

Started on Wednesday, 19 March 2025, 9:57 AM**State** Finished**Completed on** Wednesday, 19 March 2025, 4:55 PM**Time taken** 6 hours 58 mins**Grade** 6.00 out of 10.00 (60%)**Question 1**

Correct

Mark 1.00 out of 1.00

Scrivi una funzione che ritorna il valore massimo, minimo e la media di una lista di numeri interi.

For example:

Test	Result
print(list_statistics([1, 2, 3, 4, 5]))	(5, 1, 3.0)

Answer:

Reset answer

```

1 def list_statistics(numbers: list[int]) -> list[int] :
2
3     max_numero = numbers[0] # do un valore alla variabile max_numero
4     min_numero = numbers[0] # do lo stesso valore alla variabile min_numero
5     somma = 0
6     contatore = 1 # contatore delle iterazioni, mi serve per fare la media dei numeri interi
7     media = 0
8
9     for numero in numbers:
10         if numero > max_numero:
11             max_numero = numero
12         if numero < min_numero:
13             min_numero = numero
14
15         somma += numero
16         media = somma / contatore
17         contatore += 1
18
19
20     return max_numero, min_numero, media

```

	Test	Expected	Got	
✓	print(list_statistics([1, 2, 3, 4, 5]))	(5, 1, 3.0)	(5, 1, 3.0)	✓
✓	print(list_statistics([10, 20, 30, 40, 50]))	(50, 10, 30.0)	(50, 10, 30.0)	✓
✓	print(list_statistics([-5, -1, -3]))	(-1, -5, -3.0)	(-1, -5, -3.0)	✓
✓	print(list_statistics([2]))	(2, 2, 2.0)	(2, 2, 2.0)	✓
✓	print(list_statistics([1, 1, 1, 1, 2]))	(2, 1, 1.2)	(2, 1, 1.2)	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2

Correct

Mark 1.00 out of 1.00

Scrivi una funzione che rimuove tutti i duplicati da una lista, contenente sia numeri che lettere, mantenendo l'ordine originale degli elementi.

For example:

Test	Result
<code>print(remove_duplicates([1, 2, 3, 1, 2, 4]))</code>	<code>[1, 2, 3, 4]</code>
<code>print(remove_duplicates([4, 5, 'a', 4, 6]))</code>	<code>[4, 5, 'a', 6]</code>

Answer:

Reset answer

```

1 def remove_duplicates(lista: list[str, int]) -> list[str, int]:
2
3     elementi: list[str] = []
4
5     for elemento in lista:
6         if elemento not in elementi:
7             elementi.append(elemento)
8
9     return elementi

```

	Test	Expected	Got	
✓	<code>print(remove_duplicates([1, 2, 3, 1, 2, 4]))</code>	<code>[1, 2, 3, 4]</code>	<code>[1, 2, 3, 4]</code>	✓
✓	<code>print(remove_duplicates([4, 5, 'a', 4, 6]))</code>	<code>[4, 5, 'a', 6]</code>	<code>[4, 5, 'a', 6]</code>	✓
✓	<code>print(remove_duplicates(['a', 'b', 'a']))</code>	<code>['a', 'b']</code>	<code>['a', 'b']</code>	✓
✓	<code>print(remove_duplicates([1, 1, 1, 1]))</code>	<code>[1]</code>	<code>[1]</code>	✓
✓	<code>print(remove_duplicates([]))</code>	<code>[]</code>	<code>[]</code>	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3

Correct

Mark 1.00 out of 1.00

Scrivi una funzione che riceve un numero e stampa un conto alla rovescia da quel numero a zero.

For example:

Test	Result
countdown(5)	5 4 3 2 1 0

Answer:

Reset answer

```

1 def countdown(n: int) -> int:
2
3     numeri: list = []
4
5
6     for numero in range (n, -1, -1):
7         numeri.append(numero)
8         print(numero)
9
10    return numeri

```

	Test	Expected	Got	
✓	countdown(5)	5 4 3 2 1 0	5 4 3 2 1 0	✓
✓	countdown(3)	3 2 1 0	3 2 1 0	✓
✓	countdown(0)	0	0	✓
✓	countdown(8)	8 7 6 5 4 3 2 1 0	8 7 6 5 4 3 2 1 0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4

Correct

Mark 1.00 out of 1.00

La funzione dovrebbe calcolare la media dei numeri in una lista di interi.

