You can view this report online at : https://www.hackerrank.com/x/tests/1731411/candidates/57545072/report

Full Name:	Instructor
Email:	aisha.batool@sse.habib.edu.pk
Test Name:	**CS101 - LW11 - Fall23**
Taken On:	29 Oct 2023 22:16:09 PKT
Time Taken:	12 min 52 sec/ 180 min
Work Experience:	> 5 years
Invited by:	Aisha
Skills Score:	
Tags Score:	Easy 10/10 Functions 10/10 Input 10/10

100%

60/60

scored in **CS101 - LW11 - Fall23** in 12 min 52 sec on 29 Oct 2023 22:16:09 PKT

Recruiter/Team Comments:

No Comments.

Plagiarism flagged

We have marked questions with suspected plagiarism below. Please review it in detail here - https://www.hackerrank.com/x/tests/1731411/candidates/57545072/report

	Question Description	Time Taken	Score	Status
Q1	Holes in a Number - Python > **Coding**	24 sec	10/ 10	✓
Q2	Recursively speaking, is it a palindrome? > **Coding**	33 sec	10/ 10	✓
Q3	Greatest Common Divisor > **Coding**	44 sec	10/ 10	!
Q4	Computing a Fibonacci number > **Coding**	1 min 13 sec	10/ 10	✓
Q5	Integer to Text > **Coding**	9 min	10/ 10	✓
Q6	RecursiveReverse > **Coding**	41 sec	10/ 10	✓
Q7	Difficulty Meter > **Multiple Choice**	4 sec	0/ 0	✓

QUESTION 1

✓

Holes in a Number - Python > Coding

Easy Functions Input

Correct Answer

Score 10

QUESTION DESCRIPTION

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

The number of holes present in each of the digits from 0 to 9 is 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.

The total number of holes in the number, 1078, is 3 which is the the sum of

- the number of holes in 1 = 0 holes.
- the number of holes in 0 = 1 hole.
- the number of holes in 7 = 0 holes.
- the number of holes in 8 = 2 holes.

Function Description

Write a **recursive** function `countHoles()` that takes a parameter `num`, and returns the sum of the number of holes in all of its digits. The function must not convert to a `str`.

Constraints

- `num is of type int and num >= 0`
- You may not use global references.
- You must write a recursive solution--you may not use *for* or *while* loops, or any other iterative methods.
- You may not write any helper functions.

Input and Output

Input and Output has been handled by Hackerrank.

▼ Input Format For Custom Testing

The input consists of one line which contains `num`.

The output must contain the number of holes in `num`.

▼ Sample Case 0

Sample Input For Custom Testing

```
1078
```

Sample Output

```
3 holes
```

Explanation

The number of holes in 1, 0, 7, and 8 are 0, 1, 0, and 2 respectively which add up to 3.

▼ Sample Case 1

Sample Input For Custom Testing

```
819
```

Sample Output

```
3 holes
```

Explanation

The number of holes in 8, 1, and 9 are 2, 0, and 1 respectively which add up to 3.

INTERVIEWER GUIDELINES

Solution

```
def countHoles(num):
    if num < 0:
        return 0
    else:
        holes = {1:0, 2:0, 3:0, 5:0, 7:0, 0: 1, 4: 1, 6: 1, 9: 1, 8: 2}
        num, last = divmod(num, 10)
        if num==0:
            num=-1
        return holes[last] + countHoles(num)
```

