Started on	Friday, 3 May 2024, 12:11 PM
State	Finished
Completed on	Wednesday, 15 May 2024, 3:35 PM
Time taken	12 days 3 hours
Marks	5.00/5.00
Grade	**50.00** out of 50.00 (**100**%)
Name	TAMILARASI R 2022-CSD-A

Question **1**

Correct

Mark 1.00 out of 1.00

Write a program to read a string and a character and find the whether the character is available in the string or not. Print True if the character is present in the string, False otherwise.

Sample Input

Rakalakshmi

a

Sample Output

True

Sample Input

Rakalakshmi

b

Sample Output

False

Answer: (penalty regime: 0 %)

```
def is_character_present(string, char):
    return char in string

# Read input string and character
input_string = input()
input_char = input()

# Check if character is present in the string
result = is_character_present(input_string, input_char)

# Print the result
print(result)

```

	Input	Expected	Got	
✔	Rajalakshmi a	True	True	✔
✔	Rajalakshmi b	False	False	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Create a tuple, remove an item from the tuple, and display the tuple.

Sample input:

5 : No of items

2020 : tuple items

'd'

"rec"

'python'

'tuple'

python : item to be removed

Sample Output:

('2020','d,'rec','tuple')

For example:

Input	Result
4 samsung vivo redmi Vijay Vijay	('samsung', 'vivo', 'redmi')

Answer: (penalty regime: 0 %)

```
# Read the number of items in the tuple
n = int(input())

# Initialize an empty list to store tuple items
items = []

# Read tuple items
for _ in range(n):
    item = input()
    items.append(item)

# Convert the list of items to a tuple
my_tuple = tuple(items)

# Read the item to be removed
remove_item = input()

# Remove the item from the tuple
my_tuple = tuple(item for item in my_tuple if item != remove_item

# Display the modified tuple

```

	Input	Expected	Got	
✔	4 samsung vivo redmi Vijay Vijay	('samsung', 'vivo', 'redmi')	('samsung', 'vivo', 'redmi')	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

A customer wants to buy a mobile phone in a online mart, the customer finds different prices from different seller, the item price is been stored in a nested tuples in the following order ((seller_name_name,item-name,item_cost)), consider the tuple has 5 seller, write a program to help the customer to view in the order of lowest price of item first and so on.

sample input:

seller_1

samsung

45000.00

seller_2

samsung

45500.00

seller_3

samsung

44700.00

seller_4

samsung

43900.00

seller_5

samsung

44100.00

sample output:

(("seller_4","samsung","43900.00"),("seller_5","samsung","44100.00"),("seller_3","samsung","44700.00"),
("seller_1","samsung","45000.00"),("seller_2","samsung","45500.00"))

Answer: (penalty regime: 0 %)

```
# Nested tuples representing seller information (seller_name, item_
sellers = [
    ("seller_1", "samsung", "45000.00"),
    ("seller_2", "samsung", "45500.00"),
    ("seller_3", "samsung", "44700.00"),
    ("seller_4", "samsung", "43900.00"),
    ("seller_5", "samsung", "44100.00")
]

# Sort the sellers based on item cost in ascending order
sorted_sellers = sorted(sellers, key=lambda x: x[2])

# Output the sorted seller information
print(tuple(sorted_sellers))
```

	Input	Expected	Got	
✔	seller_1 samsung 45000.00 seller_2 samsung 45500.00 seller_3 samsung 44700.00 seller_4 samsung 43900.00 seller_5 samsung 44100.00	(('seller_4', 'samsung', '43900.00'), ('seller_5', 'samsung', '44100.00'), ('seller_3', 'samsung', '44700.00'), ('seller_1', 'samsung', '45000.00'), ('seller_2', 'samsung', '45500.00'))	(('seller_4', 'samsung', '43900.00'), ('seller_5', 'samsung', '44100.00'), ('seller_3', 'samsung', '44700.00'), ('seller_1', 'samsung', '45000.00'), ('seller_2', 'samsung', '45500.00'))	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

Write a program to unpack the following tuple into variables depends on the length of tuple (Max length = 10) and display each values separately.

Sample Input:

4

10

30

40

60

Sample Output:

a=10

b=30

c=40

d=60

Answer: (penalty regime: 0 %)

```
n=int(input())
List1=['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','
List=[]
for i in range(n):
    List.append(int(input()))
for i in range(len(List)):
    print("%s=%d"%(List1[i],List[i]))
```

	Input	Expected	Got	
✔	4 10 30 40 60	a=10 b=30 c=40 d=60	a=10 b=30 c=40 d=60	✔

	Input	Expected	Got	
✔	9 15 60 75 85 90 70 35 25 45	a=15 b=60 c=75 d=85 e=90 f=70 g=35 h=25 i=45	a=15 b=60 c=75 d=85 e=90 f=70 g=35 h=25 i=45	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

Create a tuple t1 with numbers 1 to 5, t2 with 6 to 10 and t3 with a string "REC".

Concatenate t1 and t2 and print the result.

Repeat the t3 10 times without using any looping statements.

Expected output:

(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
('REC', 'REC', 'REC', 'REC', 'REC', 'REC', 'REC', 'REC', 'REC', 'REC')

Answer: (penalty regime: 0 %)

```
# Create tuples t1, t2, and t3
t1 = tuple(range(1, 6))
t2 = tuple(range(6, 11))
t3 = ("REC",)

# Concatenate t1 and t2
concatenated_tuple = t1 + t2
print(concatenated_tuple)

# Repeat t3 10 times
repeated_tuple = t3 * 10
print(repeated_tuple)
```

	Expected	Got	
✔	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10) ('REC', 'REC', 'REC', 'REC', 'REC', 'REC', 'REC', 'REC', 'REC', 'REC')	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10) ('REC', 'REC', 'REC', 'REC', 'REC', 'REC', 'REC', 'REC', 'REC', 'REC')	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ Week-08_MCQ

Jump to...

Week-09_MCQ ►