09 PKT

Recruiter/Team Comments:

No Comments.

Question Description	Time Taken	Score	Status
Q1 Get Positions > Coding	42 sec	10/ 10	✓
Q2 Last name first > Coding	45 sec	10/ 10	✓
Q3 Dude, Where's My Robot? - Tuples > Coding	1 min 45 sec	10/ 10	✓
Q4 Loan Repayment Strategy > Coding	1 min	10/ 10	✓
Q5 Frog Race > Coding	1 min 36 sec	10/ 10	✓
Q6 Difficulty Meter > Multiple Choice	7 sec	0/ 0	✓

QUESTION 1

Correct Answer

Score 10

Get Positions > Coding

CS101 NestedLists Lists Tuples

QUESTION DESCRIPTION

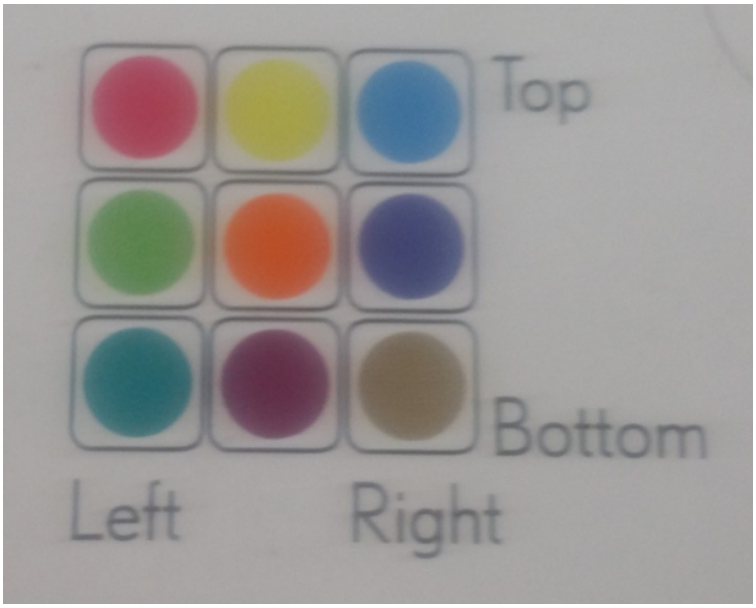
Problem

You are tasked with creating a Python program to handle a colorful cube board. The board is represented as a string, where each character denotes the color of a cube. Your goal is to write two functions: `read_board` and `get_positions`.

The `read_board` function will take a string as a parameter and return it in the following format. A board is a list containing 3 lists, one for each row. The list at index 0 corresponds to the top row, index 1 to the middle row, and index 2 to the bottom row. The list for each row contains 3 strings, each representing the color of the cubes in the row from left to right.

The `get_positions` function will take the `board` and `color` as arguments and returns a list of all positions (tuple containing the row, col) at which this color is on in the board.

Sample




```
>> board = read_board("Pink Yellow LightBlue Green
Orange DarkBlue Teal Purple Gold")
>> board
[["Pink", "Yellow", "LightBlue"], ["Green",
"Orange", "DarkBlue"], ["Teal", "Purple", "Gold"]]

>> get_positions(board, "Yellow")
   [(0,1)]
>> get_positions(board, "Red")
   []
```

Input and Output

Input and Output is handled by Hackerrank.

Constraints

You can't use built-in functions to find the index of a color such as find or index.

INTERVIEWER GUIDELINES

```
def read_board(str):
    '''read_board() -> list of lists.

    Reads a sequence of 9 space separated colors from str and
    returns them arranged as a board.
    '''
    board = str.split()
    return [board[:3], board[3:6], board[6:]]

def get_positions(board, color):
    '''get_positions(list, str) -> list of pairs
```

```
    Returns all positions of color on board. Each position is
    represented as a pair (row,col) with row and col 0-indexed row
    and column numbers.
    '''
    pos = []
    for i in range(3):
        for j in range(3):
            if board[i][j] == color:
                pos.append((i,j))
    return pos
```

