100
Pass

ryan

Other

PDF generated at: 5 Jun 2024 00:12:16 UTC

View this report on HackerRank

Score

100% • 60 / 60

scored in CodePath TIP101: Unit 1 Assessment, Version A - Summer 2024 in 25 min 55 sec on 4 Jun 2024 16:44:36 PDT

Candidate Information

Email	concepting@protonmail.com
Test	CodePath TIP101: Unit 1 Assessment, Version A - Summer 2024
Candidate Packet	View
Taken on	4 Jun 2024 16:44:36 PDT
Time taken	25 min 55 sec/ 90 min
Work Experience	< 1 years
Invited by	CodePath

Skill Distribution

There is no associated skills data that can be shown for this assessment

Tags Distribution

There is no associated tags data that can be shown for this assessment

Questions

Status	No.	Question	Time Taken	Skill	Score
✓	1	Conditional Statements Multiple Choice	30 sec	-	5/5
✓	2	What is the result? Multiple Choice	2 min 36 sec	-	5/5
✓	3	Mystery Function Multiple Choice	1 min 16 sec	-	5/5

✓	4	Count Negatives Multiple Choice	2 min 39 sec	-	5/5
✓	5	Find Sum Coding	9 min 57 sec	-	20/20
✓	6	List Minimum Coding	8 min 32 sec	-	20/20

1. Conditional Statements

✓ Correct

Multiple Choice

Question description

What would the following code print out?

```
x = 15
y = 20

if x > y and y < 30:
  print("A")
elif x < y and x < 10:
  print("B")
elif x > 15 and x < y:
  print("C")
else:

  print("D")
```

Candidate's Solution

Options: (Expected answer indicated with a tick)

- [ ] A
- [ ] B
- [ ] C
- [x] D

No comments.

2. What is the result?

Correct

Multiple Choice

Question description

Given the following code, what is the value of result?

```
final_ans = 0

for num in range(1, 6):
    final_ans += num

result = final_ans * 2
```

Candidate's Solution

Options: (Expected answer indicated with a tick)

- ○ 15
- ○ 42
- ● 30 ✓
- ○ 12

No comments.

3. Mystery Function

Correct

Multiple Choice

Question description

Given the following code, what is the value of output?

```
def mystery_function(lst1, lst2):
    for num in lst2:
        lst1.append(num)
    return lst1

output = mystery_function([1,2,3,4], [5,6,7,8])
```

Candidate's Solution

Options: (Expected answer indicated with a tick)

- [ ] [1, 2, 3, 4]
- [ ] [5, 6, 7, 8]
- [x] **[1, 2, 3, 4, 5, 6, 7, 8]**
- [ ] [1, 5, 2, 6, 3, 7, 4, 8]

No comments.

4. Count Negatives

Correct

Multiple Choice

Question description

count_negatives should return the number of negative numbers in a given input list. For example, if we passed in [-1, 2, 3, 4, -5], **count_negatives** should return 2. One of the following implementations is correct. The rest have a bug. Choose the option that correctly implements **count_negatives**.

Candidate's Solution

Options: (Expected answer indicated with a tick)

- [x] <pre> <code>def count_negatives(lst): count = 0 for num in lst: if num < 0: count += 1 return count</code></pre>

- [ ] <pre> <code>def count_negatives(lst): count = 0 for num in lst: if num <= 0: count += 1 return count</code></pre>

- [ ] <pre> <code>def count_negatives(lst): count = 0 for num in lst: if num < 0: count += num return count</code></pre>

- [ ] <pre> <code>def count_negatives(lst): count = 0 for num in range(len(lst)): if num < 0: count += 1 return count</code></pre>

No comments.

5. Find Sum

Correct

Coding

Question description

Write a function that returns the sum of all the elements in a list. Do not use the built-in **sum** function.

Example 1:

Input: [1, 2, 3, 4, 5]
Output: 15

Example 2:

Input: [2, 4, 6, 8, 10]
Output: 30

Candidate's Solution Language used: **Python 3**

```python
#!/bin/python3

import math
import os
import random
import re
import sys



#
# Complete the 'find_sum' function below.
#
# The function is expected to return an INTEGER.
# The function accepts INTEGER_ARRAY lst as parameter.
#

def find_sum(lst):
    # Write your code here
    sum = 0
    for i in range(0, len(lst)):
        sum = lst[i] + sum
    return sum

print(sum)


if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    temp = input()

    if len(temp) > 70:
        input_string = temp
        chunks = input_string.split(", ")
        list_of_lists = [list(map(int, chunk.split())) for chunk in chunks]
```

```
        result = [find_sum(lst) for lst in list_of_lists]
    else:
        result = find_sum([int(n) for n in temp.split()])

    fptr.write(str(result) + '\n')

    fptr.close()

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample	Success	0	0.0278 sec	10.4 KB
Testcase 1	Easy	Sample	Success	0	0.0324 sec	10.3 KB
Testcase 2	Easy	Hidden	Success	0	0.0324 sec	10.2 KB
Testcase 4	Easy	Hidden	Success	0	0.0321 sec	10.2 KB
Testcase 4	Easy	Hidden	Success	20	0.0386 sec	10.3 KB

No comments.

6. List Minimum

Correct

Coding

Question description

Without using the built-in function **min**, write a function that finds the minimum value in a list of integers.

Example:

Input: [5, 1, 2, 3, 4]
Output: 1

Example 2:

Input: [10, 8, 2, 4, 6]
Output: 2

Candidate's Solution Language used: **Python 3**

```python
#!/bin/python3

import math
import os
import random
import re
import sys



#
# Complete the 'find_min' function below.
#
# The function is expected to return an INTEGER.
# The function accepts INTEGER_ARRAY lst as parameter.
#

def find_min(lst):
    # Write your code here
    min = lst[0]

    for i in lst:
        if i < min:
            min = i
    return min

print(find_min)
```

```
28
29 if __name__ == '__main__':
30     fptr = open(os.environ['OUTPUT_PATH'], 'w')
31
32     temp = input()
33
34     if len(temp) > 60:
35         input_string_new = temp
36         chunks_new = input_string_new.split(", ")
37         list_of_lists_new = [list(map(int, chunk.split())) for chunk in
   chunks_new]
38         result = [find_min(lst) for lst in list_of_lists_new]
39     else:
40         result = find_min([int(n) for n in temp.split()])
41
42     fptr.write(str(result) + '\n')
43
44     fptr.close()
45
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample	Success	0	0.0307 sec	10.3 KB
Testcase 1	Easy	Sample	Success	0	0.0274 sec	10.3 KB
Testcase 2	Easy	Hidden	Success	0	0.0347 sec	10.4 KB
Testcase 3	Easy	Hidden	Success	0	0.0376 sec	10.3 KB
Testcase 4	Easy	Hidden	Success	0	0.0506 sec	10.3 KB

Testcase 5	Easy	Hidden	Success	20	0.0404 sec	10.3 KB

No comments.