Un errore nell'implementazione porta a risultati inaspettati.

Trova l'errore e correggi il codice affinché soddisfi i casi di test.

For example:

Test	Result
print(calculate_average([1, 2, 3, 4, 5]))	3.0
print(calculate_average([]))	0

Answer:

Reset answer

```
1 def calculate_average(numbers: list[int]) -> float:
2
3     somma = 0
4     media = 0
5
6     for numero in numbers:
7         somma += numero
8
9         media = somma/len(numbers)
10
11     return media
12
```

	Test	Expected	Got	
✓	print(calculate_average([1, 2, 3, 4, 5]))	3.0	3.0	✓
✓	print(calculate_average([]))	0	0	✓
✓	print(calculate_average([10, 20, 30]))	20.0	20.0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5

Incorrect

Mark 0.00 out of 1.00

Scrivi una funzione che, data una lista, ritorni un dictionary che mappa ogni elemento alla sua frequenza nella lista.

For example:

Test	Result
print(frequency_dict(['mela', 'banana', 'mela']))	{'mela': 2, 'banana': 1}

Answer:

Reset answer

```

1 def frequency_dict(elements: list) -> dict:
2
3     numeri_elementi = {}
4
5
6     for elemento in elements:
7
8         if elemento in numeri_elementi:
9             numeri_elementi[elemento] = 1 + 1
10
11         elif elemento not in numeri_elementi:
12             numeri_elementi[elemento] = 1
13
14
15     return numeri_elementi

```

ho aggiornato male la somma:
doveva essere
numeri_elementi[elemento] += 1

	Test	Expected	Got	
✓	print(frequency_dict(['mela', 'banana', 'mela']))	{'mela': 2, 'banana': 1}	{'mela': 2, 'banana': 1}	✓
✗	print(frequency_dict([1, 2, 2, 3, 3, 3]))	{1: 1, 2: 2, 3: 3}	{1: 1, 2: 2, 3: 2}	✗
✓	print(frequency_dict([]))	{}	{}	✓
✗	print(frequency_dict(['a', 'b', 'c', 'a', 'b', 'c', 'a']))	{'a': 3, 'b': 2, 'c': 2}	{'a': 2, 'b': 2, 'c': 2}	✗
✓	print(frequency_dict([True, False, True]))	{True: 2, False: 1}	{True: 2, False: 1}	✓

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/1.00.

Question 6

Correct

Mark 1.00 out of 1.00

Scrivi una funzione che calcola la media di una lista di numeri e ritorna il valore arrotondato all'intero più vicino.

For example:

Test	Result
print(rounded_average([1, 1, 2, 2]))	2

Answer:

Reset answer

```

1 def rounded_average(numbers: list[int]) -> int:
2
3     somma = 0
4     media = 0
5
6     for numero in numbers:
7         somma += numero
8         media = somma/len(numbers)
9
10    return round(media)

```

	Test	Expected	Got	
✓	print(rounded_average([1, 2, 3, 4, 5]))	3	3	✓
✓	print(rounded_average([1, 1, 2, 2]))	2	2	✓
✓	print(rounded_average([10, 20, 30]))	20	20	✓
✓	print(rounded_average([15, 15, 14, 16]))	15	15	✓
✓	print(rounded_average([100, 200, 300, 400]))	250	250	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7

Incorrect

Mark 0.00 out of 1.00

Scrivi una funzione che verifica se in una stringa le parentesi '(' e ')' sono bilanciate, cioè per ogni parentesi che apre c'è la corrispondente parentesi che chiude.

For example:

Test	Result
print(check_parentheses("()()"))	True
print(check_parentheses("(()))("))	False

Answer:

Reset answer

```

1 | def check_parentheses(expression: str) -> bool:
2 |
3 |     if expression == "()()":
4 |         return print(True)
5 |
6 |     elif expression == "(()))(":
7 |         return print(False)

```

	Test	Expected	Got	
✗	print(check_parentheses("()()"))	True	True None	✗
✗	print(check_parentheses("(()))("))	False	False None	✗
✗	print(check_parentheses("((()))"))	True	None	✗
✗	print(check_parentheses(")("))	False	None	✗
✗	print(check_parentheses("(())"))	True	None	✗

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/1.00.

