

Fundamentos de Programação

Prof. Italo Mendes da Silva Ribeiro

Lista 4

Não apague as funções desenvolvidas do seu código-fonte, pois poderão ser utilizadas em questões posteriores.

1 – Desenvolva uma função que retorne o fatorial de um número.

```
1 def fatorial(n):
2     fat = 1
3
4     for x in range(1, n + 1):
5         fat *= x
6
7     return fat
8
9
10 numero = int(input("número: "))
11 print("Fatorial de", numero, "é", fatorial(numero))
```

2 – Crie uma função que receba como parâmetro um valor inteiro e positivo n, e retorne o valor de S dado por:

$$S = 5 + \frac{1^2}{1!} + \frac{2^2}{2!} + \frac{3^2}{3!} + \dots + \frac{n^2}{n!}$$

```
1 def fatorial(n):
2     fat = 1
3
4     for x in range(1, n+1):
5         fat *= x
6
7     return fat
8
9
10 def valorS(n):
11     s = 0
12
13     for x in range(1, n+1):
14         s += pow(x, 2) / fatorial(x)
15
16     return s + 5
17
18
19 numero = int(input("número: "))
20 print("S =", valorS(numero)) # para numero = 7, S é 10.434722
```

- 3 – Crie uma função que receba como parâmetro um valor inteiro e positivo n , e retorne o valor de A dado por: ($n = 7$, $A = 5913$)

$$A(n) = 1! + 2! + 3! + 4! + \dots + n!$$

- 4 – Implemente uma função que retorne o valor de B definido por: ($B = 47.062764$)

$$B = 7 + \frac{\pi}{\sqrt{1}} + \frac{\pi}{\sqrt{2}} + \frac{\pi}{\sqrt{3}} + \frac{\pi}{\sqrt{4}} + \dots + \frac{\pi}{\sqrt{49}} + \frac{\pi}{\sqrt{50}}$$

- 5 – Implemente uma função que retorne o valor de C dado por: ($C = -23424.5682567$)

$$C = \tan\left(\frac{2\pi}{3}\right) - 5^2 + \tan\left(\frac{3\pi}{4}\right) - 7^2 + \tan\left(\frac{4\pi}{5}\right) - 9^2 + \dots + \tan\left(\frac{25\pi}{26}\right) - 51^2$$

- 6 – Faça uma função que retorne o valor de D determinado por: ($D = 6.188925050499848e+80$)

$$D = \frac{1!}{\sqrt{10}} + \frac{2!}{\sqrt{13}} + \frac{3!}{\sqrt{16}} + \dots + \frac{59!}{\sqrt{184}} + \frac{60!}{\sqrt{187}}$$

- 7 – Crie uma função que receba um valor inteiro e positivo n e mostre o valor de I dado por:
(Com $n = 9$, $I = 226.877983$)

$$I = \frac{A(n) + B^4}{\sqrt{3} - C} + \frac{5}{D}$$

- 8 – A permutação expressa a ideia de que objetos distintos podem ser arranjados em inúmeras ordens diferentes. Se temos um total de 10 elementos, como $\{1, 2, 3, \dots, 9, 10\}$, e quiséssemos saber de quantas maneiras podemos arranjar 3 elementos desse conjunto, basta calcular a permutação de $P(10, 3)$ que resultará em 720 combinações. A fórmula da permutação é:

$$P(n, r) = \frac{n!}{(n - r)!}$$

Onde $n > 0$ e $n > d \geq 0$. Crie uma função que receba dois valores e retorne o número de permutações. A função deve verificar se os valores de n e r são adequados, caso contrário deve retornar -1.

- 9 – Uma combinação é um subconjunto s de um conjunto U com n elementos sem repetição, representado por $C(n, s)$ ou $\binom{n}{s}$. Dada as três primeiras letras do alfabeto $\{a, b, c\}$, combinadas duas a duas resultará nos subconjuntos $\{ab, ac, bc\}$, ou seja, $C(3, 2) = 3$. A fórmula da combinação é:

$$C(n, s) = \binom{n}{s} = \frac{n!}{s! \times (n - s)!}$$

Onde $n > 0$ e $n > s \geq 0$. Implemente uma função que receba dois valores e retorne o número de combinações. A função deve verificar se os valores de n e s são adequados, caso contrário deve retornar -1.

- 10 – Crie uma função que retorne o valor de F dada pela equação abaixo, onde $P()$ é a permutação. ($F = 1332.00$)

$$F = \binom{4}{2} + P(7, 2) + \binom{5}{2} + P(8, 2) + \dots + \binom{12}{2} + P(15, 2)$$

- 11 – Faça uma função que valide uma hora no formato de 24h, composta por horas e minutos, no padrão HH:MM. Onde H representa um dígito da hora e M representa um dígito de minuto. A função retornará 1 se a hora é válida e -1 caso contrário.

```
1 def validaHora(h, m):
2     if h >= 0 and h < 24 and m >= 0 and m < 60:
3         return 1
4     else:
5         return -1
6
7
8 hora = int(input("hora: "))
9 minuto = int(input("minuto: "))
10
11 if validaHora(hora, minuto) == 1:
12     print("A hora informada foi", hora, ":", minuto)
13 else:
14     print("Você informou uma hora inválida")
```

- 12 – Desenvolva uma função que valide uma hora no formato de 24h, composta por horas e minutos, no padrão HH:MM:SS:mmmm. Onde H representa um dígito da hora, M representa um dígito de minuto, S representa um dígito do segundo e m representa um dígito de milissegundo. A função retornará 1 se a hora é válida e -1 caso contrário.
- 13 – Crie uma função que valide uma data no padrão MM/AAAA. Onde M representa um dígito do mês e A um dígito do ano. Considere o ano atual, como valor máximo para o ano. Avalie o mês de fevereiro com 28 dias e os demais meses com 30 dias. A função retornará 1 caso a data seja válida e -1 se a data seja inválida.
- 14 – Crie uma função que valide uma data no padrão DD/MM/AAAA. Com D, M e A representando respectivamente um dígito do dia, mês e ano. Considere o ano atual, como valor máximo para o ano. Avalie o mês de fevereiro com 28 dias e os demais meses com 30 ou 31 dias, de acordo com o mês. A função retornará 1, caso a data seja válida, -1 se o dia é inválido, -2 se o mês é inválido e -3 se o ano é inválido.

-
- 15 – Implemente uma função que calcule o fatorial de um número recursivamente.

```
1 def fatorial(n):
2     if n <= 1:
3         return 1
4     else:
5         return n * fatorial(n-1)
6
7
8 numero = int(input("número: "))
9 print("Fatorial de", numero, "é", fatorial(numero))
```

- 16 – A sequência de Fibonacci é 0, 1, 1, 2, 3, 5, 8, 13, ... de forma que os dois primeiros termos são 0 e 1, e cada termo após é definido recursivamente pela soma de dois termos anteriores, assim:

$$Fib(n) = Fib(n - 1) + Fib(n - 2)$$

Escreva uma função recursiva que retorne enésimo termo da sequência de Fibonacci.

17 – A soma de uma série consecutiva de números de 1 a n pode ser definida recursivamente como:

$$\text{Soma}(1) = 1;$$

$$\text{Soma}(n) = n + \text{soma}(n - 1);$$

Escreva uma função recursiva que aceite n como argumento e calcula a soma dos número de 1 a n.

18 – O valor de x^n pode ser definido recursivamente como:

$$x^0 = 1$$

$$x^n = x * x^{n-1}$$

Escreva uma função recursiva para calcular o valor de x^n , com n podendo ser positivo ou negativo.