



MAC2166 - Introdução à Computação - 2017S1

Prova Substitutiva

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)

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

← 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 avaliação tem duração de 120 minutos. Não desmonte este caderno.

Q1 [2 pontos] Simule o código abaixo e assinale a opção correspondente à saída impressa do programa. Ignore quebras de linhas geradas pelo comando `print`.

```
def r (G, o, m, S):
    if o == m:
        return True
    S[o] = o
    for v in G[o]:
        if S[v] == '':
            if r(G, v, m, S):
                S[o] = v
                return True
    S[o] = ''
    return False

def main ():
    G = {'A': 'BD', 'B': 'ACE', 'C': 'B',
        'D': 'AE', 'E': 'DB'}
    S = {}
    for v in G:
        S[v] = ''
    if r(G, 'A', 'E', S):
        o = 'A'
        while o != 'E':
            print (o)
            o = S[o]
    else:
        print ('* * *')

main()
```

Rascunho

A B C E

A D

B D E

A B C B E

A B A D A E

A D E

A B E D

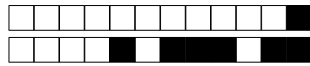
A B E

A B D E

A B C D E

* * *

A B



Q2 [3 pontos] Considere o programa abaixo.

```
def main():
    D = [1,2,3]
    R = []
    for i in D:
        x,d = f( i, 0 )
        y = g( i, d-1 )
        R.append( x-y )
    print( R )

def f(x,e,i):
    if x<e:
        return x,i
    else:
        nx,ni = f( x/2, e, i+1 )
        return x+nx,ni

def g(x,e,i):
    s = 0
    while i != 0:
        s += x
        x /= 2
        i -= 1
    return s

main()
```

Assinale, entre as opções a seguir, as afirmações verdadeiras pintando as quadrículas correspondentes.

Considerações:

1. É possível que nenhuma opção esteja correta, que algumas opções estejam corretas e algumas incorretas, ou que todas as opções estejam corretas.
2. Para cada opção incorretamente assinalada, desconta-se 0,5 ponto da nota total da questão.
3. A nota mínima da questão é zero.

- Após executar a função `main`, `R[0] == R[1]/2`
- A função `f(x,e,i)` não termina se `i < 0`
- A função `f(x,e,i)` tem como condição de parada o cálculo de `i` termos
- A função `g(x,e,i)` não termina se `i < 0`
- A função `g(x,e,i)` calcula termos até chegar na precisão especificada por `e`
- Após executar a função `main`, `R[0] == 2*R[1]`
- Após executar a função `main`, `R[0] == R[1]`
- A função `f(x,e,i)` calcula uma aproximação para a série $x + x/2 + x/4 + x/8 + \dots$
- A função `g(x,e,i)` calcula a série $x + x/2 + x/4 + x/8 + \dots + x/2^{i-1}$
- Após executar a função `main`, `len(R) == 3`



Q3 [3 pontos] Preencha as lacunas no código abaixo (L1 até L8), de forma a obter um programa que lê vários arquivos de texto e cria um dicionário que relaciona cada palavra encontrada com o nome dos arquivos em que ela ocorre. Se uma palavra ocorrer duas vezes no mesmo arquivo, ele deverá ser incluído apenas uma vez.

```
def indexaArquivo( ind, arq ):  
    f = L1  
  
    for l in f:  
        ps = L2  
        L3 :  
            if L4 :  
                L5  
            elif L6 :  
                L7  
        L8  
  
def main():  
    arquivos = [ "arq1.txt", "arq2.txt", "arq3.txt" ]  
    I = {}  
    for a in arquivos:  
        indexaArquivo( I, a )  
    print(I)  
main()
```

Rascunho

Para cada um dos 8 itens a seguir, correspondendo as lacunas no código acima, assinale a única resposta que torna o programa acima correto.

- | | | | |
|--|---|---|--|
| L1: <input type="checkbox"/> file.open(arq) | <input type="checkbox"/> open(arq, 'w') | <input type="checkbox"/> open(arq, 'r') | <input type="checkbox"/> open(arq, 'a') |
| L2: <input type="checkbox"/> l.join(' ') | <input type="checkbox"/> l.strip().split() | <input type="checkbox"/> l.split().strip() | <input type="checkbox"/> l.strip().join(' ') |
| L3: <input type="checkbox"/> for i in range(len(ps)) | <input type="checkbox"/> while i < len(ps) | <input type="checkbox"/> for p in ps | <input type="checkbox"/> for p in range(len(ps)) |
| L4: <input type="checkbox"/> ps[p] is None | <input type="checkbox"/> ps[i] in ind | <input type="checkbox"/> p in ind | <input type="checkbox"/> p not in ind |
| L5: <input type="checkbox"/> ps[p] = [arq] | <input type="checkbox"/> ind[i] = [arq] | <input type="checkbox"/> ind[p] = [arq] | <input type="checkbox"/> ps[i] = [arq] |
| L6: <input type="checkbox"/> arq not in ind[p] | <input type="checkbox"/> ind[i] | <input type="checkbox"/> not(ind[p][arq]) | <input type="checkbox"/> ind not in arq[i] |
| L7: <input type="checkbox"/> ind[p] = arq | <input type="checkbox"/> ind[i].append(arq) | <input type="checkbox"/> ind[i] = arq | <input type="checkbox"/> ind[p].append(arq) |
| L8: <input type="checkbox"/> f.close() | <input type="checkbox"/> delete(f) | <input type="checkbox"/> close(f) | <input type="checkbox"/> f.delete() |



Q4 [2 pontos] A função abaixo deve reorganizar os elementos de uma lista em uma matriz com as dimensões dadas. Por exemplo, dada a lista

```
L = [1,2,3,4,5,6,7,8,9,10]
```

a chamada

```
v2m(L,5,2)
```

retorna a matriz

```
[[1,2],[3,4],[5,6],[7,8],[9,10]] .
```

Já a chamada

```
v2m(L,2,5)
```

retorna a matriz

```
L = [[1,2,3,4,5],[6,7,8,9,10]] .
```

O código abaixo, entretanto, contém erros e não funciona como esperado. Marque as quadrículas correspondentes às correções necessárias para que a função comporte-se como esperado. O código final correto é obtido pela aplicação de todas as modificações sugeridas pelas opções selecionadas.

Observação: a cada item assinalado incorretamente, desconta-se 0,5 ponto da nota do exercício.

```
(1) def v2m(v, num_lin, num_col):
(2)     M = []
(3)     vazia = []
(4)     for j in range(num_lin):
(5)         linha = vazia
(6)         for i in range(num_col):
(7)             linha.append(v[num_lin*j + i])
(8)         M.append(linha)
(9)     return M
```

- Trocar as linhas (3) e (5) de lugar e acertar espaçamento
- Substituir a linha (4) por `for i in range(num_lin):`
- Substituir a linha (7) por `linha.append(v[num_col*j + i])`
- Substituir a linha (5) por `linha = vazia[:]`
- Substituir a linha (7) por `linha.append(v[num_lin*i + j])`
- Substituir a linha (8) por `M.append(linha[:])`
- Substituir a linha (6) por `for j in range(num_col):`