



Full Name: Thais Felipelli
Email: tfelipelli@gmail.com
Test Name: Tera Modulo 1 Assessment
Taken On: 1 Mar 2018 14:08:09 -03
Time Taken: 119 min 56 sec/ 120 min
Invited by: Luisa
Tags Score:

Algorithms 98/150
Arrays 50/50
Basic Programming 100/150
Brute Force 50/50
Classic 50/50
Conditionals 50/50
Core Skills 198/300
Easy 198/300
General Programming 100/150
Hashing 40/50
Logic 8/50
Loops 50/100
Math 8/50
Operators 50/50
Problem Solving 98/150
Strings 50/50

66%
198/300

scored in **Tera Modulo 1 Assessment** in 119 min 56 sec on 1 Mar 2018 14:08:09 -03

Candidate Feedback: more time for beginners

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	Merge Strings > Coding	23 min 40 sec	50/ 50	✓
Q2	Modify Prices > Coding	29 min 48 sec	40/ 50	✓
Q3	Count Duplicates > Coding	12 min 45 sec	50/ 50	✓
Q4	Puzzle > Coding	12 min 37 sec	50/ 50	✓
Q5	Chocolates > Coding	29 min 36 sec	8/ 50	✓
Q6	Cupcake Feast > Coding	11 min 4 sec	0/ 50	✗

QUESTION 1



Correct Answer

Score 50

Merge Strings > Coding

Easy

Strings

General Programming

Basic Programming

Core Skills

Classic

QUESTION DESCRIPTION

You are merging data from two sources connected to a network access point to create a new data packet.

You must merge strings *a* and *b*, and then return a single merged string. A merge operation on two strings is described as follows:

- Append alternating characters from *a* and *b*, respectively, to some new string, *mergedString*

mergedString.

- Once all of the characters in one of the strings have been merged, append the remaining characters in the other string to *mergedString*.

Function Description

Complete the function *mergeStrings* in the editor below. It has the following properties:

Parameters	Name	Type	Description
	a	String	Input String
	b	String	Input String
Return	The function must return the merged string.		

Constraints

- $1 \leq |a|, |b| \leq 25000$

▼ Input Format For Custom Testing

The locked stub code in your editor reads two strings, *a* and *b*, from stdin and passes them to your function.

▼ Sample Case 0

Sample Input For Custom Testing

```
abc
def
```

Sample Output

```
adbecf
```

Explanation

a = abc

b = def

Taking alternate characters from both the strings, we get adbecf

► Sample Case 1

CANDIDATE ANSWER

Language used: **Python 3**

```
1  # Complete the function below.
2
3
4  def mergeStrings(a, b):
5
6      merge = ""
7      la = len(a)
8      lb = len(b)
9      l = max(la, lb)
10
11     ca = 0
12     cb = 0
13
14     for c in range(l) :
15
16         if ca < la :
17             merge = merge + a[ca]
18             ca = ca + 1
19
20         if cb < lb :
21             merge = merge + b[cb]
22             cb = cb + 1
23
24     return merge
```

TESTCASE

TYPE

STATUS

SCORE

TIME TAKEN

MEMORY USED

TEST CASE	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Test Case 0	Easy	Success	1	0.01 sec	7.22 MB
Test Case 1	Easy	Success	1	0.01 sec	7.32 MB
Test Case 2	Medium	Success	1	0.01 sec	7.62 MB
Test Case 3	Medium	Success	4	0.01 sec	7.16 MB
Test Case 4	Medium	Success	10	0.01 sec	7.21 MB
Test Case 5	Medium	Success	11	0.0 sec	7.25 MB
Test Case 6	Hard	Success	11	0.02 sec	7.98 MB
Test Case 7	Hard	Success	11	0.02 sec	7.94 MB

No Comments

QUESTION 2



Correct Answer

Score 40

Modify Prices > Coding

Easy

Hashing

Algorithms

Core Skills

Problem Solving

QUESTION DESCRIPTION

Michael is a shop owner who keeps a list of the name and sale price for each item in the store's inventory. After each sale, his employees record the name and sale price of every item sold. Michael suspects his manager, Alex, of embezzling money and modifying the sale prices of some of the items. Write a program that finds the number of times Alex recorded an incorrect sale price.