CANDIDATE ANSWER

Language used: **Python 3**

```
def read_board(str):
    '''read_board() -> list of lists.

    Reads a sequence of 9 space separated colors from str and
    returns them arranged as a board.
    '''
    board = str.split()
    return [board[:3], board[3:6], board[6:]]

def get_positions(board, color):
    '''get_positions(list, str) -> list of pairs

    Returns all positions of color on board. Each position is
    represented as a pair (row,col) with row and col 0-indexed row
    and column numbers.
    '''
    pos = []
    for i in range(3):
        for j in range(3):
            if board[i][j] == color:
                pos.append((i,j))
    return pos

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	1	0.0149 sec	9.37 KB
Testcase 1	Easy	Sample case	Success	1	0.0142 sec	9.37 KB
Testcase 2	Easy	Hidden case	Success	1	0.018 sec	9.34 KB
Testcase 3	Easy	Sample case	Success	1	0.0173 sec	9.25 KB
Testcase 4	Easy	Hidden case	Success	1	0.0143 sec	9.24 KB
Testcase 5	Easy	Hidden case	Success	1	0.0141 sec	9.62 KB
Testcase 6	Easy	Sample case	Success	1	0.0862 sec	9.55 KB
Testcase 7	Easy	Hidden case	Success	1	0.0159 sec	9.48 KB
Testcase 8	Easy	Hidden case	Success	1	0.0161 sec	9.41 KB
Testcase 9	Easy	Hidden case	Success	1	0.1093 sec	9.62 KB

No Comments

QUESTION 2

Correct Answer

Score 10

Last name first › Coding

QUESTION DESCRIPTION

Challenge

Write a function named `last_name_first` that accepts a single parameter, `t`, which is passed a list of tuples. Each tuple contains a name in parts (first name, middle name, last name). Your function should modify each name so that the last name appears first in the tuple.

Note

This function modifies a list in place and, as such, should not return any useful value.

Sample interaction

```
>>> t = [('Ahmed', 'Dawood'), ('Haroon', 'Hussain', 'Fawad', 'Rasheed'),
('Muhammad', 'Faisal', 'Amin')]
>>> last_name_first(t)
>>> print(t)
[('Dawood', 'Ahmed'), ('Rasheed', 'Haroon', 'Hussain', 'Fawad'), ('Amin',
'Muhammad', 'Faisal')]
```

Input/Output

Input and output will be handled by HackerRank.

Constraints

`t` is a list of tuples, where each tuple has one or more strings in it.

INTERVIEWER GUIDELINES

```
def last_name_first(names):
    for p in range(len(names)):
        name = names[p]
        name = (name[-1], ) + name[:-1]
        names[p] = name
```

CANDIDATE ANSWER

Language used: **Python 3**

```
1 def last_name_first(names):
2     for p in range(len(names)):
3         name = names[p]
4         name = (name[-1], ) + name[:-1]
5         names[p] = name
6
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	1	0.0147 sec	9.42 KB
Testcase 1	Easy	Sample case	Success	1	0.015 sec	9.31 KB
Testcase 2	Easy	Sample case	Success	1	0.0144 sec	9.48 KB
Testcase 3	Easy	Sample case	Success	1	0.0148 sec	9.26 KB
Testcase 4	Easy	Sample case	Success	1	0.0146 sec	9.37 KB
Testcase 5	Easy	Hidden case	Success	1	0.0178 sec	9.24 KB

Testcase 6	Easy	Hidden case	Success	1	0.0169 sec	9.3 KB
Testcase 7	Easy	Hidden case	Success	1	0.0137 sec	9.3 KB
Testcase 8	Easy	Hidden case	Success	1	0.0758 sec	9.42 KB
Testcase 9	Easy	Hidden case	Success	1	0.0164 sec	9.38 KB

No Comments

QUESTION 3

Correct Answer

Score 10

Dude, Where's My Robot? - Tuples › Coding

QUESTION DESCRIPTION

Your neighbor's garden robot was hacked and moved from its charging pod where you had last left it. They cannot find the robot and need the grass trimmed right now for a party with their committee friends this evening. Luckily, they know of your skill with computers. You manage to retrieve the last set of instructions sent to the robot. Now you just have to figure out how far away the robot has moved.

The set contains *n* instructions. Each instruction is of the form *direction distance*. On receiving this instruction, the robot moves *distance* in *direction*. The robot executes the instructions one after the other.

