www.estudar.com.vc

Introdução à Computação em Python

Prova 2 Poli USP - 2017

1. (adaptado)

Simule o código abaixo. Qual a impressão do programa?

```
def main():
    k = 23
    a = 1
    x = 2
    while (k > 0):
        k = k – 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

2. Considere os 6 seguintes trechos de código (T1 até T6) e depois selecione as afirmações verdadeiras. **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 tabela abaixo, o termo **SEQ** nas afirmações significa “sequência”. **4.** Note que a SEQ {1,1,1} deve ser considerada **crescente** e também **decrescente**. **5.** As opções sobre cada trecho podem conter desde nenhuma afirmação correta até todas.

#Trecho T1 :
x = 1
M = 0
while (x != 0):
  x = int(input())
  if (M<x):
  M = x
print(M)
	#Trecho T2 :
	x = 1
	M = 0
	while (x != 0):
	  if (M<x):
	  M = x
	  x = int(input())
	print(M)
		#Trecho T3 :
		x = 1
		M = 0
		while (x != 0):
		  x = int(input())
		  if (M>x):
		  M = x
		print(M)

#Trecho T4 :
x = 1000
M = 1000
while (x!=0):
  if (M>x):
  M = x
  x = int(input())
print(M)
	#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”)
		#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”)

T1.

a. Seleciona o menor natural em SEQ
b. Falha ao tentar selecionar o maior natural em SEQ
c. Determina se a SEQ é crescente
d. Seleciona o maior natural em SEQ

T2.

a. Seleciona o maior natural em SEQ
b. Seleciona o menor natural em SEQ
c. Falha ao tentar selecionar o menor natural em SEQ
d. Pode selecionar o menor para algumas SEQ

T3.

a. Falha ao tentar selecionar o menor natural em SEQ
b. Seleciona o menor natural em SEQ

T4.

a. Falha ao tentar selecionar o menor natural em SEQ
b. Seleciona o menor natural em SEQ

T5.

a. Imprime sim se a SEQ for crescente
b. Imprime sim se a SEQ for estritamente crescente
c. Imprime sim se a SEQ for decrescente
d. Imprime não se a SEQ for estritamente decrescente

T6.

a. Imprime sim se os últimos 2 da SEQ são estritamente crescentes
b. Imprime não se a SEQ for decrescente
c. Imprime sim se a SEQ for decrescente
d. Imprime não se os últimos 2 da SEQ não estritamente decrescentes

3. 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 → saida = 512348, n = 888778811222 → saida = 8712

```
def pertence(d, 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(saída)
main()
```

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.

	A	B	C	D	E
L1	d <= n	n >= 0	n != 0 and achou	d < n	n > 0 and achou == False
L2	r = n%10	n = n//10	n = n%10	r = n//10	d = d + 1

L3	n = n%10	n = n//10	d = d + 1	r = n%10	r = n//10
L4	d == n	r != d	d != n	d != n%10	r == d
L5	saída = 0	saída = n//100	saída = d//10	saída = n	saída = 1
L6	pot = n*10	pot = 10**n	pot = 100	pot = 1	pot = 10
L7	n >= 10	n > 10	n > 0	n >= 0	pot <= n
L8	d = n//10	r = n%10	n = n%10	n = n//2	r = n//10
L9	r = n//2	r = n%10	n = d//10	n = n%10	n = n//1-
L10	pertence(n, r)	pertence(r,n//10)	pertence(r, n)	pertence(n, r) == False	pertence(r, n) == False
L11	saida = saida*pot + r	saida += n//pot	saida = saida + r*pot	saida += r	saida = saida*pot + r
L12	pot = pot*10	pot = pot*100	pot = 10**n	pot = pot//10	pot = pot+1

4. Nesta questão você deve elaborar um programa que resolva o seguinte problema: dada uma sequência de *n* números (*n* >= 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 → saida = 'diferencas nao 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; diferença_atual; diferença_anterior.

DICA 2: Não tente usar todas as combinações, tente codificar o programa e depois escolho os trechos adequados.

#trecho 1
diferença_anterior = atual - anterior
	#trecho 10
	contador = contador + 1
		#trecho 17
		atual = int(input("Digite um número:"))
		anterior = atual
		diferença_atual = atual - anterior

#trecho 2
crescente = False
	#trecho 11
	if(diferença_atual<diferença_anterior):
		#trecho 18
		anterior = atual
		atual = int(input("Digite um número:"))
		diferença_atual = atual - anterior

#trecho 3
crescente = True
contador = 2
	#trecho 12
	if(diferença_atual<=diferença_anterior):	

#trecho 4
diferenca_atual = diferença_anterior
	#trecho 13
	while (contador <= quantidade):
		#trecho 19
		atual = int(input("Digite um número:"))
		anterior = atual
		diferença_atual = atual - anterior

#trecho 5
contador = contador + 2
	#trecho 14
	while (contador < quantidade):	

#trecho 6
diferença_anterior = diferença_atual
	#trecho 15
	quantidade = int("tamanho da seq.:")
	anterior = int("Digite um número:")
	atual = int("Digite um número:")
		#trecho 20
		else:
		crescente = True

#trecho 7
crescente = 0
		#trecho 21
		if ((diferença_atual – diferença_anterior)>0):

#trecho 8
crescente = True
contador = 0
	#trecho 16
	quantidade = int(input("tamanho da seq.:"))
	anterior = int(input("Digite um número:"))
	atual = int(input("Digite um número:"))
		#trecho 22
		if (crescente):
		print("Diferenças formam seq. estr. \ crescente")
		else:
		print("Diferenças não formam seq. \ estr. crescente")

#trecho 9
crescente = 1
contador = 0		

Escolha uma alternativa

A. 15, 1, 8, 13, 6, 17, 11, 2, 20, 10, *, 22
B. 16, 1, 8, 13, 6, 18, 11, 2, 10, *, 22
C. 16, 1, 3, 14, 6, 18, 11, 2, 10, *, 22
D. 16, 1, 3, 14, 6, 18, 21, 2, 10, *, 22
E. 16, 1, 3, 14, 6, 18, 21, 2, *, 22
F. 16, 1, 3, 4, 14, 6, 18, 12, 2, 10, *, 22
G. 16, 1, 13, 6, 19, 11, 2, 10, *, 22
H. 16, 1, 3, 14, 6, 18, 21, 2, 20, 10, *, 22
I. 15, 1, 8, 13, 6, 17, 11, 2, 20,10, *, 22
J. 16, 1, 13, 6, 19, 11, 2, 10, *, 22
K. 15, 1, 8, 13, 6, 17, 11, 2, 20, 10, *, 22

Gabarito

1. 14
35
28
35
7

2.

T1. d
T2. a e d
T3. a
T4. b
T5. a, b e d
T6. a e d

3.

L1. e
L2. a
L3. b
L4. e
L5. a
L6. d
L7. c
L8. b
L9. e
L10. e
L11. c
L12. a

4. F