



Introdução à Ciência da Computação - 113913

Gabarito da Lista de Exercícios 4

Iteração, Estruturas de Repetição

Observações:

- As listas de exercícios serão corrigidas por um **corretor automático**, portanto é necessário que as entradas e saídas do seu programa estejam conforme o padrão especificado em cada questão (exemplo de entrada e saída). Por exemplo, a não ser que seja pedido na questão, não use mensagens escritas durante o desenvolvimento do seu código como “Informe a primeira entrada”. Estas mensagens não são tratadas pelo corretor, portanto a correção irá resultar em resposta errada, mesmo que seu código esteja correto.
- As Instâncias de Entrada serão as usadas pelo corretor e suas saídas devem estar **iguais** às apresentadas em Instâncias de Saída.

Questão 1.

```
num_advinha = int(input())
num = int(input())
while(num != num_advinha):
    if(num < num_advinha):
        print("O número correto é maior.")
    elif(num > num_advinha):#Podemos usar um else aqui também
        print("O número correto é menor.")
    num = int(input())#Lemos o próximo número

print("Parabéns! Você acertou.")
""" Aqui já vamos ter saído do loop, ou seja, num == num_advinha.
Logo, o jogador acertou o número. """
```

Instâncias de Entrada	Instâncias de Saída
7 6 8 7	O número correto é maior. O número correto é menor. Parabéns! Você acertou.
-10 -1 -5 -10	O número correto é menor. O número correto é menor. Parabéns! Você acertou.
5 6 5	O número correto é menor. Parabéns! Você acertou.
0 10 -10 0	O número correto é menor. O número correto é maior. Parabéns! Você acertou.
20 10 15 19 20	O número correto é maior. O número correto é maior. O número correto é maior. Parabéns! Você acertou.
4 3 4	O número correto é maior. Parabéns! Você acertou.
-5 -1 -2 -6 -5	O número correto é menor. O número correto é menor. O número correto é maior. Parabéns! Você acertou.
49 490 0 49	O número correto é menor. O número correto é maior. Parabéns! Você acertou.
4900 -2000 2000 4000	O número correto é maior. O número correto é maior. O número correto é maior. Parabéns! Você acertou.

4900	
-10	Parabéns! Você acertou.
-10	

Questão 2.

```
x = int(input())
maior = x
quantidade = 0
soma = 0
"""Enquanto x não for 0, vamos somando x com o valor anterior de soma,
incrementando a quantidade e verificando se x é maior do que o maior x que
já temos. Depois lemos o próximo inteiro da sequência. """
while(x != 0):
    soma += x
    quantidade += 1
    if(x > maior):
        maior = x
    x = int(input())
if(quantidade > 0):
    print(quantidade)
    print(maior)
    print("Média: %.2f"%(soma/quantidade))
else: #Caso o primeiro número lido já seja 0
    print("0\n0\n0.00")
```

Instâncias de Entrada	Instâncias de Saída
0	0 0 0.00
-5 -6 -7 -8 -9 -10 0	6 -5 -7.50
4 -4 4 -4 0	4 4 0.00
10 9 8 7 6 5 0	6 10 7.50
1 0	1 1 1.00

-12 6 6 24 -16 -8 0	6 24 0.00
3 3 0	2 3 3.00
49 490 0	2 490 269.50
2 -2 5 0	3 5 1.67
4 -8 4 0	3 4 0.00

Questão 3.

```
def fatorial(n):
    fatorial = 1
    while(n > 0):#Quando n chegar ao valor 0 o loop será terminado
        fatorial = fatorial * n
        n -= 1
    return fatorial

n = int(input())
m = int(input())
print(fatorial(n), end=" ") #Trocamos o argumento padrão do end
print(fatorial(m))
#Também podemos fazer tudo com um print: print(fatorial(n), fatorial(m))
"""Alternativamente usando o laço de repetição for:
def fatorial(n):
    fatorial = 1
    for i in range(1,n+1): #O valor i começará em 1 e irá até n
        fatorial *= i
    return fatorial
O segundo argumento do for é sempre uma lista. A função range gera uma lista
de números que iniciam no primeiro argumento e terminam no número anterior ao
segundo argumento.
Exemplo: range(0,3) gera a lista [0,1,2] """
```

Instâncias de Entrada	Instâncias de Saída
2 3	2 6
0	1 1

0	
1 0	1 1
4 4	24 24
3 10	6 3628800
5 6	120 720
7 3	5040 6
5 5	120 120
4 0	24 1
9 7	362880 5040

Questão 4.

```
x = int(input())
y = int(input())

i = 1
while(i <= y):
    if(i % x == 0):
        print(i)
    else: #Aqui não chegamos ao final da linha
        if(i == y): #Nesse caso i será o último valor a ser impresso
            print(i) #Mas a última linha pode ter menos de x números
        else:
            print(i, end= " ") #Alteramos o argumento padrão do end
        i += 1
""" Quando i for múltiplo de x, então já teremos x valores nessa linha e
devemos começar a escrever na próxima. Caso contrário apenas escrevemos o valor
de i com um espaço em branco. No final devemos incrementar i, para o while. """

""" Alternativamente usando for:
for i in range(1, y+1):
    if(i % x == 0):
        print(i)
    else:
        if(i == y):
            print(i)
        else:
            print(i, end = " ")
"""
```