Given the list of instructions, you want to find out the distance that that the robot has moved.

Input

The input is a list of tuples where each tuple contains the *direction* (where direction is one of UP, DOWN, RIGHT, and LEFT) and *distance* of type integer

Output

Output the total distance that the robot has moved from its original distance once it has carried out the given instructions. Round the distance up to the nearest integer. You should write your code inside a function called `total_distance` which takes the list of tuples as an input.

Examples

Input

```
[('UP', '5'), ('DOWN', '3'), ('LEFT', '3'), ('RIGHT', '2')]
```

Output

```
3
```

Note

The robot moved 5 up, then 3 down, then 3 left, and then 2 right. Reluctantly, the robot is 2 up and 1 left from its original position. The distance is `math.sqrt(2*2 + 1*1)` which, rounded up, is 3.

Input

```
[('RIGHT', 1), ('DOWN', 0), ('LEFT', -2), ('DOWN', -4)]
```

Output

```
5
```

Note

The robot is 3 right and 4 up from its position. The distance is `math.sqrt(3*3 + 4*4)` which, rounded up, is 5.

Input

```
[('UP', 0)]
```

Output

```
0
```

Note

The robot has not moved. The distance is 0.

INTERVIEWER GUIDELINES

Solution

```
import math
def total_distance(lst):
    x = y = 0
    for i in lst:
        d, s = i[0], i[1]
        if d == 'UP':
            y += s
        elif d == 'DOWN':
            y -= s
        elif d == 'RIGHT':
            x += s
        elif d == 'LEFT':
            x -= s
    return math.ceil(math.sqrt(x*x + y*y))
```

CANDIDATE ANSWER

Language used: **Python 3**

```
1  import math
2  def total_distance(lst):
3      x = y = 0
4      for i in lst:
5          d, s = i[0], i[1]
6          if d == 'UP':
7              y += s
8          elif d == 'DOWN':
9              y -= s
10         elif d == 'RIGHT':
11             x += s
12         elif d == 'LEFT':
13             x -= s
14     return math.ceil(math.sqrt(x*x + y*y))
15
16
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 0	Easy	Sample case	Success	1	0.0309 sec	9.5 KB
TestCase 1	Easy	Hidden case	Success	1	0.0847 sec	9.25 KB
TestCase 2	Easy	Hidden case	Success	1	0.0185 sec	9.5 KB
TestCase 3	Easy	Hidden case	Success	1	0.016 sec	9.62 KB
TestCase 4	Easy	Hidden case	Success	1	0.0182 sec	9.73 KB
TestCase 5	Easy	Hidden case	Success	1	0.0152 sec	9.32 KB
TestCase 6	Easy	Hidden case	Success	1	0.016 sec	9.56 KB
TestCase 7	Easy	Hidden case	Success	1	0.0155 sec	9.81 KB
TestCase 8	Easy	Hidden case	Success	1	0.0171 sec	9.63 KB
TestCase 9	Easy	Hidden case	Success	1	0.016 sec	9.54 KB

No Comments

QUESTION 4

Correct Answer

Score 10

Loan Repayment Strategy › Coding

QUESTION DESCRIPTION

On the advice of your relative from the stock market, you have invested in stock in the hope to eventually pay off your Habib loan. Your relative sends you daily updates on your stocks in the following form.

Purchase Date	Purchase Price	Shares	Symbol	Current Price
26 Aug 2019	43.50	100	HU	47.02
27 Aug 2019	22.07	500	PTI	19.11
30 Oct 2019	51.98	200	JHR	50.14
28 Nov 2019	137.92	50	WTF	150.28

You want to find out your current earnings from this information.

In the table above, each share of *HU* is at a *profit* of `47.02 - 43.50 = 3.52`. As you have 100 shares of *HU*, your profit is `100 * 3.52 = 352`. Similarly, with *PTI* you are at a *loss* of `500 * (22.07-19.11) = 1480`. Your JHR stocks are at a *loss* of `200 * (51.98-50.14) = 268` and and your WTF stocks are at a *profit* of `50 * (150.28-137.92) = 618`. Your total profit is `352 -1480 - 268 + 618 = -778`.

Function Description

Complete the function *compute_profit* in the editor below. It returns the total profit from given stock information which is provided as a list of tuples. Each tuple contains the following items in the given order. The type of each item is shown.

- <purchase_date> : int
- <purchase_price> : float
- <shares> : int
- <symbol> : str
- <current_price> : float

Constraints

- The argument contains at least 1 tuple.
- All tuples in the list follow the format described above.

▼ Input Format For Custom Testing

The input consists of a single line which contains all information of all stocks in a single line delimited by a space character. The output is a single numeric value indicating the total profit. The input is read and passed to your function and your function's return value is printed by the program.

▼ Sample Case 0

Sample Input For Custom Testing

