

Nome do Aluno: \_\_\_\_\_ Nota: \_\_\_\_\_

1. Para cada fragmento de código na tabela abaixo, indique se o código causa um erro. Caso seja executado sem erro, informe a saída. Caso haja um erro, explique a causa do erro. (1,0)

Código	Erro? (S/N)	Saída ou Causa do Erro
<code>print(7 / 4)</code>	N	1 ou 1.75
<code>print(7 % 4)</code>	N	3
<code>L = [1, 2, 3, 4]</code> <code>print(L[4])</code>	S	IndexError: list index out of range
<code>s = 'time'</code> <code>s[0] = 'l'</code> <code>print(s[0])</code>	S	TypeError: 'str' object does not support item assignment
<code>letters = ['a', 'b', 'c']</code> <code>letters[0] = 321</code> <code>print(letters)</code>	N	[321, 'b', 'c']

2. Considere o código abaixo. Ao fim da execução, quais são os valores que cada variável se refere? (1,0)

```
a = ['a', 'b', 'c']
a1 = a
a2 = a[:]
b = ['a', 'b', 'c']
b1 = b
b2 = b[:]
a.insert(0, 'z')
b = ['z'] + b
```

Variável	Valor
a	['z', 'a', 'b', 'c']
a1	['z', 'a', 'b', 'c']
a2	['a', 'b', 'c']
b	['z', 'a', 'b', 'c']
b1	['a', 'b', 'c']
b2	['a', 'b', 'c']

3. No código abaixo, traduza a estrutura de repetição **for** em um **while** que execute a mesma tarefa: (0,5)

<code>for i in range(1,10):</code> <code>    print "i = ", i</code>	<code>i = 1</code> <code>while i&lt;10:</code> <code>    print "i = ", i</code> <code>    i+=1</code>
--	--

4. No código abaixo, traduza a estrutura de repetição **while** em um **for** que execute a mesma tarefa: (0,5)

<code>i = 20</code> <code>while (i &gt; 0):</code> <code>    print "i = ", i</code> <code>    i -= 1</code>	<code>for i in range(20,0,-1):</code> <code>    print "i = ", i</code>
--	---

5. O que é impresso na tela como resultado da execução dos códigos abaixo? (1,0)

<u>a)</u> <code>x = 1</code> <code>while x &lt; 5:</code> <code>    print(x)</code> <code>    x = x + 2</code>	1 3	<u>b)</u> <code>words = ['A', 'short', 'list']</code> <code>print(len(words))</code> <code>for s in words:</code> <code>    print(len(s))</code>	3 1 5 4
<u>c)</u> <code>for x in [30, 40]:</code> <code>    for y in [1, 2, 3]:</code> <code>        print(x+y)</code>	31 32 33 41 42 43	<u>d)</u> <code>L1 = ['hello']</code> <code>L2 = L1</code> <code>L2[0] = 'bye'</code> <code>print L1</code>	['bye']

6. Escreva um programa que, dada uma lista de valores **L**, armazena e imprime os valores da lista sem repetições de elementos. Por exemplo, no caso de **L=[1,2,3,4,1,3,5,7]**, ao final da execução, o programa deve imprimir **[1,2,3,4,5,7]** (2,0)

```
S = []
for e in L:
    if not(e in S):
        S.append(e)
print S
```

7. Escreva um programa que, dada uma lista de valores **L** e um valor pedido ao usuário **num**, gera e imprime uma lista com os índices das posições em que **num** aparece em **L**. Por exemplo, no caso de **L=[1,2,3,4,1,3,5,7]** e **num=3** ao final da execução, o programa deve imprimir **[2,5]**. No caso do elemento não estar presente na lista, uma mensagem deverá ser exibida ao usuário, informando-o. (2,0)

```
num = input("Digite um valor para procurar na lista:")
S = []
i = 0
tamanho = len(L)
while i<tamanho:
    if L[i]==num:
        S.append(i)
    i+=1
if (len(S)>0):
    print S
else:
    print "O valor não foi encontrado na lista"
```

8. Escreva um programa que, dadas duas listas de valores **L** e **S**, gera e imprime uma nova lista com todos os elementos de **L** que não estão em **S**. Por exemplo, no caso de **L=[1,2,3,4,1,3,5,7]** e **S=[1,3,5]** ao final da execução, o programa deve imprimir **[2,4,7]**. (2,0)

```
T = [x for x in L if not(x in S)]
print T
```