

Tempo per la soluzione 1 ora

risposta corretta = 3 punti, risposta sbagliata = -1 punti, nessuna risposta = 0 punti

Risposte:

Nome: _____ Matr: _____

1	2	3	4	5	6	7	8	9	10	11

1. Qual è l'output del seguente codice?

```
x = [i for i in range(3)]
y = x
z = x[:]
y.append(3)
z.append(4)
print(x, y, z)
```

- A) [0, 1, 2, 3][0, 1, 2, 3][0, 1, 2, 3, 4]
 B) [0, 1, 2][0, 1, 2][0, 1, 2, 4]
 C) [0, 1, 2, 3][0, 1, 2, 3][0, 1, 2, 4]
 D) [0, 1, 2][0, 1, 2, 3][0, 1, 2, 4]

2. Quale sarà l'output?

```
data = {'a': [1, 2, 3], 'b': [4, 5]}
result = [val for sublist in
          data.values() for val in sublist if
          val % 2 == 1]
print(result)
```

- A) [1, 3, 5]
 B) [1, 2, 3, 4, 5]
 C) [1, 3]
 D) [2, 4]

3. Trova l'errore nel codice seguente:

```
d = {'x': 1, 'y': 2}
d[['z']] = 3
print(d)
```

- A) Nessun errore, stampa {'x': 1, 'y': 2, 'z': 3}
 B) Solleva TypeError perché le liste non possono essere usate come chiave
 C) Solleva KeyError perché 'z' non esiste
 D) Solleva SyntaxError a causa della notazione errata

4. Cosa stampa il seguente codice?

```
def func(n):
    if n == 0:
        return 0
    elif n % 2 == 0:
        return func(n//2)
    else:
        return 1 + func(n//2)

print(func(21))
```

A) 3

B) 4

C) 2

D) 5

5. Quale sarà il valore di x alla fine?

```
x = 0
for i in range(1, 6):
    x += (i % 2) * i
print(x)
```

- A) 6
 B) 9
 C) 5
 D) 15

6. Cosa stampa questo algoritmo?

```
a = [3, 1, 4, 1, 5]
b = []
for i in range(len(a)):
    b.append(sum(a[:i+1]))
print(b[-1])
```

- A) 14
 B) 15
 C) 13
 D) 10

7. Cosa succede eseguendo questo script se data.txt esiste e contiene 3 righe?

```
with open("data.txt", "r") as f:
    lines = [line.strip() for line in
             f if line.strip()]
    content = f.read()
    print(len(lines),
          len(content.splitlines()))
```

- A) Stampa 3 3
 B) Solleva eccezione
 C) Stampa 3 0
 D) Dipende dai contenuti di data.txt

8. Cosa stampa il seguente codice?

```
def make_incrementor(n):
    return lambda x: x + n

f = make_incrementor(5)
print(f(f(2)))
```

- A) 12
B) 9
C) 7
D) 10

9. Qual è l'output?

```
def decorator(func):
    def wrapper(*args):
        print("Before")
        result = func(*args)
        print("After")
        return result
    return wrapper

@decorator
def greet(name):
    print(f"Hello, {name}")

greet("Alice")
```

- A) Before Hello, Alice After
B) Hello, Alice
C) Before After
D) Before\nHello, Alice\nAfter

10. Cosa stampa il seguente codice?

```
class Base:
    def __init__(self):
        self.value = 5

class Derived(Base):
    def __init__(self):
        super().__init__()
        self.value += 1

print(Derived().value)
```

- A) 6
B) 5
C) None
D) Errore di tipo

11. Cosa stampa il seguente codice?

```
class Animal:
    def __init__(self, name):
        self.name = name

    def speak(self):
        return f"{self.name} makes a
sound"

class Dog(Animal):
    def speak(self):
        return f"{self.name} barks"

class Cat(Animal):
    def speak(self):
```

```
        return f"{self.name} meows"

def animal_sound(animal: Animal):
    if isinstance(animal, Dog):
        print(animal.speak().upper())
    elif isinstance(animal, Cat):
        print(animal.speak().replace("meows",
"purrrs"))
    else:
        print(animal.speak())

animals = [Dog("Fido"),
Cat("Whiskers"), Animal("Creature")]
for a in animals:
    animal_sound(a)
```

A)

FIDO BARKS

Whiskers purrrs

Creature makes a sound

B)

Fido barks

Whiskers meows

Creature makes a sound

C)

FIDO BARKS

WHISKERS MEOWS

CREATURE MAKES A SOUND

D)

fido barks

whiskers purrrs

creature makes a sound