

## Q1 We all saw this coming

20 Points

Para o código abaixo, determine o valor e o tipo da variável mencionada em cada item.

Insira o valor na caixa de texto na forma como seria devolvido por Python e selecione o tipo das escolhas.

```
a = 10
b = 31

c = 100 if a > b else -30.0
d = 0

while c > 0:
    c -= 10
    d += 5

print(c * 3)
e = round(b / 4)

f = [a, b, c, d]
g = f.append(e)
h = f[2:]

i = {a: f, b: 22, d: d}
j = i.get(c, (1, 2, 3))
```

### Q1.1 a

2 Points

**Valor**

10

**Tipo**

- ☒ int
- ☐ float
- ☐ NoneType
- ☐ bool
- ☐ str
- ☐ list
- ☐ tuple
- ☐ dict

**Q1.2 b**

2 Points

**Valor**

31

**Tipo**

- ☒ int
- ☐ float
- ☐ NoneType
- ☐ bool
- ☐ str
- ☐ list
- ☐ tuple
- ☐ dict

**Q1.3 c**

2 Points

**Valor**

-30.0

**Tipo**

- ☐ int
- ☒ float
- ☐ NoneType
- ☐ bool
- ☐ str
- ☐ list
- ☐ tuple
- ☐ dict

**Q1.4 d**

2 Points

**Valor**

0

**Tipo**

- ☒ int
- ☐ float
- ☐ NoneType
- ☐ bool
- ☐ str
- ☐ list
- ☐ tuple
- ☐ dict

**Q1.5 e**

2 Points

**Valor**

8

**Tipo**

- ☒ int
- ☐ float
- ☐ NoneType
- ☐ bool
- ☐ str
- ☐ list
- ☐ tuple
- ☐ dict

**Q1.6 f**

2 Points

**Valor**

[10, 31, -30.0, 0, 8]

**Tipo**

- ☐ int
- ☐ float
- ☐ NoneType
- ☐ bool
- ☐ str
- ☒ list
- ☐ tuple
- ☐ dict

**Q1.7 g**

2 Points

**Valor**

None

**Tipo**

- ☐ int
- ☐ float
- ☒ NoneType
- ☐ bool
- ☐ str
- ☐ list
- ☐ tuple
- ☐ dict

**Q1.8 h**

2 Points

**Valor**

[-30.0, 0, 8]

**Tipo**

- ☐ int
- ☐ float
- ☐ NoneType
- ☐ bool
- ☐ str
- ☒ list
- ☐ tuple
- ☐ dict

**Q1.9 i**

2 Points

**Valor**

```
{10: [10, 31, -30.0, 0, 8], 31: 22, 0: 0}
```

**Tipo**

- ☐ int
- ☐ float
- ☐ NoneType
- ☐ bool
- ☐ str
- ☐ list
- ☐ tuple
- ☒ dict

**Q1.10 j**

2 Points

**Valor**

```
(1, 2, 3)
```

**Tipo**

- ☐ int
- ☐ float
- ☐ NoneType
- ☐ bool
- ☐ str
- ☐ list
- ☒ tuple
- ☐ dict

## Q2 The check, please

20 Points

Escreva uma função `weird_checker(func, input_list)` que:

- recebe uma **lista de números** `input_list` como entrada. Por exemplo, `[1, 2, 3, 4, 5, 6]` e `[1, 2]` seriam entradas válidas.
- retorna uma nova lista que contém tuplas `(x, y)` em que `x`, `y` são elementos consecutivos da lista que têm a propriedade de que `func(x, y)` retorna `True`.

Por exemplo, chamar `weird_checker` na lista

`[1, 2, 3, 6, 9, 10, 12, 15]` com uma função que checa se o número é divisível por 3 deveria fornecer

`[(3, 6), (6, 9), (12, 15)]` como saída.

Uma vez que você tenha escrito a função e testado se é correta, use seu programa para avaliar as chamadas de função abaixo e escreva o valor de retorno em cada caso.

