

Quarta Lista de Exercícios - Computational Thinking

1. Dada uma sequência de números inteiros onde o último elemento é o 0, escreva um algoritmo que calcula a soma dos números pares da sequência.

```

1
2 somaPar = 0
3 num = int(input("Informe num da sequencia"))
4
5 while num != 0:
6     if num % 2 == 0:
7         somaPar = somaPar + num
8     num = int(input("Informe num da sequencia"))
9
10 print("A soma dos pares e {}".format(somaPar))

```

2. Dados o número n de alunos de uma turma de Algoritmos e suas notas da primeira prova, determinar a média das notas dessa turma. Considere que o usuário digite as informações corretamente.
3. Altere o algoritmo anterior para, além da média, contar os alunos que tiraram entre 0 e 5,0 ($0 \leq \text{nota} < 5,0$) e acima de 5,0 ($\text{nota} \geq 5,0$).

```

1 #Correcao dos exercicios 2 e 3
2 soma = 0
3
4 n = int(input("Informe a qtd de alunos"))
5 conta = 1
6
7 contaMenos5 = 0
8 contaMais5 = 0
9
10 while conta <= n:
11     nota = float(input("Informe a nota do aluno"))
12     soma = soma + nota
13     if nota < 5:
14         contaMenos5 = contaMenos5 + 1
15     else:
16         contaMais5 = contaMais5 + 1
17
18 media = soma / n
19 print("A media da turma e {}".format(media))
20 print("{} foi a qtd de alunos que tiraram menos que 5.".format(contaMenos5))
21 print("{} foi a qtd de alunos que tiraram mais ou igual a 5.".format(contaMais5))

```

4. Dados n um inteiro positivo e uma sequência de n números reais, escreva um algoritmo que conta e imprime a quantidade de números positivos e a quantidade de números negativos.

```

1 n = int(input("Informe a qtd de numeros da sequencia: "))
2 conta = 1
3
4 contaPos = 0
5 contaNeg = 0
6
7 while conta <= n:
8     num = float(input("Informe um numero da sequencia: "))
9     if num < 0:
10         contaNeg = contaNeg + 1
11     elif num > 0:
12         contaPos = contaPos + 1
13
14 print("{} foi a qtd de numeros negativos.".format(contaNeg))
15 print("{} foi a qtd de numeros positivos.".format(contaPos))

```

5. Escreva um algoritmo que, dados um número inteiro positivo n , imprime na tela a contagem de todos os divisores positivos de n .

```

1 n = int(input("Informe numero que deseja contar os divisores: "))
2 conta = 0
3
4 div = 1
5
6 while div <= n:
7     if n % div == 0:
8         conta = conta + 1
9
10 print("{} e a qtd de divisores de {}".format(conta, n))

```

6. Em uma prova de concurso com 70 questões haviam 20 pessoas concorrendo. Sabendo que cada questão vale 1 ponto, escreva um algoritmo que lê a pontuação da prova obtida de cada um dos candidatos e calcula:

- a) a maior e a menor nota
- b) o percentual de candidatos que acertaram até 20 questões, o percentual que acertaram de 21 a 50 e o percentual que acertou acima de 50 questões

```

1 nota = int(input("Cand: "))
2 maior = nota
3 menor = nota
4
5 conta = 2
6
7 contaAte20 = 0
8 conta20a50 = 0
9 contaAcima50 = 0
10
11 while conta <= 70:
12     nota = int(input("Cand: "))
13     if nota > maior:
14         maior = nota
15     if nota < menor:
16         menor = nota
17
18     if nota <= 20:
19         contaAte20 = contaAte20 + 1
20     elif nota <= 50:
21         conta20a50 = conta20a50 + 1
22     else:
23         contaAcima50 = contaAcima50 + 1
24
25     conta = conta + 1
26
27 print("A maior nota eh", maior)
28 print("A menor nota eh", menor)
29
30 print("Ate 20 questoes {:.2f}".format(contaAte20 * 100 / 70))
31 print("Entre 20 e 50 questoes {:.2f}".format(conta20a50 * 100 / 70))
32 print("Acima de 50 questoes {:.2f}".format(contaAcima50 * 100 / 70))