Complete the `verifyItems` function in the editor. It has four parameters:

1. `origItems`: An array of n strings, where each `origItemsi` is the name of an item in inventory.
2. `origPrices`: An array of n floating-point numbers, where each `origPricesi` contains the original (correct) price of `origItemsi`.
3. `items`: An array of m strings containing the name of the items with sales recorded by Alex.
4. `prices`: An array of m floating-point numbers, where each `pricesj` contains the sale price recorded by Alex for `itemsj`.

The function must return an integer denoting the number of sale prices incorrectly recorded by Alex.

Input Format

The first line contains an integer, n , denoting the size of the `origItems` array.

Each line i (where $0 \leq i < n$) of the n subsequent lines describes `origItemsi`.

The next line contains an integer, n , denoting the size of the `origPrices` array.

Each line i of the n subsequent lines describes `origPricesi`.

The next line contains an integer, m , denoting the size of the `items` array.

Each line j (where $0 \leq j < m$) of the m subsequent lines describes `itemsj`.

The next line contains an integer, m , the size of the `prices` array.

Each line j of the m subsequent lines contains the price of `itemsj`.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq m \leq n$
- $1.00 \leq \text{origPrices}_i, \text{prices}_j \leq 100000.00$, where $0 \leq i < n$, and $0 \leq j < m$

Output Format

Return the number of items whose sale prices were incorrectly recorded by Alex.

▼ Sample Case 0

Sample Input

```
4
rice
sugar
wheat
cheese
4
16.89
56.92
20.89
345.99
2
rice
```

cheese
2
18.99
400.89

Sample Output

2

Explanation

Given the following:

- *origItems* = ["rice", "sugar", "wheat", "cheese"]
- *origPrices* = [16.89, 56.92, 20.89, 345.99]
- *items* = ["rice", "cheese"]
- *prices* = [18.99, 400.89]

The prices for *rice* and *cheese* do not match the original price list, so we return 2 as the number of incorrectly recorded sale prices.

► Sample Case 1

CANDIDATE ANSWER

Language used: **Python 3**

```
1 # Complete the function below.
2
3 def verifyItems(origItems, origPrices, items, prices):
4
5     dnm = 0
6
7     for index1, oitem in enumerate(origItems) :
8
9         for index2, item in enumerate(items) :
10
11             if oitem == item :
12                 if prices[index2] - origPrices[index1] != 0 :
13                     dnm = dnm + 1
14
15     return dnm
```

TESTCASE	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 0	Medium	✓ Success	1	0.01 sec	7.29 MB
TestCase 1	Easy	✓ Success	1	0.01 sec	7.23 MB
TestCase 2	Easy	✓ Success	1	0.01 sec	7.39 MB
TestCase 3	Easy	✓ Success	5	0.01 sec	7.33 MB
TestCase 4	Medium	✓ Success	5	0.01 sec	7.33 MB
TestCase 5	Medium	✓ Success	5	0.02 sec	8.04 MB
TestCase 6	Medium	✓ Success	6	0.05 sec	8.64 MB
TestCase 7	Medium	✓ Success	6	0.21 sec	9.66 MB
TestCase 8	Hard	✓ Success	10	6.76 sec	23.7 MB
TestCase 9	Hard	✗ Terminated due to timeout	0	10.0 sec	31.4 MB

No Comments

QUESTION 3



Correct Answer

Score 50

Count Duplicates > Coding

Easy

Brute Force

Algorithms

Arrays

Core Skills

Problem Solving

QUESTION DESCRIPTION

Given an array of integers, your task is to count the number of duplicate array elements. We define a duplicate (non-unique value) of the array to be an integer that appears at least twice in the array. For example, if given array is [1, 1, 2, 2, 2, 3, 4, 3, 9], then there are a total of 3 non-unique values in the array (i.e., 1, 2, and 3).

Function Description

Complete the function `countDuplicates` in the editor.

Parameters	Name	Type	Description
	numbers	Integer Array	Denotes the integer array whose non-unique integers need to be counted.
Return	The function must return an integer denoting the number of non-unique values in the numbers array.		

Constraints

- $1 \leq n \leq 1000$
- $1 \leq numbers_i \leq 1000$

▼ Input Format For Custom Testing

Locked stub code in the editor reads the following input from stdin and passes it to the function:

The first line contains an integer, n , denoting the size of the `numbers` array.