### Q2.1

5 Points

```
f = lambda x, y: x ** 3 > y + 70

input_lst = [
-893, -452, -822, 574, -166, -455,
-780, 717, 845, 791, -922, -224,
98, 151, -320, -303, 745, -894,
-674, -724, -310, 148, -317, 437,
-498, -665, -997, 855, -107, 584,
-821, 799, 223, -227, -857, -987,
-354, 499, -82, -792, 854, -911,
-811, 366, -709, -742, 618, 856,
-957, -403
]

out = weird_checker(f, input_lst)
print(out[10])
```

Valor de `out[10]`

(584, -821)

**Q2.2**

5 Points

```
f = lambda x, y: (x ** 3) / (y ** 2) >= x * y

input_lst = [
    729, -212, 552, 823, -139, -918,
    -470, 977, 47, -5, -171, 880,
    605, 699, -379, 982, -24, -267,
    194, 826, 859, -553, 33, -715,
    -423, -714, 547, -806, 266, 637,
    -487, 863, 90, 444, 659, 232,
    847, -700, -365, -798, 494, -849,
    840, 741, 400, -324, -34, 146,
    -794, -276
]

out = weird_checker(f, input_lst)
print(out[20])
```

Valor de `out[20]`

(-798, 494)

**Q2.3**

5 Points

```
f = lambda x, y: abs(x % y) <= 10

input_lst = [18, -62, 70, -98, 6,
    -60, -73, -25, -54, 49, 54, -93,
    -37, 18, -6, 99, -11, 7, 24, -48,
    -59, -50, -2, -91, 73
]

out = weird_checker(f, input_lst)
print(out[6])
```

Valor de `out[6]`

(-59, -50)

**Q2.4**

5 Points



```
f = lambda x, y: x % 4 <= y % 3

input_lst = [38, -17, 71, 95, -31,
-34, -11, -26, -49, 93, -40,
-66, -17, -94, -92, 27, 18,
79, 37, 63, -26, 92, 7, -89, 87
]

out = weird_checker(f, input_lst)
print(out[4])
```

Valor de `out[4]`

(-40, -66)

### Q3 Time for change

20 Points

Implemente a função com a especificação abaixo:

```
def make_change(amount):
    """
    Assume que amount é um inteiro > 0, em centavos
    Retorna o menor número de moedas cujos valores somam a amount
    * Moedas possíveis são 100, 50, 25, 10, 5, 1 centavos
    * Um algoritmo ganancioso funciona aqui
    """
    pass # seu código aqui
```

- um algoritmo ganancioso basicamente tenta encaixar primeiro os valores maiores de moedas e vai diminuindo o valor conforme as moedas maiores não são mais adequadas.

Por exemplo, se `amount` fosse `256`, começaríamos usando moedas de 100. Depois de duas moedas de `100`, teríamos `50` sobrando, então usaremos uma moeda de `50`. Sobrando agora `6`, não conseguimos usar moedas de `25` ou `10`, precisando apenas de uma moeda de `5` e depois uma de `1` para obter a quantidade desejada.

Nesse exemplo, seu código deve retornar `5` representando o **total** de moedas (2 de `100`, 1 de `50`, 1 de `5` e 1 de `1` centavo).

Uma vez que você tenha escrito a função e testado se é correta, use seu programa para avaliar as chamadas de função abaixo e

escreva o valor de retorno em cada caso.

### Q3.1

5 Points

Valor de `make_change(1550442)`

15509

### Q3.2

5 Points

Valor de `make_change(329294392)`

3292949

### Q3.3

5 Points

Valor de `make_change(7777777)`

77781

### Q3.4

5 Points

Valor de `make_change(15564)`

161

### Q4 1-800-273-8255 📞 (188 🇧🇷)

10 Points

Implemente a função com a especificação abaixo:

```
def map_freqs(s):  
    """  
    s é uma string não-vazia de caracteres
```

```
Retorna um dicionário que mapeia cada caractere em s que
é um dígito (0-9) ao número de vezes que ele ocorre em s.
"""

pass # seu código aqui

# Por exemplo:
print(map_freqs("12"))      # {'1': 1, '2': 1}
print(map_freqs("a23b22"))  # {'2': 3, '3': 1}
print(map_freqs("abc"))     # {}
```

Uma vez que você tenha escrito a função e testado se é correta, use seu programa para avaliar as chamadas de função abaixo e escreva o valor de retorno em cada caso.

## Q4.1

5 Points

```
msg = """
Hey!

Jenny, Jenny, who can I turn to?
You give me somethin' I can hold on to
I know you think I'm like the others before
Who saw your name and number on the wall

Jenny, I got your number
I need to make you mine
Jenny, don't change your number

867-5309
867-5309
867-5309
867-5309

Jenny, Jenny, you're the girl for me
Oh, you don't know me, but you make me so happy
I tried to call you before, but I lost my nerve
I tried my imagination, but I was disturbed

Jenny, I got your number
I need to make you mine
Jenny, don't change your number

867-5309
867-5309
867-5309
867-5309

I got it, (I got it), I got it
I got your number on the wall
I got it, (I got it), I got it
For a good time, for a good time call
```

Hey, Jenny, don't change your number  
I need to make you mine  
Jenny, I call your number