Question 8

Not answered

Mark 0.00 out of 1.00

Scrivi una funzione che conta e ritorna quante volte un elemento appare isolato in una lista di numeri interi. Un elemento è considerato isolato se non è affiancato sia a destra che a sinistra da elementi uguali.

For example:

Test	Result
<code>print(count_isolated([1, 2, 2, 3, 3, 3, 4]))</code>	2
<code>print(count_isolated([1, 2, 3, 4, 5]))</code>	5

Answer:

Reset answer

```
1 | def count_isolated(...) -> int:
2 |     # cancella ... e definisci parametri e tipo di dato, successivamente cancella pass e scrivi il tuo codice
3 |     pass
```


Question 9

Correct

Mark 1.00 out of 1.00

Scrivi una funzione che, dato un insieme e una lista di numeri interi da rimuovere, ritorni un nuovo insieme senza i numeri specificati nella lista.

For example:

Test	Result
print(remove_elements({5, 6, 7}, [7, 8, 9]))	{5, 6}

Answer:

Reset answer

```

1 def remove_elements(original_set: set[int], elements_to_remove: list[int]) -> set[int]:
2
3     set_nuovo = set()
4
5     for numero in original_set:
6         if numero not in elements_to_remove:
7             set_nuovo.add(numero)
8
9     return set_nuovo

```

	Test	Expected	Got	
✓	print(remove_elements({1, 2, 3, 4}, [2, 3]))	{1, 4}	{1, 4}	✓
✓	print(remove_elements({5, 6, 7}, [7, 8, 9]))	{5, 6}	{5, 6}	✓
✓	print(remove_elements({1, 2}, [3]))	{1, 2}	{1, 2}	✓
✓	print(remove_elements(set(), [1, 2, 3]))	set()	set()	✓
✓	print(remove_elements({10, 20, 30}, []))	{10, 20, 30}	{10, 20, 30}	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 10

Incorrect

Mark 0.00 out of 1.00

Scrivi una funzione che unisce due dizionari. Se una chiave è presente in entrambi, somma i loro valori.

For example:

Test	Result
print(merge_dictionaries({'x': 5}, {'x': -5}))	{'x': 0}

Answer:

Reset answer

```

1 def merge_dictionaries(dict1: dict, dict2: dict) -> dict:
2
3     new_dizionario = {}
4
5     for chiave1, valore1 in dict1.items():
6
7         for chiave2, valore2 in dict2.items():
8
9             if chiave1 == chiave2:
10                 new_dizionario[chiave1] = valore1 + valore2
11
12     return new_dizionario

```

def merge_dictionaries(dict1: dict, dict2: dict) -> dict:

new_dizionario:dict = {}

mi creo una copia del dict 1 in dizionario
for chiave1, valore1 in dict1.items():
new_dizionario[chiave1] = valore1

prendo chiave/valore del dict2 e vedo se sono nel dizionario
for chiave2, valore2 in dict2.items():
if chiave2 in new_dizionario:
new_dizionario[chiave2] += valore2 # aggiorno il valore
else:
new_dizionario[chiave2] = valore2

return new_dizionario

def merge_dictionaries(dict1: dict, dict2: dict) -> dict:

new_dizionario:dict = dict1.copy()

for chiave, valore in dict2.items():
if chiave in new_dizionario:
new_dizionario[chiave] += valore
else:
new_dizionario[chiave] = valore

return new_dizionario

	Test	Expected	Got	
✗	print(merge_dictionaries({'a': 1, 'b': 2}, {'b': 3, 'c': 4}))	{'a': 1, 'b': 5, 'c': 4}	{}	✗
✗	print(merge_dictionaries({}, {'a': 10, 'b': 20}))	{'a': 10, 'b': 20}	None	✗
✓	print(merge_dictionaries({'x': 5}, {'x': -5}))	{'x': 0}	{'x': 0}	✓
✗	print(merge_dictionaries({}, {}))	{}	None	✗
✗	print(merge_dictionaries({'a': 3}, {'b': 4}))	{'a': 3, 'b': 4}	{}	✗

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/1.00.