CANDIDATE ANSWER

Language used: **Python 3**

```
def countHoles(num):
    if num < 0:
        return 0
    else:
        holes = {1:0, 2:0, 3:0, 5:0, 7:0, 0: 1, 4: 1, 6: 1, 9: 1, 8: 2}
        num, last = divmod(num, 10)
        if num==0:
            num=-1
        return holes[last] + countHoles(num)
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	1	0.2104 sec	11.7 KB
Testcase 1	Easy	Sample case	✓ Success	1	0.1679 sec	11.8 KB
Testcase 2	Easy	Sample case	✓ Success	1	0.131 sec	11.8 KB
Testcase 3	Easy	Sample case	✓ Success	1	0.0779 sec	11.7 KB
Testcase 4	Easy	Sample case	✓ Success	1	0.1468 sec	11.6 KB
Testcase 5	Easy	Hidden case	✓ Success	1	0.145 sec	11.8 KB
Testcase 6	Easy	Hidden case	✓ Success	1	0.0747 sec	11.8 KB
Testcase 7	Easy	Hidden case	✓ Success	1	0.0623 sec	11.8 KB
Testcase 8	Easy	Hidden case	✓ Success	1	0.0777 sec	11.8 KB
Testcase 9	Easy	Hidden case	✓ Success	1	0.0895 sec	11.8 KB

No Comments

QUESTION 2

✓

Correct Answer

Score 10

Recursively speaking, is it a palindrome? › Coding

QUESTION DESCRIPTION

Challenge

Write a **recursive** function is_palindrome(s) that returns True if the given string s is a palindrome (reads the same forward and backward), False otherwise.

Note: you may not use while or for loops, and you may not use a string slice with a negative step.

Sample

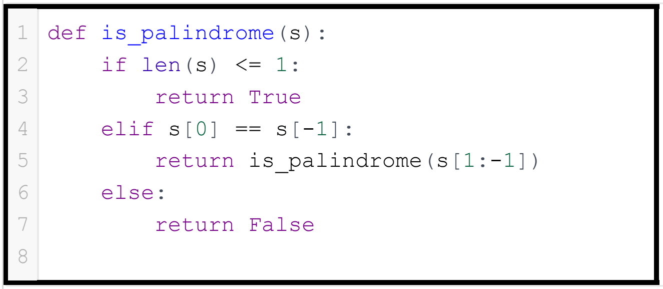
```
>>> print(is_palindrome('racecar'))
True
>>> print(is_palindrome('racecars'))
False
```

INTERVIEWER GUIDELINES

```
def is_palindrome(s):
    if len(s) <= 1:
        return True
    elif s[0] == s[-1]:
        return is_palindrome(s[1:-1])
    else:
        return False
```

CANDIDATE ANSWER

Language used: **Python 3**



```
def is_palindrome(s):
    if len(s) <= 1:
        return True
    elif s[0] == s[-1]:
        return is_palindrome(s[1:-1])
    else:
        return False

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	1	0.0607 sec	11.3 KB
Testcase 1	Easy	Sample case	✓ Success	1.5	0.0743 sec	11.6 KB
Testcase 2	Easy	Sample case	✓ Success	1.5	0.068 sec	11.5 KB
Testcase 3	Easy	Sample case	✓ Success	1.5	0.0631 sec	11.6 KB
Testcase 4	Easy	Sample case	✓ Success	1.5	0.0589 sec	11.4 KB
Testcase 5	Easy	Sample case	✓ Success	1.5	0.0588 sec	11.3 KB
Testcase 6	Easy	Sample case	✓ Success	1.5	0.0646 sec	11.4 KB

No Comments

QUESTION 3

Needs Review

Score 10

Greatest Common Divisor › Coding

QUESTION DESCRIPTION

Problem

Credit: The Famous People

Euclid was a mathematician who lived in Greece between the 4th and 3rd centuries BC. His work continues to be an inspiration today and he is often termed the Father of Geometry.[1] One of his contributions is an algorithm to find the greatest common divisor (gcd) of two numbers. The algorithm has come to be known as Euclid's algorithm and it finds the gcd of positive numbers A and B where A>B as follows.[2]

- If A = 0 then GCD(A,B)=B.
- If B = 0 then GCD(A,B)=A.
- Write A in quotient remainder form (A = B·Q + R)
- Find GCD(B,R) using the Euclidean Algorithm since GCD(A,B) = GCD(B,R)

For example, with A = 2322, B = 654.[3]

gcd(2322, 654) = gcd(654, 360)	because	2322 = 654·3 + 360
gcd(654, 360) = gcd(360, 294)	because	654 = 360·1 + 294
gcd(360, 294) = gcd(294, 66)	because	360 = 294·1 + 66
gcd(294, 66) = gcd(66, 30)	because	294 = 66·4 + 30
gcd(66, 30) = gcd(30, 6)	because	66 = 30·2 + 6
gcd(30, 6) = gcd(6,0) = 6	because	30 = 6·5

Therefore, gcd(2322,654) = 6.

Write a *recursive* function named `gcd` that takes parameters `a` and `b` and returns their gcd.

Sample

```
>>> gcd(2322,654)
6
>>> gcd(12,8)
4
>>> gcd(287,175)
7
```

Input

Input `a` and `b` from the console without any prompt.