```
25-Jan-2001 43.50 25 CAT 92.45 25-Jan-2001 42.80 50 DD 51.19 25-Jan-2001
42.10 75 EK 34.87 25-Jan-2001 37.58 100 GM 37.58
```

Sample Output

```
1101
```

Explanation

The input contains information of 4 stocks. The name and corresponding profit from each are as follows.

- CAT : `25 * (92.45 - 43.5) = 1223.75`
- DD : `50 * (51.19 - 42.8) = 419.5`
- EK : `75 * (34.87 - 42.1) = -542.25`
- GM `100 * (37.58 - 37.58) = 0`

The total profit is therefore `1223.75 + 419.5 - 542.5 + 0 = 1101`

INTERVIEWER GUIDELINES

Solution

```
import math
def compute_profit(stock_info):
    profit = 0
    for _, cost, qty, _, price in stock_info:
        profit += qty * (price - cost)
    return profit
```

CANDIDATE ANSWER

Language used: **Python 3**

```
1 import math
2 def compute_profit(stock_info):
3     profit = 0
4     for _, cost, qty, _, price in stock_info:
5         profit += qty * (price - cost)
6     return profit
7
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	1	0.0146 sec	9.91 KB
Testcase 1	Easy	Hidden case	Success	1	0.0457 sec	9.61 KB
Testcase 2	Easy	Hidden case	Success	2	0.0152 sec	9.66 KB
Testcase 3	Easy	Hidden case	Success	2	0.0186 sec	9.73 KB
Testcase 4	Easy	Hidden case	Success	2	0.0142 sec	9.69 KB
Testcase 5	Easy	Hidden case	Success	2	0.0809 sec	9.77 KB

No Comments

QUESTION 5

Correct Answer

Score 10

Frog Race › Coding

QUESTION DESCRIPTION

Two frogs named Frog Prime and Frogatron are in a well of depth 1000 meters. They decide to race each other to the top to see who gets out first. Each frog can jump up the wall of the well by a certain height but slips down a certain distance due to wetness on the wall. The frogs are not equally fit so the distance that they can jump up and end up sliding down is different from each other.

Frog Prime can jump `prime_up` meters at a time but slides down `prime_down` meters every time. Frogatron can jump `tron_up` meters at a time but slides down `tron_down` meters every time.

Both frogs take each jump at the same time and keep jumping till one of them clears the well and wins.

Your program should input the values, `prime_up`, `prime_down`, `tron_up`, and `tron_down` and then pass them to a function, `race`, which simulates the race. The function maintains a list of the positions of both frogs. Each position is stored as a *tuple* whose first element is the height of Frog Prime from the bottom of the well and second element is the height of Frogatron from the bottom of the well. As soon as a frog wins, the function prints the list, declares the winner, and announces the number of jumps the winner made.

Following is the output for some sample values.

Case 1

Input

`prime_up` = 50, `prime_down` = 2, `tron_up` = 1000, `tron_down` = 4

Output

[(0, 0), (48, 'OUT')]
Frogatron wins in 1 turns!

Explanation

The output list contains 2 positions. Both frogs begin at 0 so the first element is (0,0). Frog Prime jumps up 50 and slides down 2, thus gaining 48. At the same time, Frogatron jumps up 1000 and clears the well. As the well has been cleared, Frogatron does not slide down. The corresponding tuple in the list shows Frog Prime's position and that Frogatron has cleared the well. The winner was decided in 1 jump and no further jumps occurred so the list has no further elements.