Each line i of the n subsequent lines (where $0 \leq i < n$) contains an integer describing the value of `numbersi`.

▼ Sample Case 0

Sample Input For Custom Testing

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

Sample Output

```
2
```

Explanation


$n = 8$ and `numbers = [1, 3, 1, 4, 5, 6, 3, 2]`. The integers 1 and 3 both occur more than once, so we return 2 as our answer.

► Sample Case 1

CANDIDATE ANSWER

Language used: **Python 3**

```
1 # Complete the function below.
2
3 def countDuplicates(numbers):
4     # Write your code here.
5
6     countd = 0
7     dnumbers = []
8     for index1, loop1 in enumerate(numbers) :
9         if loop1 not in dnumbers :
10             counter = 0
11             for index2, loop2 in enumerate(numbers) :
12                 if loop1 == loop2 :
13                     counter = counter + 1
14             if counter > 1 :
15                 countd = countd + 1
16                 dnumbers = dnumbers + [loop1]
17
18     return countd
```

TESTCASE	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 0	Easy	 Success	1	0.01 sec	7.17 MB

TESTCASE	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 1	Easy	✓ Success	1	0.01 sec	7.32 MB
TestCase 2	Easy	✓ Success	1	0.01 sec	7.21 MB
TestCase 3	Easy	✓ Success	7	0.01 sec	7.22 MB
TestCase 4	Medium	✓ Success	8	0.02 sec	8.17 MB
TestCase 5	Medium	✓ Success	8	0.05 sec	8 MB
TestCase 6	Hard	✓ Success	12	0.06 sec	8 MB
TestCase 7	Hard	✓ Success	12	0.06 sec	8.17 MB

No Comments

QUESTION 4



Correct Answer

Score 50

Puzzle > Coding

Easy

General Programming

Basic Programming

Operators

Loops

Conditionals

Core Skills

QUESTION DESCRIPTION

You are designing a poster which prints out numbers with a unique style applied to each of them. The styling is based on the number of closed paths or *holes* present in a given number.

The number of holes that each of the digits from 0 to 9 have are equal to the number of closed paths in the digit. Their values are:

- 1, 2, 3, 5, and 7 = 0 holes.
- 0, 4, 6, and 9 = 1 hole.
- 8 = 2 holes.

Given a number, you must determine the sum of the number of holes for each of its digits.

Function Description

Complete the function *countHoles* in the editor below. It has the following attributes:

Parameters	Name	Type	Description
	num	Integer	Input Integer
Return	The function must return an integer denoting the total number of holes in <i>num</i> .		

Constraints

- $1 \leq num \leq 10^9$

► Input Format For Custom Testing

► Sample Case 0

► Sample Case 1

CANDIDATE ANSWER

Language used: **Python 3**

```

1  # Complete the function below.
2
3  def countHoles(num):
4
5      holes = 0
6      for c in str(num):
7          if c in ["0", "4", "6", "9"] :
8              holes = holes + 1
9          elif c == "8" :
10             holes = holes + 2
11
12     return holes
13
14
```

TESTCASE	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 0	Easy	✓ Success	1	0.01 sec	7.25 MB
TestCase 1	Easy	✓ Success	1	0.02 sec	7.1 MB

TESTCASE	TYPE	✓	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 2	Easy	✓	Success	1	0.01 sec	7.25 MB
TestCase 3	Easy	✓	Success	6	0.01 sec	7.33 MB
TestCase 4	Easy	✓	Success	3	0.01 sec	7.16 MB
TestCase 5	Easy	✓	Success	3	0.01 sec	7.22 MB
TestCase 6	Easy	✓	Success	7	0.01 sec	7.2 MB
TestCase 7	Easy	✓	Success	7	0.01 sec	7.16 MB
TestCase 8	Easy	✓	Success	7	0.01 sec	7.18 MB
TestCase 9	Easy	✓	Success	7	0.01 sec	7.25 MB
TestCase 10	Easy	✓	Success	7	0.01 sec	7.18 MB

No Comments

QUESTION 5



Correct Answer

Score 8

Chocolates > Coding

Easy

Math

Logic

Algorithms

Core Skills

Problem Solving

QUESTION DESCRIPTION

Sam loves chocolates and starts buying them on the 1^{st} day of the year. Each day of the year, x , is numbered from 1 to Y . On days when x is odd, Sam will buy x chocolates; on days when x is even, Sam will not purchase any chocolates.