Constraints

- `isinstance(a, int)`
- `isinstance(b, int)`
- `a>=0 and b>=0`

INTERVIEWER GUIDELINES

Solution

```
a = int(input())
b = int(input())
def gcd(a, b):
    if a < b:
        return gcd(b,a)
    if a == 0:
        return b
    if b == 0:
        return a
    return gcd(b, a%b)
```

CANDIDATE ANSWER

Language used: **Python 3**

```python
# Enter your code here.
a = int(input())
b = int(input())
def gcd(a, b):
    if a < b:
        return gcd(b,a)
    if a == 0:
        return b
    if b == 0:
        return a
    return gcd(b, a%b)

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 0	Easy	Sample case	✓ Success	2.5	0.094 sec	11.5 KB
TestCase 1	Easy	Sample case	✓ Success	2.5	0.0747 sec	11.8 KB
TestCase 2	Easy	Sample case	✓ Success	2.5	0.0588 sec	11.8 KB
TestCase 3	Easy	Sample case	✓ Success	2.5	0.0987 sec	11.6 KB

No Comments

QUESTION 4

✓

Correct Answer

Score 10

Computing a Fibonacci number › Coding

QUESTION DESCRIPTION

Problem

Complete the ***recursive*** function named `Fibonacci` for computing a Fibonacci number using Recursion. Your program should return the nth number entered by user residing in the Fibonacci series.

The Fibonacci series begins with 0 and 1. Each subsequent term is computed as the sum of the last 2 terms, thus yielding

```
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...
```

Sample

```
>>> Fibonacci(2)
1
>>> Fibonacci(9)
34
```

Constraints

- `isinstance(n, int)` is `True`
- `n >= 2` is `True`

INTERVIEWER GUIDELINES

```python
def Fibonacci(index):
    if index == 0: # Base case
        return 0
    elif index == 1: # Base case
        return 1
    else: # Reduction and recursive calls
        return Fibonacci(index-1) + Fibonacci(index-2)
```

CANDIDATE ANSWER

Language used: **Python 3**

```

# Enter your code here. Read input from STDIN. Print output to STDOUT
def Fibonacci(index):
    if index == 0: # Base case
        return 0
    elif index == 1: # Base case
        return 1
    else: # Reduction and recursive calls
        return Fibonacci(index-1) + Fibonacci(index-2)

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	1.65	0.1039 sec	11.6 KB
Testcase 1	Easy	Sample case	✓ Success	1.67	0.055 sec	11.6 KB
Testcase 2	Easy	Hidden case	✓ Success	1.67	0.0846 sec	11.6 KB
Testcase 3	Easy	Hidden case	✓ Success	1.67	0.057 sec	11.4 KB
Testcase 4	Easy	Hidden case	✓ Success	1.67	0.1394 sec	11.7 KB
Testcase 5	Easy	Hidden case	✓ Success	1.67	0.1323 sec	11.5 KB

No Comments

QUESTION 5

✓

Correct Answer

Score 10

Integer to Text › Coding

QUESTION DESCRIPTION

Problem

The `ord` function in python returns the *ASCII code* of a given character, e.g.

```
>>> ord('C')
67
>>> ord(']')
93
```

The `chr` function performs the reverse, i.e. it returns the character corresponding to a given ASCII code, e.g.

```
>>> chr(67)
'C'
>>> chr(93)
']'
```

The ASCII code of each digit differs from the digit by 48. This can be used to obtain the character corresponding to a digit, e.g.

```
>>> n=0; chr(n+48)
'0'
>>> n=5; chr(n+48)
'5'
```

Write a *recursive* function named `int_to_text` that takes an integer parameter `n` and returns its conversion to string without using the `str` function. For its base case, i.e. when `n` is a single digit, it uses the above method to convert the digit to a string.

Sample

```
>>> int_to_text(0)
'0'
>>> int_to_text(3401)
'3401'
>>> int_to_text(-56)
'-56'
```

Input

The input contains `n` on the first line. Your function need not take input.

