

Lista 1 - AEDS 2

Prof. Rone Ilídio

1) Escreva um programa que o usuário informe seu nome e sua idade, separadamente, e o computador escreva: "Fulano de Tal, sua idade é: XX" (onde Fulano de Tal é o nome da pessoa e XX sua idade).

2) Crie um programa que receba do usuário o valor que ele deseja aplicar em um fundo de renda fixa, o juro que será acrescido a esse montante todo mês e a quantidade de meses que o dinheiro ficará aplicado. Ao final, o programa deve exibir a quantidade de dinheiro que o usuário terá. A fórmula para calcular juros compostos é:

$$VF = VP * (1+J)^n$$

Onde VF é o valor futuro (dinheiro no final da aplicação), VP é o valor presente (montante inicial), J o juro e n o número de meses.

3) Crie um programa que calcule a média dos salários de uma empresa. O programa deve pedir um salário de cada vez. Quando o usuário inserir um salário menor que 0, o programa deve exibir a média.

4) Escreva um programa que calcule o índice de massa corporal (IMC) de uma pessoa. Tal programa deve receber o peso (em Kilograma) e a altura (em metros). Após isso, ele deve exibir o IMC e qual é o grau de obesidade da pessoa. O IMC é calculado pela seguinte fórmula:

$$IMC = \text{peso} / \text{altura}^2$$

A tabela a seguir mostra o grau de obesidade:

IMC	Classificação
menor que 18,5	Abaixo do peso
18,6 a 25,9	Saudável
25 a 29,9	Peso em excesso
30 a 34,9	Obesidade grau I
35 a 39,9	Obesidade grau II
maior que 40	Obesidade mórbida

Observação: crie um método que recebe o peso e a altura e retorne o IMC.

5) Crie um programa que recebe do usuário os parâmetros a, b e c de uma equação do segundo grau e exibe para o usuário os valores x1 e x2, respostas da equação. Após isso, o programa deve perguntar se o usuário deseja sair do programa ou realizar outra operação. Crie uma função para calcular cada uma das raízes e uma função para calcular o delta.

6) Escreva um programa para controlar o consumo de combustível de seu carro. Esse programa deve pedir ao usuário quantos quilômetros foram

percorridos (int) e quantos litros foram gastos (int). O programa deve imprimir quantos quilômetros (float) foram percorridos por litro. Ao final este programa deve oferecer ao usuário a opção de fazer outro cálculo ou finalizar o programa.

7) Crie um programa para calcular os salários dos empregados de uma empresa. Ele deve receber o número de horas trabalhadas pelo empregado no mês, o valor da hora trabalhada, o número de dependentes e os descontos. O cálculo do salário utiliza a seguinte fórmula:

Salário = número de horas * valor da hora + (50 * número de dependentes) - descontos

8) Crie um programa que receba do usuário os valores (double) dos dois catetos de um triângulo retângulo e mostre para o usuário o valor da hipotenusa ($a^2 = b^2 + c^2$). Tal cálculo deve ser feito dentro de uma função denominada "hipotenusa". Lembre-se, o método `Math.sqrt(x)` retorna a raiz quadrada do double "x" que é passado como parâmetro.

9) Crie um programa que receba uma temperatura, ofereça para o usuário a opção de conversão de Celsius para Fahrenheit ou de Fahrenheit para Celsius e exiba o resultado da conversão.

As fórmulas para conversão são:

Celsius (C) para Fahrenheit (F): $F = 9.0 / 5.0 * C + 32$

Fahrenheit (F) para Celsius (C): $C = 5.0 / 9.0 * (F - 32)$

Cada fórmula deve estar em um método, que recebe os parâmetros necessários e retorna o resultado.

10) Faça um programa que receba do usuário 10 valores inteiros e imprima tais valores na ordem inversa que foram inseridos.

11) Faça um programa que receba do usuário 10 valores *float*. Após isso ele deve imprimi-los e imprimir qual é o maior valor, o menor valor e a média destes valores.

12) Crie um programa onde o usuário forneça os quinze valores inteiros de um vetor, logo após tal programa deve pedir um número ao usuário e multiplicar todos os valores deste vetor pelo número fornecido pelo usuário. Ao final, o vetor deve ser ordenado e exibido na tela.

13) Faça um programa que possui um vetor de inteiro (10 posições), preenchido com valores passados pelo usuário. Tal programa deve pedir para que um segundo usuário tente adivinhar um dos números que estão no vetor. O usuário deve ter no máximo 5 chances para tentar acertar, caso ele não consiga uma mensagem deve aparecer informando que ele "perdeu".

14) Crie um programa com dois vetores, um de 3 e outro de 15 posições. O primeiro vetor é fixo e deve ser preenchido no momento de sua criação com os valores 2, 5 e 7. O segundo deve receber valores do usuário. Tal

programa deve verificar se o primeiro vetor está contido dentro do segundo. Ex:

Primeiro vetor: 2 5 7

Segundo vetor: 3 4 3 5 3 6 3 2 5 7 8 2 1 5

No exemplo, o primeiro vetor está contido dentro do segundo a partir da posição de índice 7

15) Faça um programa que possua um menu com as seguintes opções:

- 1 - Calcular a área de um quadrado
- 2 - Calcular a área de um círculo
- 3 - Calcular a área de um triângulo
- 4 - Sair

Em cada uma das opções o usuário deve pedir para que o usuário informe os dados necessários antes da exibição do resultado. Em outras palavras, para a opção 1 o usuário deve informar o lado do quadrado, para opção 2 o raio do círculo e para a opção 3 a base e a altura do triângulo. Contudo, cada cálculo de área deve ser realizado dentro de seu próprio método.

