Universidade Federal de Viçosa
Centro de Ciências Exatas e Tecnológicas
Departamento de Informática
INF101 – Introdução à Programação II
21/09/2016

Nome: GABARITO Matr.: _____ Turma Teórica: ___

1ª Prova (Valor: 30%)

1. Dado o seguinte programa em Python:

```
def f1(n):
   x = 0
   for i in range(1, n+1):
      x = x + i;
   print(x)
   return x

def f2(L, n, x, i):
   for i in range(0, n):
      if L[i] == x:
         return i
   return -1

def main():
   L = [1, 2, 3]

   for i in range (0, 3):
      L[i] = f1(L[i])
   j = 3
   for i in range(3, 0, -1):
      j = f2(L, 3, i*2, j)
      if j >= 0:
         print(i, " > ", j)
      else:
         print(i, " < ")

main()
```

Escreva abaixo qual será a saída exata fornecida por esse programa:

```
1
3
6
3 > 2
2 <
1 <
```

2. (5%) Associe a segunda coluna com o que melhor se adequa da primeira.

(a)	Escopo global	(f)	Mergesort
(b)	Escopo local	(d)	Fila
(c)	Lista de listas de números	(i)	Inserção direta
(d)	Lista em que todas as inserções são feitas em uma extremidade e todas as remoções são feitas na outra extremidade	(c)	Matriz de números
(e)	Lista *L* com *n* elementos em ordem crescente	(b)	É o escopo que se estende pelo corpo de uma função apenas, isto é, o que estiver indentado na definição da função
(f)	Lista *L* com *n* elementos em ordem decrescente	(a)	É o escopo que é definido explicitamente por uma declaração global do Python
(g)	Método de ordenação em que a lista a ser ordenada é dividida em duas metades, cada metade é ordenada pelo mesmo método e, depois, as metades já ordenadas são intercaladas	(e)	$L[i] \leq L[j]$, se $i < j$, para $i = 0, 1, \ldots, n-1$; $j = 0, 1, \ldots, n-1$
(h)	Método de ordenação em que a lista a ser ordenada é particionada nos elementos que são menores do que um elemento tomado como pivô e nos elementos que são maiores que o pivô. Cada parte é ordenada por sua vez usando o mesmo método e, então, a primeira parte é concatenada ao pivô e depois, à segunda parte.	( )	Escopo
(i)	Método de ordenação em que o elemento em foco é inserido na sua posição correta na parte da lista que já está ordenada	(h)	Quicksort
(j)	Região de um programa onde um nome é reconhecido	(f)	$L[i] \geq L[j]$, se $i < j$, para $i = 0, 1, \ldots, n-1$; $j = 0, 1, \ldots, n-1$

3. (5%) Escreva uma função em Python para verificar a consistência do número de matrícula de um funcionário de uma empresa, sendo que o último dígito (o mais à direita) é o verificador, usando o critério a seguir, como, por exemplo, para o número: 12345-V.

1	2	3	4	5
3	5	3	5	3

Fazem-se os produtos das colunas, uma a uma, e os soma. Neste caso, obtém-se: 1*3 + 2*5 + 3*3 + 4*5 + 5*3 = 3 + 10 + 9 + 20 + 15 = 57. Depois faz-se o cálculo do resto da soma módulo 11. Assim, tem-se: 57 % 11 = 2. Se o resto for menor que 2, o dígito verificador será 0. Se for maior ou igual a 2, será 11 menos o resto. Logo, no nosso caso, o dígito será V = 11 – 2 = 9. Suponha que o número de matrícula com os seis dígitos será passado como parâmetro na forma de cadeia de caracteres contendo somente os seis dígitos. A função verificadora da consistência deve retornar verdadeiro, se o número de matrícula for válido; caso contrário, retorna falso. Você pode definir funções auxiliares para estruturar a implementação.

```
def digver(matr):
   peso = [3, 5, 3, 5, 3]
   soma = 0
   for i in range(0, len(peso)):
      soma = soma + int(matr[i]) * peso[i]
   resto = soma % 11
   if resto < 2:
      dig = str(0)
   else:
      dig = str(11 – resto)
   return dig

def matrval(matr):
   return matr[5] == digver(matr)
```

4. (5%) Escreva, em Python, uma função para determinar e retornar o índice do *menor* elemento de uma lista L de tuplas do tipo `(matr, nota)` onde `matr` é o número de matrícula de uma aluno e `nota` é sua (do aluno) nota final de INF101. O menor elemento deve ser calculado de acordo com os valores do componente `nota`. O cabeçalho da função também fica por sua conta escrevê-lo.

```
def minimo(L):
   m = 0
   for i in range(1, len(L)):
      if L[i][1] < L[m][1]:
         m = i
   return m
```

5. (5%) Represente com V ou F no espaço apropriado em cada item, caso a correspondente frase seja, respectivamente, verdadeira ou falsa.

a. (F) O Python não permite escrever funções recursivas.

b. (F) Toda função em Python tem que ter, pelo menos, um parâmetro.

c. (V) Um erro de sintaxe em Python é sempre detectado pelo interpretador da linguagem.

d. (V) Uma função em Python pode retornar um valor.

e. (F) Uma lista depois de criada não pode ter nenhum componente removido.

f. (F) Uma lista em Python tem que ter todos os componentes do mesmo tipo.

g. (F) Uma lista foi criada com 10 componentes, então o último elemento da lista tem certamente índice igual 10.

h. (F) Uma pilha é uma lista em que as inserções são sempre realizadas no início da pilha e as remoções, sempre no fim da pilha.

i. (F) Uma tupla depois de criada pode ter um componente removido.

j. (F) Uma tupla em Python tem que ter todos os componentes do mesmo tipo.

6. (5%) Seja a seguinte lista L de números inteiros:

L = [40, 58, 10, 42, 96, 18, 6, 64]

Mostre todas as configurações pelas quais a lista passa para obter ordem crescente pelo método do *mergesort*. Use os espaços apropriados abaixo. Pode haver mais espaços do que seja necessário.