CS 101 - Lab 8 - Fall 2021 > rr07656@st.habib.edu.pk

You can view this report online at : https://www.hackerrank.com/x/tests/1238592/candidates/32749776/report

Full Name:	Rohan Raj
Email:	rr07656@st.habib.edu.pk
Test Name:	**CS 101 - Lab 8 - Fall 2021**
Taken On:	16 Nov 2021 12:13:15 PKT
Time Taken:	10819 min 46 sec/ 15000 min
Work Experience:	< 1 years
Invited by:	Aisha
Skills Score:	
Tags Score:	CS101 40/40 Recursion 40/40

100%

260/260

scored in **CS 101 - Lab 8 - Fall 2021** in 10819 min 46 sec on 16 Nov 2021 12:13:15 PKT

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	**Countdown - Recursively > Coding**	18 min 8 sec	20/ 20	✓
Q2	**Ranged countdown - Recursively > Coding**	9 min 35 sec	20/ 20	✓
Q3	**Countup - Recursively > Coding**	2 min 33 sec	20/ 20	✓
Q4	**Calculating Odd sum - Recursively > Coding**	37 min 39 sec	30/ 30	✓
Q5	**Calculating Sum of Digits - Recursively > Coding**	1 hour 18 min 45 sec	30/ 30	✓
Q6	**Recursive product > Coding**	2 hour 23 min 36 sec	70/ 70	✓
Q7	**Recursive exponent > Coding**	1 hour 13 min 55 sec	70/ 70	✓

QUESTION 1

✓ Correct Answer

Score 20

Countdown - Recursively > Coding

QUESTION DESCRIPTION

Problem

Write a *recursive* function named `countdown` that takes a parameter `n` and prints all numbers from `n` down to 1.

Sample

```
>>> countdown(4)
```

```
4, 3, 2, 1
>>> countdown(1)
1
```

Input

Input n from the console without any prompt.

Constraints

- isinstance(n, int) is True
- n > 0 is True

INTERVIEWER GUIDELINES

Solution

```
n = int(input())
def countdown(n):
    if n == 1:
        print(1)
    else:
        print(n, end=', ')
        countdown(n-1)
```

CANDIDATE ANSWER

Language used: **Python 3**

```
n = int(input())
def countdown (n):
    if n == 1:
        print (n)
    else:
        print (n, end = ", ")
        countdown (n - 1)
# Enter your code here.
import inspect
source = inspect.getsource(countdown)
if 'for' in source or 'while' in source:
  print('Try to solve the problem recursively!')
else:
  countdown(n)

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	5	0.0579 sec	10.1 KB
Testcase 1	Easy	Sample case	✓ Success	5	0.0695 sec	10.1 KB
Testcase 2	Easy	Hidden case	✓ Success	5	0.07 sec	10.1 KB
Testcase 3	Easy	Hidden case	✓ Success	5	0.0594 sec	10.2 KB

No Comments

QUESTION 2 ✓

Ranged countdown - Recursively › Coding

Recursion CS101

Correct Answer

Score 20

QUESTION DESCRIPTION

Problem

Write a *recursive* function named `countdown2` that takes two parameters `a` and `b` and prints all numbers from `b` down to `a` inclusive.

Sample

```
>>> countdown2(-3, 5)
5, 4, 3, 2, 1, 0, -1, -2, -3
>>> countdown2(0, 0)
0
```

Input

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

Constraints

- `isinstance(a, int)` is `True`
- `isinstance(b, int)` is `True`
- `b >= a` is `True`

INTERVIEWER GUIDELINES

Solution

```
a = int(input())
b = int(input())
def countdown2(a, b):
    if b == a:
        print(b)
    else:
        print(b, end=', ')
        countdown2(a, b-1)
```

CANDIDATE ANSWER

Language used: **Python 3**

```
a = int(input())
b = int(input())

def countdown2(a, b):
    if b > a:
        print (b, end = ", ")
        countdown2 (a, b - 1)
    if b == a:
        print (b)



```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	5	0.0675 sec	10.2 KB
Testcase 1	Easy	Sample case	✓ Success	5	0.0515 sec	10.2 KB
Testcase 2	Easy	Hidden case	✓ Success	5	0.0686 sec	10.2 KB
Testcase 3	Easy	Hidden case	✓ Success	5	0.0605 sec	10.1 KB

Testcase 3	Easy	Hidden case	✓ Success	5	0.0605 sec	10.1 KB

No Comments

QUESTION 3

✓

Correct Answer

Score 20

Countup - Recursively › Coding

Recursion | CS101

QUESTION DESCRIPTION

Problem

Write a *recursive* function named `countup` that takes two parameters `a` and `b` and prints all numbers from `a` to `b`.

Sample

```
>>> countup(-3, 5)
-3, -2, -1, 0, 1, 2, 3, 4, 5
>>> countup(0, 0)
0
```

Input

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

Constraints

- `isinstance(a, int)` is `True`
- `isinstance(b, int)` is `True`
- `b >= a` is `True`

INTERVIEWER GUIDELINES

Solution

```
a = int(input())
b = int(input())
def countup(a, b):
    if a == b:
        print(a)
    else:
        print(a, end=', ')
        countup(a+1, b)
```

CANDIDATE ANSWER

Language used: **Python 3**

```
a = int(input())
b = int(input())
def countup (a,b):
    if b > a:
        print (a, end = ", ")
        countup (a + 1, b)
    if b == a:
        print (a)

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	5	0.0598 sec	10.1 KB
Testcase 1	Easy	Sample case	✓ Success	5	0.0702 sec	10.1 KB
Testcase 2	Easy	Hidden case	✓ Success	5	0.0545 sec	10 KB
Testcase 3	Easy	Hidden case	✓ Success	5	0.051 sec	10.1 KB

No Comments

QUESTION 4

✓

Correct Answer

Score 30

Calculating Odd sum - Recursively › Coding

QUESTION DESCRIPTION

Problem

Write a *recursive* function named `sum_odd` that takes two parameters `a` and `b` and returns the sum of all odd numbers between `a` and `b` inclusive.

Sample

```
>>> sum_odd(1,1)
1
>>> sum_odd(2,2)
0
>>> sum_odd(1,2)
1
>>> sum_odd(2,3)
3
>>> sum_odd(2,13)
48
```

Input

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

Constraints

- `isinstance(a, int)` is `True`
- `isinstance(b, int)` is `True`
- `b >= a` is `True`

INTERVIEWER GUIDELINES

Solution

```
a = int(input())
b = int(input())
def sum_odd(a,b):
    if a > b:
        return 0
    if a == b:
        if a % 2 == 1:
            return a
        else:
            return 0
    else:
        if a % 2 == 1:
            return a + sum_odd(a+2, b)
        else:
            return sum_odd(a+1, b)

'''
# Enter your code here.
a = int(input())
```

```
a = int(input())
b = int(input())

def sum_odd(a,b):
    if a == b:
        if a%2 == 0:
            return 0
        return a
    elif a % 2 != 0 :
        return a + sum_odd(a+1,b)
    return sum_odd(a+1,b)
'''
```

CANDIDATE ANSWER

Language used: **Python 3**

```python
a = int(input())
b = int(input())
def sum_odd(a,b):
    if a != b:
        if a % 2 == 0:
            a = a + 1
        if b % 2 == 0:
            b = b - 1
    if b > a:
        return ((int(a)) + int(sum_odd (a + 2, b)))
    elif a == b:
        if a % 2 == 0:
            return 0
        else:
            return (a)


```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	5	0.0523 sec	10.1 KB
Testcase 1	Easy	Sample case	✓ Success	5	0.052 sec	10.2 KB
Testcase 2	Easy	Hidden case	✓ Success	5	0.0566 sec	10.1 KB
Testcase 3	Easy	Hidden case	✓ Success	5	0.0698 sec	10.3 KB
Testcase 4	Easy	Hidden case	✓ Success	5	0.0806 sec	10.2 KB
Testcase 5	Easy	Hidden case	✓ Success	5	0.0578 sec	10.1 KB

No Comments

QUESTION 5

✓

Correct Answer

Score 30

Calculating Sum of Digits - Recursively › Coding

QUESTION DESCRIPTION

Problem

Write a *recursive* function named `sum_digits` that takes a parameter `n` and returns the sum of all the digits in `n`.

Sample

```
>>> sum_digits(24)
6
>>> sum_digits(0)
0
>>> sum_digits(1092)
12
```

Input

Input n from the console without any prompt.

Constraints

- isinstance(n, int) is True
- n >= 0 is True

INTERVIEWER GUIDELINES

Solution

```
n = int(input())
def sum_digits(n):
    if n == 0:
        return 0
    else:
        (n, last_digit) = divmod(n, 10)
        return last_digit + sum_digits(n)
```

CANDIDATE ANSWER

Language used: **Python 3**

```
n = int(input())
def sum_digits(n):
    a = n % 10
    b = n // 10
    if b == 0:
        return a
    else:
        return (a + (sum_digits(b)))
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	5	0.0737 sec	10.1 KB
Testcase 1	Easy	Sample case	✓ Success	5	0.0445 sec	10.1 KB
Testcase 2	Easy	Sample case	✓ Success	5	0.0733 sec	9.93 KB
Testcase 3	Easy	Hidden case	✓ Success	5	0.0518 sec	9.98 KB
Testcase 4	Easy	Hidden case	✓ Success	5	0.0629 sec	10.1 KB
Testcase 5	Easy	Hidden case	✓ Success	5	0.0629 sec	10.2 KB

No Comments

QUESTION 6

✓

Recursive product › Coding

Correct Answer

Score 70

QUESTION DESCRIPTION

Problem

Write a function named `product` to recursively compute the product of its parameters a and b using addition only.

Sample

```
>>> product(1,-5)
-5
>>> product(-1,-5)
5
>>> product(1,0)
0
>>> product(0,1)
0
>>> product(0,0)
0
>>> product(4,-3)
-12
```

Input

Your code should input a and b from the console without a prompt.

Constraints

- `isinstance(a, int)` is `True`
- `isinstance(b, int)` is `True`

INTERVIEWER GUIDELINES

Solution

```
a = int(input())
b = int(input())
def product(a, b):
    if b == 0 or a == 0:
        return 0
    elif b < 0:
        return -product(a,-b)
    elif a < 0:
        return -product(-a,b)
    else:
        return a + product(a, b-1)
```

CANDIDATE ANSWER

Language used: **Python 3**

```
a = int(input())
b = int(input())
def product(a, b):
    if b == 0:
        return 0
    elif (a < 0 and b < 0) or (a > 0 and b < 0):
        return (-a + int(product(a, b+1)))
    else:
        return (a + int(product(a, b-1)))

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED

Testcase 0	Easy	Sample case	✓ Success	5	0.0849 sec	10.1 KB
Testcase 1	Easy	Sample case	✓ Success	5	0.0469 sec	10.1 KB
Testcase 2	Easy	Sample case	✓ Success	5	0.0722 sec	10 KB
Testcase 3	Easy	Sample case	✓ Success	5	0.0792 sec	10.1 KB
Testcase 4	Easy	Hidden case	✓ Success	5	0.0609 sec	10.3 KB
Testcase 5	Easy	Hidden case	✓ Success	5	0.0523 sec	10.1 KB
Testcase 6	Easy	Sample case	✓ Success	10	0.0493 sec	10.1 KB
Testcase 7	Easy	Sample case	✓ Success	10	0.0738 sec	10.1 KB
Testcase 8	Easy	Sample case	✓ Success	10	0.0512 sec	10.3 KB
Testcase 9	Easy	Sample case	✓ Success	10	0.0517 sec	10.1 KB

No Comments

QUESTION 7

✓

Correct Answer

Score 70

Recursive exponent › Coding

QUESTION DESCRIPTION

Problem

Write a function named `power` to recursively compute the exponentiation of its parameters a and b i.e. a^b using multiplication only.

Sample

```
>>> power(3,4)
81
>>> power(0,2)
0
>>> power(2,0)
1
>>> power(-2,1)
-2
>>> power(-2,-1)
-0.5
```

Input

Your code should input a and b from the console without a prompt.

Constraints

- `isinstance(a, int)` is `True`
- `isinstance(b, int)` is `True`

INTERVIEWER GUIDELINES

Solution

```
a = int(input())
b = int(input())
def power(a, b):
    if b == 0:
        return 1
    elif b < 1:
        return 1/power(a,-b)
    else:
        return a * power(a, b-1)
```

CANDIDATE ANSWER

Language used: **Python 3**

```
a = int(input())
b = int(input())
def power(a, b):
    if b == 0:
        return 1
    elif b == 1 or a == 0:
        return a
    elif b > 0:
        return (a * power (a, b - 1))
    elif b < 0:
        return ((1 / a) * power (a, b + 1))

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	5	0.0549 sec	10.2 KB
Testcase 1	Easy	Sample case	✓ Success	5	0.0552 sec	9.98 KB
Testcase 2	Easy	Sample case	✓ Success	5	0.0548 sec	10.2 KB
Testcase 3	Easy	Sample case	✓ Success	5	0.0566 sec	10 KB
Testcase 4	Easy	Sample case	✓ Success	5	0.0502 sec	10.1 KB
Testcase 5	Easy	Sample case	✓ Success	5	0.0577 sec	10.1 KB
Testcase 6	Easy	Sample case	✓ Success	10	0.0529 sec	9.97 KB
Testcase 7	Easy	Sample case	✓ Success	10	0.0795 sec	10.3 KB
Testcase 8	Easy	Sample case	✓ Success	10	0.0846 sec	10.3 KB
Testcase 9	Easy	Sample case	✓ Success	10	0.0733 sec	10.3 KB

No Comments

PDF generated at: 23 Nov 2021 19:35:05 UTC