16) Crie um programa onde o usuário preencha um vetor de 10 posições. Tal programa deverá, então, exibir esses valores ordenados.

17) Crie um *programa* que gere um número aleatório X entre 1 e 100 em um método denominado *aleat*. Tal programa deve pedir para que o usuário tente adivinhar o valor de X . Se ele errar, deve aparecer na tela que o número informado pelo usuário é maior ou menor que X e logo depois deve ser oferecida nova oportunidade para que o usuário tente adivinhar tal número. O usuário deve poder tentar adivinhar quantas vezes quiser, mas quando ele acertar o valor de X , deverá aparecer na tela o número de vezes que ele gastou até adivinhar.

18) Faça um programa que peça para o usuário um número N . Esse programa deve criar um vetor com os N primeiros números da sequência de Fibonacci.

Sequência de Fibonacci é uma sequência de números na qual o primeiro e o segundo são 1 e a partir do terceiro, todo número é igual à soma de seus dois antecessores.

Segue a série: 1 1 2 3 5 8 13 21 ...

19) (4 pontos) Crie um programa que faça a conversão entre Real e Dólar. A informação de quantos Reais valem um Dólar deve ser passada pelo usuário no início da execução. Logo após isso, esse *programa* deve pedir para que o usuário informe o valor em Reais a ser convertido. O resultado da conversão deve, então, ser apresentado para o usuário. Contudo, é obrigatório que as conversões entre as moedas sejam feitas por um método denominado *converte*. Depois da apresentação de uma conversão, o *programa* deve perguntar ao usuário se ele deseja sair ou fazer outra conversão.

Obs1: só é necessário que o usuário informe quantos Reais vale um Dólar uma única vez.

Obs2: o número de casas decimais não será levado em consideração, de forma que se uma resposta tiver uma ou mais de duas casas decimais ela também será considerada correta.

20) Um professor precisa de um programa para obter dados estatísticos sobre uma de suas provas. Crie um programa para auxiliá-lo. As notas de cada um dos alunos devem ser informadas por tal professor (entre 0 e 100) e armazenadas em um vetor do tipo double (60 posições). Ao final, o programa deve exibir na tela qual a porcentagem de alunos que passou, qual a porcentagem de alunos reprovados, qual a maior e qual a menor nota. Considere aprovados os alunos com nota igual ou acima de 60.

21) Crie um *programa* que receba do usuário um número double de cada vez, sendo que o último sempre será 0 (zero). Para cada número, com exceção do zero, tal *programa* deve informar sua parte inteira e sua fracionária.

22) Crie um programa que calcule e exiba a frequência de saída de cada uma das faces de um dado. O programa deve pedir para o usuário informar o número de vezes que o dado será jogado e exibir quantas vezes cada face saiu. Para isso, crie uma função denominada jogardado(), a qual retorna um número aleatório entre 1 e 6, inclusive. Crie um vetor de inteiros de 6 posições. Toda vez que a função for chamada, seu retorno deve ser utilizado para incrementar uma posição do vetor. Exemplo: na primeira chamada da função (que corresponde ao lançamento do dado) o retorno é 4. Com isso, o valor contido dentro da quarta posição do vetor deve ser incrementado em uma unidade. Esse procedimento deve ser repetido N vezes, sendo que N é informado pelo usuário. Ao final, exiba os números contidos no vetor, os quais correspondem a quantas vezes cada face saiu.

23) Crie um programa que receba do usuário os 8 primeiros elementos de um vetor de 9 posições. Contudo esse vetor só deve receber valores 0 e 1. O último valor deve ser inserido pelo programa e deve ser 0 se a quantidade de números 1 contidos no vetor for par e 0 se a quantidade de números 1 contidos o vetor for ímpar. Ex1:

1	0	1	1	0	0	0	1	
---	---	---	---	---	---	---	---	--

O nono elemento deverá receber o valor 1 pois existe 4 números 1 no vetor (e 4 é par).

Ex2:

0	1	1	0	0	1	0	0	
---	---	---	---	---	---	---	---	--

O nono elemento deverá receber o valor 0 pois existe 3 números 1 no vetor (e 3 é ímpar).

Obs: Esse processo é chamado de bitparidade par e é utilizado para conferir erros em transmissões de dados em redes de computadores.

24) Crie um programa para armazenar a umidade relativa do ar coletada em uma floresta durante vários dias. Crie uma struct chamada *dia*, com os campos: *data (string)* e *umidade(double)*. Crie um vetor do tipo *dia* (100) posições e faça o usuário fornecer informações até completar o vetor. Ao final, exiba na tela todos os dados de todos os dias que tiveram umidade acima da média das umidades.

25) Crie uma struct chamada *empregado*, com os seguintes campos: *nome(string)*, *salario(double)*. Crie um programa com um vetor do tipo *empregado*. Tal programa deve ter um menu para manipular empregados, no seguinte formato:

1-Inserir um empregado

2-Exibir média dos salários

3-Exibir empregados com salário superior ao informado pelo usuário

4-Sair

Se o usuário escolher a opção 1, o programa deve pedir ao usuário os dados para inserir UM ÚNICO empregado. A opção 2 exibe a média de todos os salários e a opção 3 exibe os dados daqueles empregados com salário superior a um valor informado pelo usuário.