⊥

UNIVERSIDADE FEDERAL DE VIÇOSA — DEPARTAMENTO DE INFORMÁTICA
INF 100 – Introdução à Programação I — SEMESTRE II / 2016

Prova 2 **26/11/2016** **10:00 –12:00h**

Matrícula:_____ Nome:_____

Marque com um X sua turma **TEÓRICA**:

	Turma	Dia da Semana	Horário	Professor
	1	quinta-feira	10:00	Marcos H.
	2	sexta-feira	10:00	Lucas
	3	sexta-feira	08:00	Mauro
	4	sexta-feira	14:00	Levi
	5	quinta-feira	14:00	Marcos H.

OBSERVAÇÕES:

- As questões podem ser resolvidas a lápis.
- Entende-se por algoritmo refinado completo um algoritmo contendo a representação do código em português, de forma clara, não ambígua, de modo que cada instrução do algoritmo possa ser traduzida em uma instrução da linguagem de programação.
- Para a leitura de dados, basta usar instruções em formato algorítmico como "**`Leia n`**", "**`Leia A, B, C`**" etc. Não precisa se preocupar com as mensagens escritas antes dessas leituras. Também não é preciso se preocupar com formatação nas escritas.

A tabela abaixo apresenta a ordem de prioridade dos operadores e comandos Python mais comuns. Essa informação pode ser útil para você na resolução das questões da prova.

Prioridade	Operador(es) e comando =	Exemplo
1	`**`	`x ** 3`
2	`- (unário)`	`-x`
3	`*  /  //  %`	`x / y`
4	`+  -`	`x - y`
5	`<  <=  >  >=  ==  !=`	`x < y`
6	`not`	`not x > 0`
7	`and`	`x < y and x > 0`
8	`or`	`x < y or x == 0`
9	`=`	`x = 2`

⊥

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:

⊥

( ) `f1( 1, 10 )`

( ) `f2( 8, 2 )`

( ) `f2( 2, 8 )`

( ) `m = f3( 5, 1 )`

( ) `f4( m )`

( ) `f5( m )`

( ) `print( f5( m ))`

(A) `0 0 0 0 0`

(B) `1 1 1 1`

(C) `1 1 1 1 1`

(D) `1 2 3 4 5 6 7 8 9`

(E) `1 2 3 4 5 6 7 8 9 10`

(F) `2 3 4 5 6 7`

(G) `2 3 4 5 6 7 8`

(H) `4`

(I) `5`

(K) `8 7 6 5 4 3`

(L) `8 7 6 5 4 3 2`

(M) `Nenhuma saída`

⊥

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 menor elemento do arranjo a partir do índice i_0, e retornar o índice desse menor elemento.

Exemplo de uso da função:

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

for i in range( 0, 4 ):
    j = menor( 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 `menor()`, 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:

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

Solução:

⊥

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 maior e a segunda maior nota, para que esses alunos recebam a medalha de ouro e de prata, respectivamente. 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

Medalha de Ouro:  aluno 3 (nota 93)
Medalha de Prata: aluno 2 (nota 89)
```

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.

⊥

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 a soma dos elementos de cada linha e outro vetor **V** com o produto 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:
69 69 3.3

Vetor V:
2346 180 -170 0
```

Solução: