**MAC2166 - Introducao a Computacao - 2017S2** **Avaliacao REC**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |

Utilize caneta azul ou preta e preencha completamente a quadrícula.
Exemplo: ■. **Não use ☒.**

**Turma:** (somente um número; consulte a pessoa responsável se não souber)

| 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 20 |
|---|---|---|---|---|---|---|---|---|---|

⟵ Marque as quadrículas ao lado para formar o seu número USP e escreva seu nome completo em letra legível na linha pontilhada abaixo. **Se seu número possui menos que 8 dígitos complete com zeros à esquerda.**

Nome:

.....

*Esta prova tem duração de 120 minutos. Não desmonte a prova.*

**Q1 [1 ponto]** Simule o código abaixo e selecione a opção correspondente a saída impressa do programa.

```
def func(n, a, b, c):
    if n <= 1:
        if a > c:
            return [a]
        else:
            return [ ]
    else:
        L1 = func(n-1, a, c, b)
        L1.append(a)
        L2 = func(n-1, c, b, a)
        return L1 + L2

def main():
    n = 3
    L = func(n, 1, 2, 3)
    print(L)
main()
```

Rascunho

| | | |
|---|---|---|
| ☐ [1, 2, 1, 3, 1] | ☐ [1, 1, 2, 3, 1] | ☐ [1, 1, 2, 3, 2] |
| ☐ [1, 2, 3, 3, 1] | ☐ [1, 2, 2, 3, 2] | ☐ [1, 1, 1, 3, 3] |
| ☐ [1, 1, 2, 3, 3] | ☐ [1, 3, 1, 3, 3] | ☐ [1, 3, 2, 3, 2] |
| ☐ [1, 1, 3, 3, 3] | ☐ [1, 3, 1, 3, 1] | ☐ [1, 2, 2, 3, 1] |
| ☐ [1, 3, 2, 3, 3] | ☐ [1, 2, 3, 3, 3] | ☐ [1, 2, 1, 3, 3] |
| ☐ [1, 1, 3, 3, 1] | ☐ [1, 3, 1, 3, 2] | ☐ [1, 1, 3, 3, 2] |
| ☐ [1, 3, 3, 3, 3] | ☐ [1, 3, 2, 3, 1] | ☐ [1, 1, 1, 3, 1] |
| ☐ [1, 1, 1, 3, 2] | ☐ [1, 3, 3, 3, 1] | ☐ [1, 2, 2, 3, 3] |
| ☐ [1, 2, 1, 3, 2] | ☐ [1, 2, 3, 3, 2] | ☐ [1, 3, 3, 3, 2] |

**Q2 [3 pontos]** A moda de um conjunto de dados é o valor que detém o maior número de observações, ou seja, o valor mais frequente (ou seja, o que "está na moda"). No caso de empate qualquer um dos valores de frequência máxima pode ser considerado como sendo uma moda.

Considere os 4 seguintes trechos de código (T1 até T4) e depois assinale as opções corretas sobre eles pintando as quadrículas. **Considerações: 1.** As opções sobre cada trecho podem conter desde nenhuma afirmação correta até todas. **2.** A cada item errado que for selecionado, desconta-se nota do exercício.

```
#T1:
 1 def moda_T1(L):
 2     freqmax = 0
 3     imax = 0
 4     for i in range(len(L)):
 5         freq = 1
 6         for j in range(i+1, len(L)):
 7             if L[j] == L[i]:
 8                 freq += 1
 9         if freq > freqmax:
10             freqmax = freq
11             imax = i
12     return L[imax]
```

```
#T2:
 1 def moda_T2(L):
 2     inic = 0
 3     fim = len(L)-1
 4     meio = (inic+fim)//2
 5     if inic == fim:
 6         return L[inic]
 7     m1 = moda_T2(L[inic:meio+1])
 8     m2 = moda_T2(L[meio+1:fim+1])
 9     freq1,freq2 = 0,0
10     for i in range(inic, fim+1):
11         if L[i] == m1:
12             freq1 += 1
13         elif L[i] == m2:
14             freq2 += 1
15     if freq1 > freq2:
16         return m1
17     else:
18         return m2
```

```
#T3:
 1 def frequencias(L):
 2     dic = {}
 3     for i in range(len(L)):
 4         if i in dic:
 5             dic[L[i]] += 1
 6         else:
 7             dic[L[i]] = 1
 8     return dic

 9 def moda_T3(L):
10     dic = frequencias(L)
11     freqmax = 0
12     M = []
13     for freq in dic.keys():
14         if freq > freqmax:
15             freqmax = freq
16     for num in dic:
17         if dic[num] == freqmax:
18             M.append(dic[num])
19     return M
```

```
#T4:
 1 def freq(L, x):
 2     if len(L) == 0:
 3         return 0
 4     elif L[0] == x:
 5         return freq(L[1:], x) + 1
 6     else:
 7         return freq(L[:len(L)-1], x)

 8 def moda_T4(L):
 9     m = 0
10     for i in range(len(L)):
11         if freq(L, L[i]) > freq(L, L[m]):
12             m = freq(L, L[i])
13     return L[m]
```

O código T1 devolve uma moda da lista L. No caso de empate qual é a moda devolvida?

T1
- ☐ Para L = [2,4,8,8,4,4,8,2] devolve a moda 4
- ☐ Para L = [2,4,8,8,4,4,8,2] devolve a moda 8

Se trocarmos a linha 6 em T1 para `for j in range(0, i):`

T1
- ☐ O código se torna incorreto para calcular uma moda
- ☐ Para L = [2,4,8,8,4,4,8,2] teremos a moda 8
- ☐ Para L = [2,4,8,8,4,4,8,2] teremos a moda 4

Caso o código T2 possuir um erro de lógica, marque as listas abaixo que são contra-exemplos, isto é, listas para as quais o código T2 não devolve uma moda.

T2
- ☐ L = [1,1,3,3,3,1,1,5,5,7]
- ☐ L = [1,1,3,3,3,1,1,5,5,5]
- ☐ L = [8,8,7,7,7]
- ☐ L = [8,2,3,8,3,3,4,4,4,8,8]

O código T3 deveria devolver todas as modas da lista L em um lista M. **Exemplo:** Para L = [4,5,4,3,3,1] deveria devolver M = [3,4]. Marque as correções necessárias em T3 para que o código funcione corretamente:

T3
- ☐ Trocar linha 13 por: `for freq in dic.values():`
- ☐ Trocar a linha 4 por: `if L[i] in dic:`
- ☐ Trocar a linha 18 por: `M.append(num)`
- ☐ Trocar a linha 4 por: `if L[i] not in dic:`
- ☐ Trocar a linha 3 por: `for i in L:`

O código T4 deveria devolver uma moda da lista L. Marque as correções:

T4
- ☐ Trocar a linha 12 por: `m = i`
- ☐ Trocar a linha 7 por: `return freq(L[1:], x)`
- ☐ Trocar a linha 11 por: `if freq(L, L[i]) > m:`

**Q3 [3 pontos]** Preencha as lacunas no código abaixo (L1 até L14), de forma a obter um programa que lê um arquivo de texto e calcula o numero de vezes que cada palavra ocorre. No resultado final as palavras devem estar listadas em ordem alfanumérica crescente. Este programa tem tres funções: uma para ler o arquivo e produzir um texto único, um para calcular as frequências e outro para ordenar uma lista de palavras.

```
def main ():
    t = le_texto()
    freq = computa_frequencias(t)
    ps = []
    for x in L1
        L2
    ordena(ps)
    for L3
        L4
def le_texto()
    L5
    res = L6
    for L7
        i = i.strip()
        L8
    arq.close()
    return res
def computa_frequencias(par)
    pals = L9
    res = L10
    for p in pals:
        L11
    return res
def ordena(lista)
    L12
        L13
            L14
main()
```

Rascunho

Para cada um dos **14** itens a seguir, correspondendo as lacunas no código acima, assinale a única resposta que torna o programa acima correto.

L1:
- [ ] `keys(freq)`
- [ ] `freq`
- [ ] `freq(keys)`
- [ ] `freq.keys()`

L2:
- [ ] `ps = ps + [x]`
- [ ] `ps = [ps] + [x]`
- [ ] `append(ps,x)`
- [ ] `ps = ps + x`

L3:
- [ ] `p in ps`
- [ ] `ps`
- [ ] `p in keys(ps)`
- [ ] `p in ps.keys()`

L4:
- [ ] `print(ps, '==>', freq,)`
- [ ] `freq[p] = 0`
- [ ] `print(p,'=>',freq[p]))`
- [ ] `freq[p] += 1`

L5:
- [ ] `arq = open('text','r')`
- [ ] `open(arq,'text.txt,'rw')`
- [ ] `arq = file('text').open`
- [ ] `arq.open('text.txt','rw')`

L6:
- [ ] `None`
- [ ] `''`
- [ ] `1`
- [ ] `0`

L7:
- [ ] `i in arq.lines()`
- [ ] `i in arq.file()`
- [ ] `i in arq:`
- [ ] `i in file:`

L8:
- [ ] `res = res.[i]`
- [ ] `res += i`
- [ ] `res += [i]`
- [ ] `res.append(i)`

