

Introdução

As séries numéricas são usadas em diversas aplicações. Com a programação é possível gerar qualquer série de números quando os elementos possuem uma definição matemática em função de elementos anteriores. Os elementos iniciais, que servem de base para a geração dos demais elementos são chamados de elementos base.

Alguns exemplos:

a) Série de números inteiros pares: 2, 4, 6, 8, 10, 12, ..., $(2*k)$

A formula geral do elemento da posição k é $2*k$. Usando computador, o elemento base é o valor 2 e a geração dos próximos valores pode ser feita usando um comando repetitivo e, em cada iteração o novo elemento será obtido somando-se 2 ao valor elemento atual.

b) Série de Fibonacci: 1, 1, 2, 3, 5, 8, 13, 21, ..., $(elem(k-1) + elem(k-2))$

A formula geral do elemento da posição k é a soma dos 2 elementos anteriores (posição $k-1$ e posição $k-2$), para $k > 2$. Usando computador, os elemento base são os valores 1 e 1 (primeiro e segundo elementos) e a geração dos próximos valores (a partir do terceiro) pode ser feita usando um comando repetitivo e, em cada iteração o novo elemento será obtido somando-se os dois elementos anteriores.

Um algoritmo genérico para a geração de uma série de números pode ser definido com os seguintes passos:

```
Defina N como o número de elementos da série;
Defina o(s) elemento(s) base da série
Use o(s) elemento(s) gerado(s)
Inicializa o contador de elementos gerados
enquanto "contador de elementos gerados < N" :
    Gere o próximo elemento da série
    Use o elemento gerado
    Incremente o contador de elementos gerados
```

O algoritmo acima mapeado para gerar e imprimir a série de números pares seria:

```
N = leia('Número de elementos da série')
elem = 2
Imprima(elem)
contador = "valor inicial"
enquanto "contador < N" :
    elem = elem + 2
    Imprima(elem)
    contador = contador + 1
```

Lembre-se que, na linguagem Python, comandos em um mesmo bloco DEVEM TER A MESMA INDENTAÇÃO, ou seja, mesmo espaçamento antes de cada um dos comandos do bloco, como mostrado no algoritmo acima.

Instruções

Nome do arquivo a ser entregue: **p05.py**

A entrega deverá ser feita até 20h20 (1h50 após o início do horário de sua aula prática).

Obs.: Recomenda-se salvar o arquivo com certa frequência para não perder a digitação já feita em caso de uma falha na rede elétrica.

Faça o “download” do arquivo **p05.py**. Em seguida, entre no IDLE e abra o arquivo através do menu **File > Open...**. Segue abaixo um código semelhante ao que você deverá ver na tela:

```
# Nome do aluno:
# Matrícula:
# Data:

# Escreva aqui um breve comentário dizendo o que o programa 1 faz
#                                     ← Espaço para o comentário
# Escreva o código para a resolução abaixo
```

Para executar o seu programa e verificar a resposta, apenas pressione F5.

📄 Não esqueça de preencher o cabeçalho com seus dados e uma breve descrição do programa. Após certificar-se que seu programa está correto, envie o arquivo do programa fonte (**p05.py**) através do sistema do LBI.

Roteiro de Prática

Use o arquivo **p05.py** para implementar a solução da questão a seguir.

- 1) Escreva um programa para gerar e imprimir os N primeiros números da série descrita abaixo:

$$-0.333, 1.000, -1.667, 2.333, -3.000, 3.667, \dots, \frac{2 \times k - 1}{(-1)^k \times 3}$$

Ou, escrevendo na forma de fração a série seria:

$$-1/3, 3/3, -5/3, 7/3, -9/3, 11/3, \dots, \frac{2 \times k - 1}{(-1)^k \times 3}$$

Ou seja, o numerador tem a série de números ímpares, iniciando com o 1, enquanto que o denominador é sempre o valor 3, alternando, o sinal (negativo e positivo).

O programa deverá calcular e imprimir os elementos da série, no formato de frações e a soma de todos os elementos.

A quantidade de elementos a serem gerados deve ser fornecida pelo usuário do programa e o programa deve aceitar apenas valores maiores que 2 (para gerar pelo menos 3 elementos).

O uso do comando repetitivo WHILE é OBRIGATÓRIO!


A figura a seguir mostra 2 execuções sucessivas do programa:

```
N = 10

Série = {1/-3, 3/3, 5/-3, 7/3, 9/-3, 11/3, 13/-3, 15/3, 17/-3, 19/3}
A soma dos 10 elementos da série é: 3.3333
>>>
===== RESTART: /home/goulart/p05-solucao.py =====
N = -8
N deve ser maior que 2!
N = 0
N deve ser maior que 2!
N = 2
N deve ser maior que 2!
N = 5

Série = {1/-3, 3/3, 5/-3, 7/3, 9/-3}
A soma dos 5 elementos da série é: -1.6667
```

Dica: a diretiva de impressão que altera o terminador padrão e mantém a próxima impressão na mesma linha é: **end="<terminador>"**, onde <terminador> é o que será escrito no final da linha impressa. Ex: end="" # escreve nada (string nula)

 A saída do seu programa deve obedecer à formatação **exata** mostrada no exemplo acima (incluindo o sinal negativo no denominador).

Após certificar-se que seu programa está correto, envie o arquivo do programa fonte (**p05.py**) através do sistema do LBI.

A entrega deverá ser feita até 1h50 após o início do horário de sua aula prática.