



MAC2166 - Introducao a Computacao - 2017S1

Avaliacao 1

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

Utilize caneta azul ou preta para marcar as caixas e preencha a caixa totalmente para correta interpretação. Exemplo: ■. Não use ☒.

Turma (marque somente um número, consulte seu professor se não souber):

<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10	<input type="checkbox"/> 11	<input type="checkbox"/> 12	<input type="checkbox"/> 13	<input type="checkbox"/> 14	<input type="checkbox"/> 20
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

Marque as caixas ao lado para formar o seu número USP e escreva na caixa abaixo seu NUSP e seu nome em letra legível.

NUSP:	Nome (completo):
.....	

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

Q1 [1 pontos] Simule o código abaixo e selecione a opção correspondente à saída impressa do programa. Ignore as quebras de linhas geradas pelo comando print.

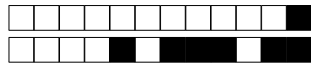
```
def main() :
    k = 25
    a = 1
    x = 2
    while (k > 0) :
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

Rascunho

<input type="checkbox"/>	14 21 42 0
<input type="checkbox"/>	14 35 35 28
<input type="checkbox"/>	14 28 14 14 28
<input type="checkbox"/>	14 42 35 7
<input type="checkbox"/>	14 21 42 0 0
<input type="checkbox"/>	14 42 42 7 35
<input type="checkbox"/>	14 42 42 7

<input type="checkbox"/>	14 42 35 7 42
<input type="checkbox"/>	14 0 14 7 28
<input type="checkbox"/>	14 21 0 42 0
<input type="checkbox"/>	14 0 7 28
<input type="checkbox"/>	14 35 28 35 7
<input type="checkbox"/>	14 28 14 14
<input type="checkbox"/>	14 21 0 42

<input type="checkbox"/>	14 28 28 14 14
<input type="checkbox"/>	14 35 35 28 7
<input type="checkbox"/>	14 28 28 14
<input type="checkbox"/>	14 35 28 35
<input type="checkbox"/>	14 0 7 28 14
<input type="checkbox"/>	14 0 14 7



Q2 [3 pontos] Considere os 6 seguintes trechos de código (T1 até T6) e depois selecione as afirmações verdadeiras sobre eles pintando as quadriculas. **Considere:** **1.** Considere que o usuário SEMPRE digitará uma sequência de naturais entre 1 e 100, terminando com o valor zero (0). Além disso, a sequência (SEQ) tem ao menos 2 elementos. **2.** As afirmações devem estar corretas quaisquer que sejam as entradas atendendo essas restrições. **3.** Nos trechos de código da imagem abaixo, o termo **SEQ** nas afirmações significa “sequência”. **4.** No que a SEQ {1,1,1} deve ser considerada **crecente** e também **decrecente**. **5.** As opções sobre cada trecho podem conter desde nenhuma afirmação correta até todas.

Atenção: A cada item errado que for selecionado, desconta-se ao menos 0.2 da nota do exercício.

<pre># Trecho T1 : x = 1 M = 0 while (x!=0) : x = int(input()) if (M<x) : M = x print(M)</pre>	<pre># Trecho T2 : x = 1 M = 0 while (x!=0) : if (M<x) : M = x x = int(input()) print(M)</pre>	<pre># Trecho T3 : x = 1 M = 0 while (x!=0) : x = int(input()) if (M>x) : M = x print(M)</pre>
<pre># Trecho T4 : x = 1000 M = 1000 while (x!=0) : if (M>x) : M = x x = int(input()) print(M)</pre>	<pre># Trecho T5 : x = -1 y = -2 M = 1 while (x!=0) : if (y>x) : M = 0 y = x x = int(input()) if (M==1) : print("sim") else : print("nao")</pre>	<pre># Trecho T6 : x = -1 y = -2 M = 1 while (x!=0) : if (y>x) : M = 0 else : M = 1 y = x x = int(input()) if (M==1) : print("sim") else : print("nao")</pre>

- | | | |
|----|---|---|
| T1 | <input type="checkbox"/> Seleciona o menor natural em SEQ | <input type="checkbox"/> Determina se a SEQ é crescente |
| | <input type="checkbox"/> Seleciona o maior natural em SEQ | <input type="checkbox"/> Falha ao tentar selecionar o maior natural em SEQ |
| T2 | <input type="checkbox"/> Seleciona o maior natural em SEQ | <input type="checkbox"/> Seleciona o menor natural em SEQ |
| | <input type="checkbox"/> Falha ao tentar selecionar o menor natural em SEQ | <input type="checkbox"/> Pode selecionar o menor para algumas SEQ |
| T3 | <input type="checkbox"/> Falha ao tentar selecionar o menor natural em SEQ | <input type="checkbox"/> Seleciona o menor natural em SEQ |
| T4 | <input type="checkbox"/> Falha ao tentar selecionar o menor natural em SEQ | <input type="checkbox"/> Seleciona o menor natural em SEQ |
| T5 | <input type="checkbox"/> Imprime <u>sim</u> se a SEQ for estritamente crescente | <input type="checkbox"/> Imprime <u>nao</u> se a SEQ for estritamente decrescente |
| | <input type="checkbox"/> Imprime <u>sim</u> se a SEQ for decrescente | <input type="checkbox"/> Imprime <u>sim</u> se a SEQ for crescente |
| T6 | <input type="checkbox"/> Imprime <u>sim</u> se a SEQ for decrescente | <input type="checkbox"/> Imprime <u>nao</u> se a SEQ for decrescente |
| | <input type="checkbox"/> Imprime <u>nao</u> se últimos 2 da SEQ são estritamente decrescentes | <input type="checkbox"/> Imprime <u>sim</u> se últimos 2 da SEQ são estritamente crescentes |



Q3 [3 pontos] Faça um programa em Python, preenchendo as lacunas (L1 até L12) no código abaixo, que remove todos dígitos repetidos de um inteiro n fornecido, mantendo apenas a primeira ocorrência mais a esquerda de cada dígito.

Exemplos:

$n = 55122345558 \rightarrow \text{saida} = 512348$, $n = 888778811222 \rightarrow \text{saida} = 8712$, $n = 10022333002233001100 \rightarrow \text{saida} = 1023$.