Case 2

Input

`prime_up` = 520, `prime_down` = 100, `tron_up` = 410, `tron_down` = 20

Output

[(0, 0), (420, 390), (840, 780), ('OUT', 'OUT')]
Tie in 3 turns!

Explanation

The output list contains 4 positions. Both frogs begin at 0 so the first element is (0,0). Frog Prime jumps up 520 and slides down 100, thus gaining 420. At the same time, Frogatron jumps up 410 and slides down 20, thus gaining 390. The second element in the list is therefore (420, 390). At the next jump, each frog gains the same amount as in the previous jump. The next tuple in the list is thus (820, 780). At the next jump, each frog exceeds 1000 and gets out. The result is a tie decided in 3 jumps.

Input

4 space separated integer values. These are values of `prime_up`, `prime_down`, `tron_up`, and `tron_down` respectively.

Constraints

- `prime_up`, `prime_down`, `tron_up`, and `tron_down` are of type *int*.
- `prime_up`, `prime_down`, `tron_up`, and `tron_down` are *> 0*.
- `prime_up` > `prime_down`
- `tron_up` > `tron_down`

INTERVIEWER GUIDELINES

Solution

```
def race(f1up, f1down, f2up, f2down):
    f1 = f2 = 0
    positions = [(0,0)]
    done = False
    while not done:
        f1 += f1up
        f2 += f2up
        jumps = len(positions)
        if f1 >= 1000:
            if f2 >= 1000:
```

```
                positions.append(('OUT', 'OUT'))
                winner = 'Tie in {} turns!'.format(jumps)
            else:
                f2 -= f2down
                positions.append(('OUT', f2))
                winner = 'Frog Prime wins in {} turns!'.format(jumps)
            print(positions)
            print(winner)
            done = True
        elif f2 >= 1000:
            f1 -= f1down
            positions.append((f1,'OUT'))
            winner = 'Frogatron wins in {} turns!'.format(jumps)
            print(positions)
            print(winner)
            done = True
        else:
            f1 -= f1down
            f2 -= f2down
            positions.append((f1,f2))

prime_up, prime_down, tron_up, tron_down = map(int, input().split())
race(prime_up, prime_down, tron_up, tron_down)
```

CANDIDATE ANSWER

Language used: **Python 3**

```
def race(f1up, f1down, f2up, f2down):
    f1 = f2 = 0
    positions = [(0,0)]
    done = False
    while not done:
        f1 += f1up
        f2 += f2up
        jumps = len(positions)
        if f1 >= 1000:
            if f2 >= 1000:
                positions.append(('OUT', 'OUT'))
                winner = 'Tie in {} turns!'.format(jumps)
            else:
                f2 -= f2down
                positions.append(('OUT', f2))
                winner = 'Frog Prime wins in {} turns!'.format(jumps)
            print(positions)
            print(winner)
            done = True
        elif f2 >= 1000:
            f1 -= f1down
            positions.append((f1,'OUT'))
            winner = 'Frogatron wins in {} turns!'.format(jumps)
            print(positions)
            print(winner)
            done = True
        else:
            f1 -= f1down
            f2 -= f2down
            positions.append((f1,f2))

prime_up, prime_down, tron_up, tron_down = map(int, input().split())
race(prime_up, prime_down, tron_up, tron_down)
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Medium	Sample case	Success	2	0.0139 sec	9.65 KB
Testcase 1	Medium	Sample case	Success	2	0.0146 sec	9.48 KB
Testcase 2	Medium	Sample case	Success	2	0.0534 sec	9.56 KB
Testcase 3	Medium	Sample case	Success	2	0.0152 sec	9.44 KB
Testcase 4	Medium	Sample case	Success	2	0.0277 sec	9.41 KB

No Comments

QUESTION 6

Correct Answer

Score 0

Difficulty Meter › Multiple Choice

QUESTION DESCRIPTION

On a scale of 1 to 5, with 1 being very easy and 5 being extremely challenging, how would you rate this worksheet?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- [x] 1
- [ ] 2
- [ ] 3
- [ ] 4
- [ ] 5

No Comments

PDF generated at: 24 Nov 2023 18:11:41 UTC