Complete the `calculate` function in the editor so that for each day N_i (where $1 \leq x \leq N \leq Y$) in array `arr`, the number of chocolates Sam purchased (during days 1 through N) is printed on a new line. This is a function-only challenge, so input is handled for you by the locked stub code in the editor.

Input Format

The `calculate` function takes an array of integers as a parameter.

The locked code in the editor handles reading the following input from `stdin`, assembling it into an array of integers (`arr`), and calling `calculate(arr)`.

The first line of input contains an integer, T (the number of test cases). Each line i of the T subsequent lines describes the i^{th} test case as an integer, N_i (the number of days).

Constraints

- $1 \leq T \leq 2 \times 10^5$
- $1 \leq N \leq 2 \times 10^6$
- $1 \leq x \leq N \leq Y$

Output Format

For each test case, T_i in `arr`, your `calculate` method should print the total number of chocolates Sam purchased by day N_i on a new line.

Sample Input 0

```
3
1
2
3
```

Sample Output 0

```
1
1
4
```

Explanation

Test Case 0: $N = 1$

Sam buys 1 chocolate on day 1 , giving us a total of 1 chocolate. Thus, we print 1 on a new line.

Test Case 1: $N = 2$

Sam buys 1 chocolate on day 1 and 0 on day 2 . This gives us a total of 1 chocolate. Thus, we print 1 on a new line.

Test Case 2: $N = 3$

Sam buys 1 chocolate on day 1 , 0 on day 2 , and 3 on day 3 . This gives us a total of 4 chocolates. Thus, we print 4 on a new line.

CANDIDATE ANSWER

Language used: **Python 3**

```

1  # Complete the function below.
2
3  def calculate(arr):
4
5      for days in arr:
6          counter = days
7          tc = 0
8          while counter > 0 :
9              if counter % 2 != 0 :
10                 tc = tc + counter
11                 counter = counter - 1
12
13         print(tc)
14

```

TESTCASE	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 0	Easy	✓ Success	0	0.01 sec	7.25 MB
TestCase 1	Easy	✓ Success	2	0.01 sec	7.23 MB
TestCase 2	Easy	✓ Success	2	0.02 sec	8.12 MB
TestCase 3	Easy	✓ Success	2	0.02 sec	8.04 MB
TestCase 4	Easy	✓ Success	2	1.61 sec	8.62 MB
TestCase 5	Easy	✗ Terminated due to timeout	0	10.0 sec	8.54 MB
TestCase 6	Easy	✗ Terminated due to timeout	0	10.0 sec	9.06 MB
TestCase 7	Easy	✗ Terminated due to timeout	0	10.0 sec	9.07 MB
TestCase 8	Easy	✗ Terminated due to timeout	0	10.0 sec	9.11 MB
TestCase 9	Easy	✗ Terminated due to timeout	0	10.0 sec	12.2 MB
TestCase 10	Easy	✗ Terminated due to timeout	0	10.0 sec	12.1 MB
TestCase 11	Easy	✗ Terminated due to timeout	0	10.0 sec	16.3 MB
TestCase 12	Easy	✗ Terminated due to timeout	0	10.0 sec	16.1 MB

No Comments

QUESTION 6



Wrong Answer

Score 0

Cupcake Feast > Coding

Easy

Loops

Basic Programming

General Programming

Core Skills

QUESTION DESCRIPTION

Sarah loves going to her favorite bakery, Zillycakes, with n dollars to buy cupcakes. Each cupcake has a flat cost of c dollars, and the store has a promotion where they give you 1 free cupcake for every m cupcake wrappers you trade in.

For example, if $m = 2$ and Sarah has $n = 4$ dollars that she uses to buy 4 cupcakes at $c = 1$ dollar apiece, she can trade in the 4 wrappers to buy 2 more cupcakes. Now she has 2 more wrappers that she can trade in for 1 more cupcake. Because she only has 1 wrapper left at this point and $1 < m$, she was only able to eat a total of 7 cupcakes.

Complete the `maximumCupcakes` function in your editor. It has 1 parameter: an array of strings named `trips`. Each string in `trips` contains the following three space-separated values:

1. An integer, n , denoting the amount of money (in dollars) Sarah can spend during that trip to Zillycakes.
2. An integer, c , denoting the cost of a cupcake in dollars.
3. An integer, m , denoting the number of wrappers that can be traded in for one new cupcake.