```
def pertence(d, n): # testa se dígito d está em n
    achou = False
    while L1:
        L2
        L3
        if L4:
            achou = True
    return achou

def main():
    n = int(input("Digite n: "))
    L5
    L6
    while L7:
        L8
        L9
        if L10:
            L11
            L12
    print(saida)

main()
```

Rascunho

Considere os **12** itens seguintes, correspondentes respectivamente a cada uma das 12 lacunas no código acima. Cada item tem 5 opções, selecione aquela que torna o programa acima correto.

L1: A `n > 0 and achou == False` B `n >= 0` C `d < n` D `d <= n` E `n != 0 and achou`

L2: A `n = n // 10` B `r = n % 10` C `n = n % 10` D `r = n // 10` E `d = d + 1`

L3: A `r = n % 10` B `n = n % 10` C `d = d + 1` D `r = n // 10` E `n = n // 10`

L4: A `r == d` B `d != n % 10` C `d == n` D `d != n` E `r != d`

L5: A `saida = n // 100` B `saida = d // 10` C `saida = n` D `saida = 1` E `saida = 0`

L6: A `pot = 10**n` B `pot = 1` C `pot = n*10` D `pot = 100` E `pot = 10`

L7: A `pot <= n` B `n >= 0` C `n > 10` D `n >= 10` E `n > 0`

L8: A `r = n // 10` B `d = n // 10` C `n = n // 2` D `n = n % 10` E `r = n % 10`

L9: A `r = n % 10` B `n = n // 10` C `r = n // 2` D `n = d // 10`
 E `n = n % 10`

L10: A `pertence(n,r)==False` B `pertence(n,r)` C `pertence(r,n // 10)` D `pertence(r,n)`
 E `pertence(r,n)==False`

L11: A `saida = saida*pot + r` B `saida += r` C `saida = r*saida + pot` D `saida += n // pot`
 E `saida = saida + r*pot`

L12: A `pot = 10**n` B `pot = pot * 10` C `pot = pot // 10` D `pot = pot * 100`
 E `pot = pot + 1`



Q4 [3 pontos] Nesta questão você deve elaborar um programa que resolva o seguinte problema; dada uma sequência de n números ($n \geq 2$) diga se o valor das diferenças entre cada dois números consecutivos é uma sequência (S) **estritamente** crescente. **Exemplos:** $n = 4, S = 8, 9, 10, 11 \rightarrow$ saída = 'diferencas nao formam seq. crescente', $n = 5, S = 8, 9, 11, 15, 19 \rightarrow$ saída = 'diferencas nao formam seq. crescente', $n = 5, S = 8, 9, 11, 15, 20 \rightarrow$ saída = 'diferencas formam seq. crescente'.

Para isso você deve usar APENAS os trechos de código indicados abaixo. Assinale a alternativa que contém os blocos corretos na ORDEM correta. ATENÇÃO: os blocos não estão identados assim o final do laço é indicado por um '*', após o número de bloco que contém o último comando do laço. **DICA 1: As variáveis do programa são APENAS:** quantidade; anterior; atual; contador; diferenca_atual; diferenca_anterior.

