

## Lista de Exercícios N° 2

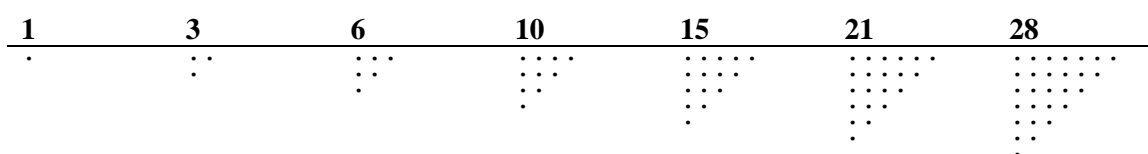
Esta lista pode ser desenvolvida em grupos de até dois alunos, seguindo as especificações contidas no arquivo 00\_ProcListas.pdf disponível na área pública da disciplina na rede. A entrega ao professor deverá ocorrer até o dia 26 de junho de 2018.

- 1) Faça um programa que recebe um número inteiro positivo informado pelo usuário e determina:
- a) se ele é um número primo ou não. Lembre-se que um número é primo se ele é maior que 1 e não possui divisor.
  - b) se ele é um número triangular.
  - c) se ele é um número de Fibonacci. Caso seja, informar qual termo ele é da sequência de Fibonacci, lembrando que o termo 0 da sequência é o próprio 0, o termo 1 é 1, o termo 2 é 1, o termo 3 é 2, etc.

Utilize uma função para determinar cada característica (ou seja, uma para verificar se é primo, outra para se é triangular, outra ainda para verificar se é um número da sequência de Fibonacci). O programa deverá receber vários números, encerrando-se quando for informado o número -999.

### Dica:

Um número triangular é aquele que pode ser representado na forma de um triângulo equilátero. Os primeiros números triangulares são 1, 3, 6, 10, 15, 21 e 28, conforme ilustra a figura a seguir.



Uma definição recursiva da geração do  $n$ -ésimo número triangular é:

- Primeira linha tem  $n$  elementos
  - Soma-se a próxima linha com  $n - 1$  elementos até que reste apenas um elemento.
- 2) Uma importante medida estatística é a variância, usada para avaliar o grau de dispersão de um conjunto de números em relação à sua média. A fórmula da variância é dada por

$$V = \frac{\sum (x - m)^2}{N}$$

onde  $N$  é a quantidade de elementos e  $m$  é a média aritmética simples dos valores, que é dada por

$$m = \frac{\sum x}{N}$$

Escreva um programa que recebe inicialmente um valor de  $N$  (um inteiro entre 1 e 100), indicando a quantidade de números a serem considerados. Em seguida o programa deverá receber e armazenar em um vetor  $N$  números reais de precisão simples (tipo `float` da linguagem C), calculando e imprimindo (com 4 casas depois da vírgula) a média e a variância desses valores conforme as fórmulas anteriormente apresentadas, também utilizando números reais de precisão simples. Usar uma função para calcular a média dos

## Lista de Exercícios N° 2

valores do vetor e outra função para calcular a variância. Imprimir os resultados apenas na rotina principal (a rotina `main()`).

*Exemplo de entrada:*

```
12
1.0  3.5  0.5  8.9  19.2  88.91  1.23  8.42  7.98  -12.52  128.39  -200.57
```

*Saída esperada para o exemplo de entrada:*

```
Media: 4.5783          Variância: 5426.6636
```

- 3) Existem inúmeras formas conhecidas de se calcular o valor aproximado da constante numérica  $\pi$ . Neste exercício você deverá criar duas funções, cada uma implementando um método específico descrito a seguir, e exibir o resultado obtido por cada uma delas.

**Método A:** O valor aproximado de  $\pi$  pode ser calculado usando-se a série:

$$S = \frac{1}{1^3} - \frac{1}{3^3} + \frac{1}{5^3} - \frac{1}{7^3} + \frac{1}{9^3} - \dots$$

sendo  $\pi = \sqrt[3]{S * 32}$

A subrotina deverá calcular e retornar o valor de  $\pi$  com  $S$  sendo gerado a partir de 51 termos (o exemplo dado anteriormente tem 5 termos). Considere que a função pré-definida `pow` permite elevar um número a uma potência.

**Método B:** A série abaixo produz um valor aproximado para  $\pi$

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

Fazer uma subrotina para, por meio dessa série numérica, calcular o valor de  $\pi$  com precisão de 0.0001. Para obter a precisão desejada, adicionar ou subtrair apenas os termos cujo valor absoluto seja maior ou igual a 0.0001.

A rotina principal `main()` deverá exibir na tela os valores produzidos pelas duas sub-rotinas (a do método A e a do método B) com dez casas após a vírgula cada um.

- 4) Faça um programa que abre o arquivo de alunos e verifica se os seus números de CPF (antigo CIC) são válidos ou não. Para ser válido, um CPF não pode ter todos os seus dígitos repetidos e também precisa atender as regras descritas ao final deste enunciado. Para cada registro lido do arquivo, fazer a validação do CPF e emitir uma das mensagens a seguir:
- 'válido': CPF correto
  - 'inválido – esperado: 99, encontrado: 99': dígitos verificadores incorretos, onde o primeiro 99 indica o valor correto dos dígitos verificadores e o segundo 99 representa o par de dígitos verificadores efetivamente encontrado no CPF avaliado
  - 'inválido – todos os dígitos repetidos': o CPF possui todos os seus dígitos repetidos.

Considere que todo CPF lido do arquivo terá exatamente na forma '999.999.999-99' e que o lay-out do arquivo de alunos é dado no quadro a seguir.

## Lista de Exercícios N° 2

**Arquivo:** Alunos02.dat

**Estrutura**

- Código do RA: int
- Nome do aluno: char ...[21]
- CPF do aluno: char ...[15]
- Sexo do aluno: char
- Estado civil do aluno: char

Utilize a rotina de validação de CPF feita para a lista 1, adaptando-a para uma função conforme necessário ao programa.