Constraints

- `isinstance(n, int)`

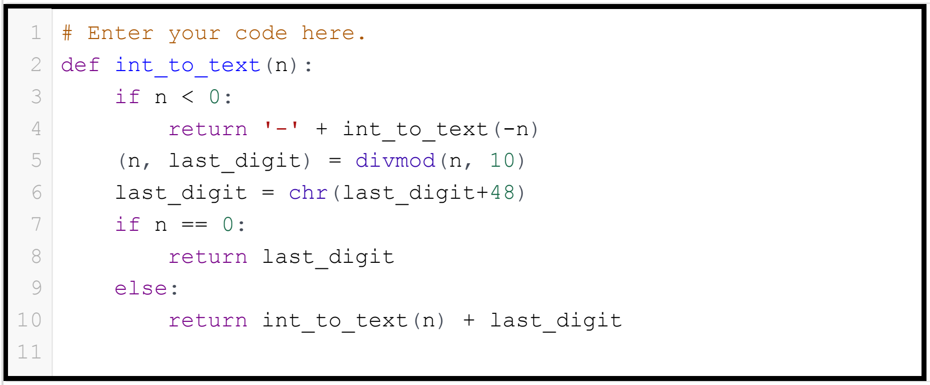
INTERVIEWER GUIDELINES

Solution

```
def int_to_text(n):
    if n < 0:
        return '-' + int_to_text(-n)
    (n, last_digit) = divmod(n, 10)
    last_digit = chr(last_digit+48)
    if n == 0:
        return last_digit
    else:
        return int_to_text(n) + last_digit
```

CANDIDATE ANSWER

Language used: **Python 3**



```
# Enter your code here.
def int_to_text(n):
    if n < 0:
        return '-' + int_to_text(-n)
    (n, last_digit) = divmod(n, 10)
    last_digit = chr(last_digit+48)
    if n == 0:
        return last_digit
    else:
        return int_to_text(n) + last_digit

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED

Testcase 0	Easy	Sample case	✓ Success	1.1	0.0593 sec	11.8 KB
Testcase 1	Easy	Sample case	✓ Success	1.1	0.0666 sec	11.3 KB
Testcase 2	Easy	Sample case	✓ Success	1.1	0.1164 sec	11.5 KB
Testcase 3	Easy	Sample case	✓ Success	1.1	0.1089 sec	11.6 KB
Testcase 4	Easy	Sample case	✓ Success	1.1	0.0804 sec	11.5 KB
Testcase 5	Easy	Hidden case	✓ Success	1.1	0.0573 sec	11.5 KB
Testcase 6	Easy	Hidden case	✓ Success	1.1	0.0715 sec	11.5 KB
Testcase 7	Easy	Hidden case	✓ Success	1.1	0.086 sec	11.5 KB
Testcase 8	Easy	Hidden case	✓ Success	1.2	0.0936 sec	11.8 KB

No Comments

QUESTION 6

✓

Correct Answer

Score 10

RecursiveReverse › Coding

QUESTION DESCRIPTION

Problem Description:

You are required to create a recursive method to reverse the characters in a given string. Do not use "for" or "while" loops.

Function Description:

Write a recursive method `reverse_string` that takes a string `s` as an argument and returns the reversed string. The method should work through a recursive process to achieve the reversal.

Sample:

```
>> reverse_string("book")
koob
```

Constraints:

- You can't use the built-in reverse function
- You may not use a string slice with a negative step.

INTERVIEWER GUIDELINES

```
def reverse_string(s):
    # Base Case: If the string is empty or has only one character, return
the string as it is.
    if len(s) <= 1:
        return s
    else:
        # Recursive Case: Reverse the substring excluding the first
character and append the first character at the end.
        return reverse_string(s[1:]) + s[0]
```

CANDIDATE ANSWER

Language used: **Python 3**

```

def reverse_string(s):

    # Base Case: If the string is empty or has only one character, return the string as it is.
    if len(s) <= 1:
        return s
    else:
        # Recursive Case: Reverse the substring excluding the first character and append the first character at the end.
        return reverse_string(s[1:]) + s[0]
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	2.5	0.1434 sec	11.8 KB
Testcase 1	Easy	Hidden case	✓ Success	2.5	0.0836 sec	11.5 KB
Testcase 2	Easy	Hidden case	✓ Success	2.5	0.0578 sec	11.7 KB
Testcase 3	Easy	Hidden case	✓ Success	2.5	0.116 sec	11.7 KB

No Comments

QUESTION 7

✓

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)

✓ ● 1

✓ ○ 2

✓ ○ 3

✓ ○ 4

✓ ○ 5

No Comments