Instâncias de Entrada	Instâncias de Saída
1	1
7	2
	3

	4 5 6 7
3 10	1 2 3 4 5 6 7 8 9 10
5 4	1 2 3 4
5 10	1 2 3 4 5 6 7 8 9 10
1 1	1
3 4	1 2 3 4
3 27	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
10 15	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
15 20	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
10 80	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

Questão 5.

```
t = int(input())
while(t > 0):
    a, n = input().split()
    a, n = [int(a), int(n)]
    soma = 0
    x = a
    while(x < a + n - 1):
        soma += x
        print(x, end= " ")
        x += 1
    soma += x
    print(x)
    print(soma)
    t -= 1 #Decrementamos t

"""Temos que somar os próximos N números a partir de A. Logo, somamos o valor
atual de soma com o anterior + o próximo A (que nesse caso é x) e
incrementamos o valor de x (para termos o valor do próximo A). Para imprimir
a soma usamos o caractere \n, pois ela deve ser impressa na próxima linha. """

""" Usando for:
for y in range(0, t):
    a, n = input().split()
    a, n = [int(a), int(n)]
    soma = 0
    for x in range(a, a+n):
        soma += x
        print(x, end= " ")
    print("\nSoma= %d"%soma) """
```

Instâncias de Entrada	Instâncias de Saída
0	
1 4 10	4 5 6 7 8 9 10 11 12 13 85
3 5 4 3 10 2 2	5 6 7 8 26 3 4 5 6 7 8 9 10 11 12 75 2 3 5
2 0 1 2 3	0 0 2 3 4 9
2 1 1 10 10	1 1 10 11 12 13 14 15 16 17 18 19 145
4 0 1 -1 4 -5 10	0 0 -1 0 1 2 2

-3 1	-5 -4 -3 -2 -1 0 1 2 3 4 -5 -3 -3
2 -10 5 -4 2	-10 -9 -8 -7 -6 -40 -4 -3 -7
6 -1 1 -2 2 5 2 4 3 9 1 -10 2	-1 -1 -2 -1 -3 5 6 11 4 5 6 15 9 9 -10 -9 -19
2 10 10 15 3	10 11 12 13 14 15 16 17 18 19 145 15 16 17 48
5 -10 10 -9 10 -5 8 -4 10 -3 15	-10 -9 -8 -7 -6 -5 -4 -3 -2 -1 -55 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 -45 -5 -4 -3 -2 -1 0 1 2 -12 -4 -3 -2 -1 0 1 2 3 4 5 5 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 60

Questão 6.

```

print(soma)
"""Pegamos a soma da 1ª dupla de X e Y como maior e menor soma, depois vamos
comparando com as somas das outras duplas. """
while(n > 1): #Lemos os próximos n-1 casos de teste
    x, y = input().split()
    x, y = [int(x), int(y)]
    soma = 0
    while(y > 0):#While *
        if(x % 2 != 0):
            soma += x
            y -= 1
        x += 1
    if(soma > maior):
        maior = soma
    if(soma < menor):
        menor = soma
    print(soma)
    n -= 1
print(maior)
print(menor)

""" Para o while *, se o atual x for ímpar, então somamos ao valor de soma e
atribuímos a variável soma. Também decrementamos o valor de y, pois dos Y
ímpares consecutivos já atribuímos 1 a soma. Depois incrementamos x e continua
mos verificando. Após o while * imprimimos a soma."""
""" Usando for (complete colocando para imprimir a maior e a menor soma):
for i in range(0, n):
    x, y = input().split()
    x, y = [int(x), int(y)]
    soma = 0
    if(x % 2 == 0):
        for x in range(x, x+2*y):
            if(x%2 != 0):
                soma += x
        print(soma)
    else:#Se x for ímpar então precisamos somar y-1 ímpares a partir de x
        for x in range(x, x+2*y - 1):
            if(x%2 != 0):
                soma += x
        print(soma) """

```

Instâncias de Entrada	Instâncias de Saída
1	24
-1 6	24
	24
2	32
4 4	21
5 3	32
	21
3	8
2 2	-25
-10 5	12
-4 6	12
	-25

4 -5 2 1 1 1 10 1 15	-8 1 100 225 225 -8
2 5 6 3 4	60 24 60 24
3 5 4 4 5 -5 4	32 45 -8 45 -8
4 -7 5 -8 5 5 4 4 5	-15 -15 32 45 45 -15
3 49 5 30 2 15 9	265 64 207 265 64
5 -10 5 -6 8 -14 10 -16 20 10 9	-25 16 -40 80 171 171 -40
6 -11 6 -7 9 -24 10 -32 40 -7 5 -8 5	-36 9 -140 320 -15 -15 320 -140

Questão 7.

```
n = int(input())

m = 1
while(m <= 1000):
    i = 2 #Começamos com 2 para testar se n*m + 1 não é primo
    while(i < n*m + 1):
        if((n*m + 1) % i == 0):#Caso n*m+1 não seja primo
            print(m)
            m = 1001 #Interrompemos o primeiro loop
            break #Interrompemos o segundo loop
        i += 1 #Vamos incrementando 1 até n*m, para descobrir se n*m+1 é primo
    m += 1
```

Amostras de Entrada	Amostras de Saída
2	4
1	3
6	4
1000	1
16	2
36	4
70	2
228	3
600	4
576	3