L9:
- [ ] `split(par)`
- [ ] `par.words()`
- [ ] `[par]`
- [ ] `par.split()`

L10:
- [ ] `[]`
- [ ] `''`
- [ ] `None`
- [ ] `{}`

L11:
- [ ]
```
if (resp):
   res[p] +=1
else:
   res[p]=0
```
- [ ]
```
if (p in res):
   res[p] +=1
else:
   res[p]=0
```
- [ ]
```
if (p in res):
   res[p] +=1
else:
   res[p]=1
```
- [ ] `res[p] +=1`

L12:
- [ ]
```
tam = len(lista)
for i in range(tam-1):
```
- [ ]
```
tam = len(lista)
for i in range(1,tam-1):
```
- [ ]
```
tam = lista.size()
for i in range(tam-1):
```
- [ ]
```
tam = len(lista)
for i in range(tam):
```

L13:
- [ ] `for j in range(0,tam):`
- [ ] `for j in range(i+1,tam):`
- [ ] `for j in range(1,tam-1):`
- [ ] `for j in range(i,tam-1):`

L14:
- [ ]
```
if (l[i] < l[j]):
   l[i],l[j] = l[i],l[j]
```
- [ ]
```
if (l[i] > l[j]):
   l[i] = aux
   l[j] = l[i]
   aux=l[i]
```
- [ ] `l.change(i,j)`
- [ ]
```
if (l[i] > l[j]):
   l[i],l[j] = l[j],l[i]
```

**Q4 [3 pontos]** Nesta questão você deve elaborar um programa que, dado um número maior ou igual a zero, diga qual o dígito que apresenta um número maior de repetições consecutivas. Por exemplo para o número 12233340000 você deveria indicar que o número 0 aparece 4 vezes. Assinale a alternativa que contém os blocos corretos na ORDEM correta.

**DICA 1**: As variáveis do programa são APENAS: `dig`:um digito; `digmax`:dígito de frequência máxima; `cont`:um contador; `max`:-um valor máximo; `num`:um número; `ult`:último valor.

**DICA 2**: Não tente usar todas as combinações, tente codificar o programa na área de rascunho e depois escolha os trechos adequados. O RASCUNHO NÃO SERÁ CONSIDERADO NA NOTA.

**DICA 3**: Na solução sugerida nós tratamos o primeiro dígito fora do loop para simplificar o código inerno; os dígitos são extraídos da direita para a esquerda; o número zero precisa de tratamento especial.

```
#A
main()

#B
def main():
    num = int(input())

#C
    max = 0

#D
    max = 0
    ult = 0
    digmax = 0

#E
    max = 1
    ult = ''
    digmax = 0

#F
    dig = num % 10
    num = num // 10
    digmax = dig
    cont = 1
```

```
#G
    num = num /10
    dig = num % 10
    cont = 1

#H
    dig = num //10
    num  = num % 10
    cont = 0
    maxdig = 0

#I
            cont = 1

#J
            dig = 0

#K
            udig = 1

#L
    if (max == 1):

#M
    if (udig == 0):

#N
    if (digmax == 0):
```

```
#O
        if (dig == udig):
            cont += 1

#P
        if (dig > udig):
            cont = cont + 1

#Q
    else:

#R
        else:
            if (cont > max):
                max = cont
                digmax = udig

#S
            if (cont > max):
                max = cont
                digmax = udig

#T
    while (num > 0):
        udig = dig
        dig = num %10
        num = num // 10
```

```
#U
    while (not(dig ==0)):
        udig = dig
        dig = num %10
        num = num // 10

#V
    while (num > 0):
        num = num // 10
        dig = num %10
        udig = dig

#W
        print('Maior tem apena um digito (',digmax,')')

#X
         print('Maior tem ', max, ' digitos (',digmax,')')
#Y
    else:

#Z
    if (num == 0):
        cont = 1
        print('Maior tem apenas um digito (0)')
        return
```

- ☐ B,C,Z,F, U, O, R, I, L,W,Y,X,A
- ☐ B,C,Z,G, U, O, S, J, L,W,X
- ☐ B,C,H, T, O, S, I, L,W,Y,X
- ☐ B,C,Z,G, M,N, S, J, L,W,X,A
- ☐ B,C,Z,F, T, O, R, I, L,W,Y,X,A
- ☐ B,Z,H, V, O, S, J,K, L,W,Y,X,A
- ☐ B,C,Z,H, V, O, R, I, L,W,Y,X,A
- ☐ B,C,Z,G, U, O, S, J, L,W,X,A
- ☐ B,Z,H, V, O, R, I, L,W,Y,X,A
- ☐ B,C,H, T, O, R, J, KL,W,Y,X
- ☐ B,C,U,G, T, O, S, J, L,W,X,A
- ☐ B,C,Z,G, U,M,N,J, S, J, L,W,X,A

Rascunho