

SIMULADO DE PROVA – Raciocínio Algorítmico
Prof. André Hochuli

Este simulado tem por objetivo apresentar ao estudante o nível de complexidade das questões da prova. Evidente que novos desafios e questões lógicas serão apresentados, no entanto, seguindo este tipo de apresentação / resolução. As listas de exercícios são uma boa base de referência para o estudante estar bem-preparado para a prova.

Basicamente as questões podem ser do tipo:

- Análise de código: É informado um código e pede-se para encontrar erros, informar a saída ou completar o código
- Implementação: Uma breve implementação deve ser fornecida

Vale lembrar que a prova será individual e deve ser resolvida a mão. Não é permitida a consulta a qualquer tipo de dispositivo, computador etc.

A prova terá duração de 1:30 minutos. O número de questões será dimensionado para esse tempo.

1. Informe o que será impresso na tela.

<pre>vet = [10,20,-50,40,60] for i in range (5): print(vet[i], vet[4-i])</pre>	
-------------------------------------------------------------------------------------	--

2. Ao implementar um código para encontrar o menor elemento, alguns erros foram cometidos. Reescreva o código correto no quadrante ao lado

<pre>vet = [10,20,12,40,60,2,60] menor = 0 for i in range(1,5): if vet[i] > menor: menor = i else vet[i] = menor</pre>	
-----------------------------------------------------------------------------------------------------------------------------------------------------	--

3. Marque qual a saída (print) do código abaixo:

```
vet = []
n=0
while n < 100:
    if n % 3 == 0:
        vet.append(n)
    n += 5
t = len(vet)
for i in range(t-1,-1,-1):
    print(vet[i],end=' ')
```

- 1 2 3 4 5 6
- 15,30,45,60,75,90
- 3,6,9,12....93,95,99
- 90,75,60,45,30,15
- -1,0,1

4. Complete o código a seguir. Você deve inferir o que é solicitado a partir dos textos disponíveis e da lógica pré-implementada:

```
vet = []

n = input('Digite o tamanho do vetor: ')

for i in _____ :
    vet.append( random.randint( __ , __ ))

for i in _____ :

    if vet[ _____ ] _____ vet [ _____ ]:
        print("O números, ", vet[ _____ ], vet [ _____ ], "são sequenciais no vetor")
```

5. Complete a matriz abaixo para que o print seja "OK"

```
mat = [[ _____ , _____ , _____ ],
        [ _____ , _____ , _____ ],
        [ _____ , _____ , _____ ]]

soma = 0

if mat[0][1] % 3 == 1:
    soma += 1

if mat[1][1] == mat[0][2]:
    soma += 1

if mat[1][2] > mat[0][0]:
    soma += 1

if mat[2][1] > mat[1][0]:
    soma += 1

if soma == 4:
    print("OK")
else:
    print("FALHOU")
```

6. Defina um trecho de código que crie uma matriz 100x100 com números aleatórios
7. Dado uma matriz 60x20, defina um código que informe a média das colunas
8. Defina um trecho de código imprima a diagonal principal de uma matriz NxN qualquer, sendo N definido pelo usuario
9. Implemente uma **função** que retorne um o maior **elemento** de um vetor
10. Implemente uma **função** que retorne um vetor contendo N elementos de Fibonacci em um vetor, sendo N um parâmetro da função.