

**Prova 2**

**26/11/2016**

**10:00 –12:00h**

**Questão 1**

**(14 pontos)**

Considere o programa abaixo:

```
import numpy as np

def f1( a, b ):
    for i in range( a, b ):
        print( i, end=' ')
    print()

def f2( a, b ):
    i = a
    while i >= b:
        print( i, end=' ')
        i = i - 1
    print()

def f3( x, y ):
    a = np.empty( x, dtype=int )
    for i in range( 0, x ):
        a[i] = y
    return a

def f4( x ):
    for i in range( 0, len(x) ):
        print( x[i], end=' ')
    print()

def f5( x ):
    y = 0
    for i in range( 0, len(x) ):
        y = y + x[i]
    return y

print('f1( 1, 10 )')
f1( 1, 10 )
print('f2( 8, 2 )')
f2( 8, 2 )
print('f2( 2, 8 )')
f2( 2, 8 )
print('m = f3( 5, 1 )')
m = f3( 5, 1 )
print('f4( m )')
f4( m )
print('f5( m )')
f5( m )
print('print( f5( m ) )')
print( f5( m ) )
```

Coloque dentro dos parênteses da coluna à esquerda a letra correspondente à coluna da direita, de acordo com o texto que será escrito por esse programa nas linhas em branco correspondentes:

<b>( I )</b>	<code>f1( 1, 10 )</code>	<b>(A)</b>	Nenhuma saída
<b>( B )</b>	<code>f2( 8, 2 )</code>	<b>(B)</b>	8 7 6 5 4 3 2
<b>( A )</b>	<code>f2( 2, 8 )</code>	<b>(C)</b>	8 7 6 5 4 3
<b>( A )</b>	<code>m = f3( 5, 1 )</code>	<b>(D)</b>	5
<b>( A )</b>	<code>f4( m )</code>	<b>(E)</b>	4
<b>( K )</b>	<code>f5( m )</code>	<b>(F)</b>	2 3 4 5 6 7 8
<b>( A )</b>	<code>print( f5( m ) )</code>	<b>(G)</b>	2 3 4 5 6 7
<b>( D )</b>		<b>(H)</b>	1 2 3 4 5 6 7 8 9 10
		<b>(I)</b>	1 2 3 4 5 6 7 8 9
		<b>(K)</b>	1 1 1 1 1
		<b>(L)</b>	1 1 1 1
		<b>(M)</b>	0 0 0 0 0

## Questão 2

(7 pontos)

Escreva uma função em Python que recebe como parâmetros um arranjo **A** de números inteiros e um valor inteiro  $i_0$  indicando um índice qualquer do arranjo. Essa função deverá procurar o maior elemento do arranjo a partir do índice  $i_0$ , e retornar o índice desse maior elemento.

Exemplo de uso da função:

```
v = [1, 4, 0, 2, 5]
print( v )

for i in range( 0, 4 ):
    j = maior( v, i )
    if i != j:
        # Troca o conteúdo de v[i] com o de v[j]
        v[i], v[j] = v[j], v[i]
    print( v )
```

Obs.: para obter o número de elementos do arranjo **A** dentro da função `maior()`, você deve usar a função `len( A )`. Segue a saída de execução do programa acima, contando que a função já foi implementada no início do programa:

```
[1, 4, 0, 2, 5]
[5, 4, 0, 2, 1]
[5, 4, 0, 2, 1]
[5, 4, 2, 0, 1]
[5, 4, 2, 1, 0]
```

Solução:

```
def maior( A, i0 ):
    i_maior = i0
    for i in range( i0+1, len(A) ):
        if A[i] > A[i_maior]:
            i_maior = i
    return i_maior
```

Outra solução:

```
def maior( A, i0 ):
    maior = A[i0]
    i_maior = i0
    for i in range( i0+1, len(A) ):
        if A[i] > maior:
            maior = A[i]
            i_maior = i
    return i_maior
```

### Questão 3

(7 pontos)

Um professor possui a relação das notas dos alunos em um arranjo **notas** de  $n$  números inteiros. As variáveis **notas** e  $n$  já estão preenchidas dentro do programa (veja abaixo). O professor agora deseja saber qual é a menor e a segunda menor nota. Considere que as notas são todas distintas.

```
import numpy as np

n = int( input('Entre com o número de alunos da turma: '))
print('Entre com a nota de cada aluno (uma em cada linha):')
notas = np.empty( n )
for i in range(0,n):
    notas[i] = int( input(''))
...
```

Exemplo de execução do programa:

```
Entre com o número de alunos da turma: 6
Entre com a nota de cada aluno (uma em cada linha):
65
89
93
75
80
77

Menor nota:      aluno 1 (nota 65)
2ª menor nota:  aluno 4 (nota 75)
```

Escreva abaixo a parte que falta no programa para ele funcionar corretamente. Se preferir, pode usar um algoritmo refinado completo em vez da sintaxe em Python.

```
menor1 = menor2 = 1000
i1 = i2 = -1
para i = 0 até n-1:
    se notas[i] < menor1:
        menor2 = menor1
        i2 = i1
        menor1 = notas[i]
        i1 = i
    senão se notas[i] < menor2:
        menor2 = notas[i]
        i2 = i
```

Escreva i1+1, menor1  
Escreva i2+1, menor2

Outra solução:

```
menor1 = notas[0]
i1 = 0
para i = 1 até n-1:
    se notas[i] < menor1:
        menor1 = notas[i]
        i1 = i

menor2 = 1000
para i = 0 até n-1:
    se notas[i] < menor2 e notas[i] ≠ menor1:
        menor2 = notas[i]
        i2 = i

Escreva i1+1, menor1
Escreva i2+1, menor2
```

#### Questão 4

(7 pontos)

Escreva um algoritmo refinado completo ou um programa em Python que leia uma matriz **A** de  $m$  linhas e  $n$  colunas contendo números reais. A partir da matriz **A** o programa deve gerar e escrever na tela um vetor **U** com o produto dos elementos de cada linha e outro vetor **V** com a soma dos elementos de cada coluna. Segue um exemplo de execução do programa:

```
Entre com o número de linhas: 3
Entre com o número de colunas: 4
Entre com os valores da matriz:
23 45 1 0
34 2 34 -1
3 2 -5 3.3

Vetor U:
0 -2312 -99

Vetor V:
60 49 30 2.3
```

Solução:

```
Leia m, n
for i in range( 0, m ):
    for j in range( 0, n ):
        Leia A[i][j]

U = criar arranjo de m elementos reais, ou: U = np.empty( m )
for i in range( 0, m ):
    U[i] = 1
    for j in range( 0, n ):
        U[i] = U[i] * A[i][j]

V = criar arranjo de n elementos reais, ou: V = np.empty( n )
for j in range( 0, n ):
    V[j] = 0
    for i in range( 0, m ):
        V[j] = V[j] + A[i][j]

for i in range( 0, m ):
    print( U[i] )

for j in range( 0, n ):
    print( V[j] )
```

Outra solução:

```
Leia m, n
for i in range( 0, m ):
    for j in range( 0, n ):
        Leia A[i][j]
```

```
U = np.empty( m )
for i in range( 0, m ): U[i] = 1
V = np.zeros( n )

for i in range( 0, m ):
    for j in range( 0, n ):
        U[i] = U[i] * A[i][j]
        V[j] = V[j] + A[i][j]

for i in range( 0, m ):
    Escreva U[i]

for j in range( 0, n ):
    Escreva V[j]
```