For each trip to Zillycakes in `trips`, your function must print an integer denoting the maximum number of cupcakes Sarah can eat on a new line.

Input Format

The locked stub code in your editor reads the following input from stdin and passes it to your function:

The first line contains an integer, t , denoting the number of trips Sarah makes to the store (i.e., the size of the `trips` array).

Each line i of the t subsequent lines contains *three* space-separated integers describing the respective n , c , and m values for one of Sarah's trips to the store. This corresponds to element

respective n , c , and m values for one of Sarah's trips to the store. This corresponds to element i in *trips*.

Constraints

- $1 \leq t \leq 10^3$
- $2 \leq n \leq 10^5$
- $1 \leq c \leq n$
- $2 \leq m \leq n$

Output Format

For each trip in *trips*, your function must print a single integer on a new line denoting the *maximum* number of cupcakes Sarah can eat during that trip.

Sample Input 1

```
3
10 2 5
12 4 4
6 2 2
```

Sample Output 1

```
6
3
5
```

Sample Input 2

```
2
8 4 2
7 2 3
```

Sample Output 2

```
3
4
```

Explanation

Sample Case 1

Sarah makes the following three trips to Zillycakes:

$$T = 3$$

In 1st test case : She spends 10 dollars on 5 cupcakes at 2 dollars apiece. She then eats them and exchanges all 5 wrappers to get 1 more cupcakes. We print the total number of cupcakes she ate, which is 6.

In 2nd test case : She spends 12 dollars on 3 cupcakes at 4 dollars apiece; however, she needs 4 wrappers to trade for her next cupcake. Because she only has 3 wrappers, she cannot purchase or trade for any more cupcakes. We print the total number of cupcakes she ate, which is 3.

In 3rd test case : She spends 6 dollars on 3 cupcakes at 2 dollars apiece. She then exchanges 2 of the 3 wrappers for 1 additional cupcake. Next, she combines her third leftover cupcake wrapper from her initial purchase with the wrapper from her trade-in to do a second trade-in for 1 more cupcake. At this point she has 1 wrapper left, which is not enough to perform another trade-in. We print the total number of cupcakes she ate, which is 5.

Sample Case 2

$$T = 2$$

Sarah makes the following three trips to Zillycakes

In 1st test case : She spends 8 dollars on 2 cupcakes at 4 dollars apiece. She then eats them and exchanges both of her wrappers for 1 more cupcake. At this point she has 1 wrapper left, which is not enough to perform another trade-in. We print the total number of cupcakes she ate, which is 3.

In 2nd test case : She spends 6 of her 7 dollars on 3 cupcakes at 2 dollars apiece. She then eats them and exchanges all three of her wrappers for 1 more cupcake. At this point she has 1 wrapper left, which is not enough to perform another trade-in. We print the total number of

cupcakes she ate, which is 4.

CANDIDATE ANSWER

Language used: **Python 3**

```
1 # Complete the function below.
2
3 def maximumCupcakes(trips):
4
5     for t in trips[1:] :
6         m = int(t[0])
7         p = int(t[1])
8         w = int(t[3])
9
10        c1 = round(m / p)
11        c2 = round(c1 / w)
12        c = c1 + c2
13
14    print(c)
```

TESTCASE	TYPE		STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	✗	Runtime Error	0	0.01 sec	7.16 MB
Testcase 1	Easy	✗	Runtime Error	0	0.01 sec	7.1 MB
Testcase 2	Easy	✗	Runtime Error	0	0.01 sec	7.27 MB
Testcase 3	Easy	✗	Runtime Error	0	0.01 sec	7.2 MB
Testcase 4	Easy	✗	Runtime Error	0	0.01 sec	7.09 MB
Testcase 5	Easy	✗	Runtime Error	0	0.01 sec	7.33 MB
Testcase 6	Easy	✗	Runtime Error	0	0.01 sec	7.39 MB
Testcase 7	Easy	✗	Runtime Error	0	0.01 sec	7.22 MB
Testcase 8	Easy	✗	Runtime Error	0	0.02 sec	7.22 MB
Testcase 9	Easy	✗	Runtime Error	0	0.01 sec	8.5 MB

No Comments