867-5309  
867-5309  
867-5309  
867-5309

Jenny, Jenny, who can I turn to?  
867-5309  
For the price of a dime I can always turn to you  
867-5309

867-5309  
867-5309  
867-5309  
867-5309

(5309) 867-5309  
(5309) 867-5309  
(5309) 867-5309

-----

Out of the 7,000,000 people in this world, there's only you  
Almost 1,000,000 words that I could say, but none of them will do  
So many years that I have lived, but it feels like I've just begun  
Out of the 7,000,000 people, baby, you're the only 1  
I've been too many places. I've seen too many faces  
I wrote too many pages; never found a love like you (Love like you)  
I don't know how to say it. It's been more than amazing  
My whole life, I've been waiting; never found a love like you (Love like you)  
Sleep, dream, you, repeat, live, die next to me  
Sleep, dream, you, repeat, sleep, dream  
Out of the 7,000,000 people in this world, there's only you  
Almost 1,000,000 words that I could say, but none of them will do  
So many years that I have lived, but it feels like I've just begun  
Out of the 7,000,000 people, baby, you're the only 1  
You were unexpected, when our lips connected  
I was resurrected; never found a love like you (love like you)  
Don't care where I'm headed when the world is ending  
'Cause you are my heaven; never found a love like you. (Love like you)  
Sleep, dream, you, repeat. Live, die next to me  
Sleep, dream, you, repeat. Sleep, dream  
Out of the 7,000,000 in this world, there's only you  
Almost a million words that I could say, but none of them will do  
So many years that I have lived, but it feels like I've just begun  
Out of the 7,000,000 people, baby, you're the only 1  
Surrounded by numbers  
(Numbers, numbers, numbers)  
You're the only 1  
(Numbers, numbers, numbers, numbers)  
You're the only 1  
You brought me back to life  
You did the unthinkable  
Yeah, you are my miracle  
Out of the 7,000,000 people in this world, there's only you

```

Almost a million words that I could say, but none of them will do
So many years that I have lived, but it feels like I've just begun
Out of the 7,000,000 people, baby, you're the only 1
Surrounded by numbers
(Numbers, numbers, numbers)
You're the only 1
(Numbers, numbers, numbers, numbers)
You're the only 1
Surrounded by numbers
(Numbers, numbers, numbers)
You're the only 1
(Numbers, numbers, numbers, numbers)
You're the only 1
""""

```

Valor de `map_freqs(msg)`

```
{'8': 21, '6': 21, '7': 29, '5': 24, '3': 24, '0':
84, '9': 24, '1': 12}
```

## Q4.2

5 Points

```

msg = ""
6604876475938242194892411578156
5938778408016097535139332871158714
8418583989471965934232094711220186
8483396947751591795330413525601230
9891013991615109032173008691413145
6208709163457923022584197207698456
4280715084237594599246610935233769
6069602714278789007547063812066503
0089131934421761047142851240003485
5909776582369402245551590042294568
2417304281465461187755171760452296
1113306016884779361534926351108731
7643039213765821972966875773893055
5082492694711801320407522758688091
8916348967699300248945174466602234
5007627912560976701720099251853671
0979519426418306753751007408993318
8684126961161162076607541511505520
3519230310450932271754203557448529
1189426223583324560290611264862964
5137580660657354712141892762662375
8257173407970341489267173864014004
6896714543910144851832164482983816
8644759221166972845763778570742709
3057608116050190443294316756256627
2852327566763373906031250239491845
6708869774735400025040216396837359
1915587540803513853344486475304810

```

```
7770677711131263791686023732455618
4983478978443019126334088601641953
8433184535749220885590262281237933
2365992719089577403827785142963546
0305357345240598025070030035105623
7625425660648870916620829821532302
8216919729904560071542738526547604
8487089840761657004049438856431953
9852250909255
""""
```

Valor de `map_freqs(msg)`

```
{'6': 124, '0': 136, '4': 124, '8': 111, '7': 125,
'5': 128, '9': 114, '3': 118, '2': 120, '1': 134}
```

## Q5 Doppelgangers

10 Points

Dado o código abaixo, escolha todas as opções verdadeiras.

```
def f(L):
    """
    L é uma lista não-vazia
    """
    try:
        M = L
        P = M[:]
        #point1#
        Lnew = L.append("")
        M = sorted(M + M)
        #point2#
    except:
        print("Nothing.")
```

☒ Em `#point1#`, `L` e `M` estão relacionados por *aliasing*

☒ Em `#point1#`, `P` é uma cópia de `L`.

☐ Em `#point2#`, `Lnew` é uma cópia de `L`.