```

7. A conversão de graus Fahrenheit para centígrados é obtida pela fórmula $C = \frac{5}{9}(F - 32)$. Escreva um algoritmo que calcule e escreva uma tabela de graus centígrados em função de graus Fahrenheit que variem de 50 a 150 Fahrenheit de 1 em 1.

```

1
2 fahr = 50
3 while fahr <= 150:
4     celsius = (fahr - 32) * 5 / 9
5     print("F: {} Celsius: {}".format(fahr, celsius))
6     fahr = fahr + 1

```

8. Um número inteiro positivo n é denominado primo se existirem apenas dois divisores inteiros positivos dele: o 1 e o próprio n . Escreva um algoritmo que recebe um inteiro n e verifica se n é primo ou não.

```

1 n = int(input("Digite n: "))
2
3 div = 1
4 contador = 0
5
6 while div <= n:
7     if n % div == 0:
8         contador = contador + 1
9     div = div + 1
10
11 if contador == 2:
12     print(n, " eh primo")
13 else:
14     print(n, " nao eh primo")

```

9. Dados um montante em dinheiro inicial d , uma taxa de juros mensal j e um período de tempo em meses t , escreva um algoritmo que calcula o valor final em dinheiro se d ficar aplicado a taxa de juros j durante t meses.

```

1 dinheiro = float(input("Montante: "))
2 juros = float(input("Juros: "))
3 tempo = int(input("Tempo aplicacao: "))
4
5 while t > 0:
6     dinheiro = dinheiro * (1 + juros / 100)
7     t = t - 1
8
9 print("O valor final aplicado e de {:.2f}{}".format(dinheiro))

```

10. Escreva um algoritmo que recebe um inteiro positivo n e calcula $n! = 1 \cdot 2 \cdot 3 \dots \cdot (n-1) \cdot n$. Por exemplo, se $n = 6$, então $6! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$.

```

1 n = int(input("Informe n: "))
2
3 fatorial = 1
4
5 while n > 0:
6     fatorial = fatorial * n
7     n = n - 1
8
9 print("O fatorial vale {}".format(fatorial))

```

11. No problema de verificar se um número é primo ou não resolvemos contando o número de divisores. Também podemos pensar em resolver este problema encontrando um divisor diferente de 1. Se tal divisor for o próprio n , significa que n é primo, caso contrário, dizemos que ele não é primo. Pensando nessa ideia, escreva um algoritmo que verifica se n é primo ou não. Ao invés do comando `for` use o comando `while`

```

1 n = int(input("Informe n: "))
2
3 div = 2
4 if n == 1:
5     print("Nao e primo")
6 else:
7     while n % div != 0:
8         div = div + 1
9
10    if n == div:
11        print("E primo")
12    else:
13        print("Nao e primo")

```

12. Se F_n é o n -ésimo número da sequência de Fibonacci, podemos calculá-la através da seguinte fórmula de recorrência:

$$F_n = \begin{cases} 1 & \text{se } n = 1 \text{ ou } n = 2; \\ F_{n-1} + F_{n-2} & \text{se } n > 2 \end{cases}$$

Vamos mostrar os 10 primeiros números da sequência de Fibonacci:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55

Escreva um algoritmo que dado n , calcula o n -ésimo número da sequência de Fibonacci.

```

1  n = int(input("Qual numero de Fibonacci voce quer: "))
2
3  anterior = 1
4  atual = 1
5
6  if n <= 2:
7      print(1)
8  else:
9      i = 3
10     while i <= n:
11         proximo = atual + anterior
12         anterior = atual
13         atual = proximo
14         i = i + 1
15
16     print(atual)
```