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

<pre>#trecho 1 diferenca_anterior = atual - anterior</pre>	<pre>#trecho 10 contador = contador + 1</pre>	<pre>#trecho 17 atual = int(input("Digite um numero:")) anterior = atual diferenca_atual = atual - anterior</pre>
<pre>#trecho 2 crescente = False</pre>	<pre>#trecho 11 if (diferenca_atual < diferenca_anterior):</pre>	<pre>#trecho 18 anterior = atual atual = int(input("Digite um numero:")) diferenca_atual = atual - anterior</pre>
<pre>#trecho 3 crescente = True contador = 2</pre>	<pre>#trecho 12 if (diferenca_atual <= diferenca_anterior):</pre>	<pre>#trecho 19 atual = int(input("Digite um numero:")) anterior = atual diferenca_atual = atual - anterior</pre>
<pre>#trecho 4 diferenca_atual = diferenca_anterior - -</pre>	<pre>#trecho 13 while (contador <= quantidade):</pre>	<pre>#trecho 20 atual = int(input("Digite um numero:")) else: crescente = True</pre>
<pre>#trecho 5 contador = contador + 2</pre>	<pre>#trecho 14 while (contador < quantidade):</pre>	<pre>#trecho 21 if ((diferenca_atual - diferenca_anterior) > 0):</pre>
<pre>#trecho 6 diferenca_anterior = diferenca_atual</pre>	<pre>#trecho 15 quantidade = int("tamanho da seq.:") anterior = int("Digite um numero:") atual = int("Digite um numero:")</pre>	<pre>#trecho 22 if (crescente): print("Diferencas formam seq. estr. crescente") else: print("Diferencas nao formam seq. estr. crescente")</pre>
<pre>#trecho 7 crescente = 0</pre>	<pre>#trecho 16 quantidade = int(input("tamanho da seq.:")) anterior = int(input("Digite um numero:")) atual = int(input("Digite um numero:"))</pre>	
<pre>#trecho 8 crescente = True contador = 0</pre>		
<pre>#trecho 9 crescente = 1 contador = 0</pre>		

- | | | |
|--|--|--|
| <input type="checkbox"/> 16, 1, 3, 14, 6, 18, 21, 2, *, 22 | <input type="checkbox"/> 16, 1, 3, 4, 14, 6, 18, 12, 2, 10, *, 22 | <input type="checkbox"/> 16, 1, 3, 14, 6, 18, 11, 2, 10, *, 22 |
| <input type="checkbox"/> 16, 1, 3, 14, 6, 18, 21, 2, 20, 10, *, 22 | <input type="checkbox"/> 15, 1, 8, 13, 6, 17, 11, 2, 20, 10, *, 22 | <input type="checkbox"/> 16, 1, 3, 14, 6, 18, 21, 2, 10, *, 22 |
| <input type="checkbox"/> 16, 1, 13, 6, 19, 11, 2, 10, *, 22 | <input type="checkbox"/> 16, 1, 8, 13, 6, 18, 11, 2, 10, *, 22 | <input type="checkbox"/> 15, 1, 8, 13, 6, 17, 11, 2, 20, 10, *, 22 |
| <input type="checkbox"/> 15, 1, 8, 13, 6, 17, 11, 2, 20, 10, *, 22 | <input type="checkbox"/> 16, 1, 13, 6, 19, 11, 2, 10, *, 22 | <input type="checkbox"/> 15, 1, 8, 13, 6, 17, 11, 2, 20, 10, *, 22 |

Rascunho