☐ Em `#point2#`, `M` e `L` estão relacionados por *aliasing*

☒ Alguns valores iniciais de `L` (que satisfazem a especificação) levarão essa função a entrar no bloco `except`

## Q6 Oh look, another flat-pointer

20 Points

Considere o código abaixo para as classes `Coordinate` e `Coordinate3D` em Python.

```
class Coordinate:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def distance(self, other):
        x_diff_sq = (self.x-other.x)**2
        y_diff_sq = (self.y-other.y)**2
        return (x_diff_sq + y_diff_sq)**0.5

    def __str__(self):
        return "<" + str(self.x) + "," + str(self.y) + ">"

class Coordinate3D(Coordinate):
    """ Uma coordenada3D contém valores x, y, e z """
    def __init__(self, c=Coordinate(0, 0), a=0):
        """ Sets the x, y, and z values """
        super().__init__(c.x, c.y) #point1#
        self.z = a                #point2#

    def distance(self, other):
        """Retorna a distância euclidiana entre duas Coordinate3D"""
        x_diff_sq = (self.x-other.x)**2
        y_diff_sq = (self.y-other.y)**2
        z_diff_sq = (self.z-other.z)**2
        return (x_diff_sq + y_diff_sq + z_diff_sq)**0.5

    def __str__(self):
```

```
return '[' + str(self.x) + ', ' + str(self.y) + ']
```

```
c3d = Coordinate3D
```

## Q6.1

12 Points

Escolha todas as opções abaixo que são verdadeiras.

☐ `Coordinate3D` é uma superclasse de `Coordinate`

☒ `Coordinate3D` é uma subclasse de `Coordinate`

☐ `c3d` é uma superclasse de `Coordinate3D`

☒ `c3d` é uma subclasse de `Coordinate3D`

☒ O método `distance` em `Coordinate` é sobrescrito em `Coordinate3D`

☐ Imprimir a instância de `Coordinate3D` criada com `Coordinate3D(Coordinate(1,2), 3)` mostra `[1,2,3]`

☒ `#point1#` é semanticamente equivalente a dizer `self.x, self.y = c.x, c.y`

☒ `#point2#` adiciona um atributo a uma instância de `Coordinate3D`

## Q6.2

2 Points

Considere agora o código abaixo para os próximos quatro itens, em adição ao apresentado acima:

```
p1 = Coordinate(3, 4)
p2 = Coordinate3D(Coordinate(2, 3), 6)
orig1 = Coordinate(0, 0)
orig2 = Coordinate3D(orig1, 0)

print(p1.distance(orig1)) #point3#
```



```
print(p2.distance(orig2)) #point4#  
  
print(Coordinate3D.__str__(p1)) #point5#  
  
print(p1.__str__() == p2.__str__()) #point6#
```

Qual o valor impresso pelo comando `print` em `#point3#`?  
Escreva `error` se ocorreria um erro.

5.0

### Q6.3

2 Points

Qual o valor impresso pelo comando `print` em `#point4#`?  
Escreva `error` se ocorreria um erro.

7.0

### Q6.4

2 Points

Qual o valor impresso pelo comando `print` em `#point5#`?  
Escreva `error` se ocorreria um erro.

[3, 4]

### Q6.5

2 Points

Qual o valor impresso pelo comando `print` em `#point6#`?  
Escreva `error` se ocorreria um erro.

False



# Teste Final

● **UNGRADED**

## STUDENT

João Vitor Izael Souza

## TOTAL POINTS

- / 100 pts

### QUESTION 1

We all saw this coming  20 pts

- 1.1 a 2 pts
- 1.2 b 2 pts
- 1.3 c 2 pts
- 1.4 d 2 pts
- 1.5 e 2 pts
- 1.6 f 2 pts
- 1.7 g 2 pts
- 1.8 h 2 pts
- 1.9 i 2 pts
- 1.10 j 2 pts

### QUESTION 2

The check, please  20 pts

- 2.1 (no title) 5 pts
- 2.2 (no title) 5 pts
- 2.3 (no title) 5 pts
- 2.4 (no title) 5 pts

### QUESTION 3

Time for change  20 pts

- 3.1 (no title) 5 pts
- 3.2 (no title) 5 pts
- 3.3 (no title) 5 pts
- 3.4 (no title) 5 pts

### QUESTION 4

1-800-273-8255  (188 ) 10 pts

4.1 (no title) 5 pts

4.2 (no title) 5 pts

**QUESTION 5**

Doppelgangers  10 pts

**QUESTION 6**

Oh look, another flat-pointer  20 pts

6.1 (no title) 12 pts

6.2 (no title) 2 pts

6.3 (no title) 2 pts

6.4 (no title) 2 pts

6.5 (no title) 2 pts