13. Dizemos que um número natural n é palíndromo se o 1º algarismo de n é igual ao seu último algarismo, o 2º algarismo de n é igual ao penúltimo algarismo, e assim sucessivamente. Exemplos: 567765 e 32423 são palíndromos. 567675 não é palíndromo.

```

1  num = int(input("Informe o numero: "))
2
3  numOriginal = num
4  resp = 0
5
6  while num != 0:
7      d = num % 10
8      num = num // 10
9      resp = resp * 10 + d
10
11 if numOriginal == resp:
12     print("Eh palindrome")
13 else:
14     print("Nao eh palindrome")
```

14. Uma das maneiras de evitar erros na digitação de números como conta corrente, CPF, boleto bancário é a utilização de um ou mais dígitos de controle. Um dos métodos de cálculo é a utilização do método módulo 10. Segue a descrição do algoritmo: Dado um número inteiro n devemos pegar cada dígito desse número começando pela casa das unidades e multiplicar, alternadamente, por 2 e por 1. Caso o resultado da multiplicação seja um número maior ou igual a 10 devemos simplificar esse valor somando os dois dígitos. Após feitas as multiplicações e as simplificações devemos somar todos os valores e calcular o resto da divisão dessa soma por 10. Se o resto for 0 o dígito de controle é zero, caso contrário o dígito de controle será 10 menos o resto.

A Figura 1 pode servir de exemplo para o algoritmo módulo 10.

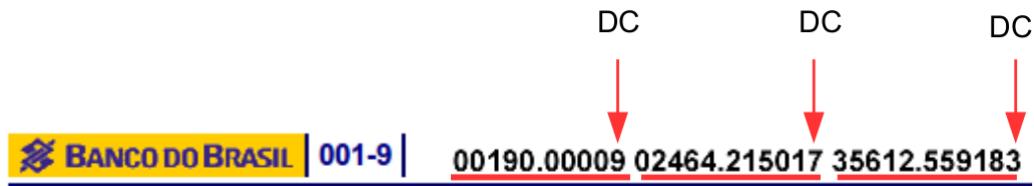


Figura 1: Trecho de boleto bancário do Banco do Brasil

Por exemplo vamos pegar o número do meio do boleto, o número corresponde a $n = 246421501$ e o dígito de controle será 7. Vamos efetuar os seguintes cálculos:

$$\begin{aligned}
 1 * 2 &= 2 \\
 0 * 1 &= 0 \\
 5 * 2 &= 10 \\
 1 * 1 &= 1 \\
 2 * 2 &= 4 \\
 4 * 1 &= 4 \\
 6 * 2 &= 12 \\
 4 * 1 &= 4 \\
 2 * 2 &= 4
 \end{aligned}$$

Daí somamos $2 + 0 + 1(1 + 0) + 1 + 4 + 4 + 3(1 + 2) + 4 + 4 = 23$. O resto da divisão de 23 por 10 é 3 e como ele é diferente de zero o dígito de controle de 246421501 será 7 ($10 - 3$).

Escreva um algoritmo que lê um número inteiro positivo e calcula o seu dígito de controle usando o método do módulo 10.

```

1  numero = 2464215017
2
3  dcDigitado = numero % 10
4  numero = numero // 10
5  soma = 0
6  fator = 2
7  while numero != 0:
8      dig = numero % 10
9      dig = dig * fator
10     valor = (dig // 10) + (dig % 10)
11     soma = soma + valor
12     numero = numero // 10
13     if fator == 2:
14         fator = 1
15     else:
16         fator = 2
17
18 resto = soma % 10
19 dcCalculado = 0
20 if resto > 0:
21     dcCalculado = 10 - resto
22
23 if dcDigitado == dcCalculado:
24     print("Parabéns, tudo certo!")
25 else:
26     print("Algo deu errado!")

```

Boa sorte!