

EXERCÍCIO ESTRUTURA DE DADOS – Entrada/Saída, controle de fluxo e funções

Lista de Exercícios 01-N1 – Instruções:

Cada programa deve ser salvo em um arquivo-fonte separado com o seguinte formato:

QuestaoXX.c onde **XX** é o número da questão.

Ao final, o aluno deve selecionar todos os arquivos-fonte (os arquivos com a extensão .c) e compactá-los em um ÚNICO arquivo que, este sim, será enviado (**o formato pode ser zip, rar ou tar**).

OBSERVAÇÃO: TODOS os programas entregues devem ter o seguinte cabeçalho:

```
/*
    Função :
    Autor :
    Data :
    Observações :
*/
```

Onde deverá estar escrito o que o programa faz, o autor (nome, turma, a data e as observações que forem pertinentes. Os trabalhos **são individuais** e não serão aceitos após a data.

Enviar o arquivo compactado para o Blackboard (E SOMENTE PARA LÁ).

Faça os seguintes programas:

- Sejam **a** e **b** os catetos de um triângulo retângulo cuja hipotenusa **h** é obtida pela equação:
 $h = \sqrt{a^2 + b^2}$. Faça um programa que leia os valores de **a** e **b**, e calcule o valor da hipotenusa através da fórmula dada. Imprima o resultado.
- Criar um programa para calcular e imprimir o salário líquido de um (pobre) professor. Os dados fornecidos serão: o valor da hora-aula, número de aulas dadas no mês e percentual de desconto do INSS.
- Fazer uma função chamada cubo que retorna o cubo de um número.
- Fazer uma função que retorna o valor do produto notável quadrado da soma de dois termos = $(x + y)^2 = x^2 + 2xy + y^2$ entrando com x e y e retornando o produto notável (usar a função quadrado vista anteriormente)
- Fazer o mesmo para o cubo da soma de dois termos = $(x + y)^3 = x^3 + 3x^2y + 3xy^2 + y^3$. Usar as funções quadrado e cubo anterior.
- Faça um programa que leia 4 valores reais (a, b, c e d) e determine:
 - A média aritmética;
 - A média harmônica (MH);
 - A média geométrica (MG);
 - A média quadrática(MQ).
 - Considere que:

$$MH = \frac{4}{\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d}} \quad MG = \sqrt[4]{a \cdot b \cdot c \cdot d} \quad MQ = \sqrt{\frac{a^2 + b^2 + c^2 + d^2}{4}}$$

7. Dados os pontos A, de coordenadas **A(x₁, y₁)** e B de coordenadas **B(x₂, y₂)**, escreva um programa que determine a distância entre os dois pontos. Considere que:

$$distancia = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

8. Fazer uma função para calcular a área de uma esfera, dado $A = 4\pi r^2$. Considere $\pi = 3.14159265$.
Protótipo:
double areaEsfera (float raio);
9. Criar uma função que calcule a diferença entre duas horas distintas de um mesmo dia e retorne o intervalo de tempo entre os dois horários (ler no formato **hh:mm** e imprimir no formato **hh:mm**)
10. Escrever uma função para calcular a velocidade média de um percurso e apresentar no formato km/h.
Protótipo:
float calculaVM (float espaco, float tempo);
11. Implementar uma função que diga se um número é ou não primo (números primos são os divisíveis somente por um e por eles mesmos). (Retornar 1 se primo e 0 se não primo).
12. Faça uma função que retorne o somatório de um número num. (somatório=soma de todos os números de 1 a N)
Protótipo:
int somatorio (int num);
13. Faça uma função que leia um ângulo em graus e retorne o valor do mesmo convertido para radianos. A fórmula de conversão é $R = G * \pi / 180$, sendo G o ângulo em graus e R em radianos e **pi = 3.14159265**
Protótipo:
float converteGrauRadiano (float angulo);
14. Criar um programa para ler vários números e imprimir todos os divisores de cada número digitado. A leitura se encerra com um número negativo.
15. Considere que dispõe de um capital **x** que rende **j%** de juro ao ano. Escreva uma função que leia **x** e **j** e mostre o capital acumulado com o juro ao final de **n** anos. (Juros compostos)
16. Faça um programa que receba o número do mês e mostre o mês correspondente. Valide mês inválido. (Exemplo: entrada: 08, saída: Agosto. Entrada: 13, saída: Mês inválido!)
17. Implemente uma função para calcular o número de arranjos de **n** objetos diferentes, onde **r** objetos são escolhidos de cada vez. A fórmula é:

$$A_r^n = \frac{n!}{(n - r)!}$$

18. A concessionária de veículos “FERMAT” está vendendo os seus veículos com descontos especiais para aproveitar o IPI reduzido. Faça um programa que calcule e exiba o valor do desconto e o valor a ser pago pelo cliente. O desconto deverá ser calculado sobre o valor do veículo de acordo com o combustível (álcool – 28%, gasolina – 22% ou diesel –12%). Com valor do veículo zero encerra entrada de dados. Informe total de desconto e total pago pelos clientes.
19. Um determinado material radioativo perde metade de sua massa a cada 50 segundos. Dada a massa inicial, em gramas, faça um programa que determine o tempo necessário para que essa massa se torne menor que 0,05 gramas

20. Crie uma aplicação para calcular o IMC (Índice de Massa Corporal) que leia o peso do usuário em quilogramas e a altura em metros e que depois calcule e apresente o IMC segundo a fórmula:

$$IMC = \frac{\text{pesoEmQuilos}}{\text{alturaEmMetros} \times \text{alturaEmMetros}}.$$

Além disso, o programa deverá exibir as informações seguintes do Ministério da Saúde para que a pessoa possa avaliar seu IMC:

VALORES DE IMC

Abaixo do peso: menor que 18,5

Normal: entre 18,5 e 24,9

Acima do peso: entre 25 e 29,9

Obeso: 30 ou mais

21. A sequência de Fibonacci é uma sequência de números inteiros, começando normalmente por 0 e 1, na qual, cada termo subsequente (número de Fibonacci) corresponde a soma dos dois anteriores. A sequência leva o nome de seu criador que descreveu, em 1202, o crescimento de uma população de coelhos a partir desta. A sequência pode ser gerada de diversas formas. Uma delas usa uma função geradora na seguinte forma: $a_n = \frac{\phi^n - (-\phi)^{-n}}{\sqrt{5}}$. Onde $\phi = 1.6180339887498$. Faça um programa onde o usuário digita um número **n** e a seguir ele exiba TODOS os números de Fibonacci de 0 até **n** usando a função geradora acima. Calcule o número de Fibonacci usando uma função.

DESAFIO: Crescimento Populacional (opcional)

Adaptado por Neilor Tonin, URI  Brasil (beecrowd 1160)

Mariazinha quer resolver um problema interessante. Dadas as informações de população e a taxa de crescimento de duas cidades quaisquer (A e B), ela gostaria de saber quantos anos levará para que a cidade menor (sempre é a cidade A) ultrapasse a cidade B em população. Claro que ela quer saber apenas para as cidades cuja taxa de crescimento da cidade A é maior do que a taxa de crescimento da cidade B, portanto, previamente já separou para você apenas os casos de teste que tem a taxa de crescimento maior para a cidade A. Sua tarefa é construir um programa que apresente o tempo em anos para cada caso de teste.

Porém, em alguns casos o tempo pode ser muito grande, e Mariazinha não se interessa em saber exatamente o tempo para estes casos. Basta que você informe, nesta situação, a mensagem "Mais de 1 século."

Entrada

A primeira linha da entrada contém um único inteiro **T**, indicando o número de casos de teste ($1 \leq T \leq 3000$). Cada caso de teste contém 4 números: dois inteiros **PA** e **PB** ($100 \leq PA < 1000000$, $PA < PB \leq 1000000$) indicando respectivamente a população de A e B, e dois valores **G1** e **G2** ($0.1 \leq G1 \leq 10.0$, $0.0 \leq G2 \leq 10.0$, $G2 < G1$) com um dígito após o ponto decimal cada, indicando respectivamente o crescimento populacional de A e B (em percentual).

Atenção: A população é sempre um valor inteiro, portanto, um crescimento de 2.5 % sobre uma população de 100 pessoas resultará em 102 pessoas, e não 102.5 pessoas, enquanto um crescimento de 2.5% sobre uma população de 1000 pessoas resultará em 1025 pessoas. Além disso, não utilize variáveis de precisão simples para as taxas de crescimento.

Saída

Imprima, para cada caso de teste, quantos anos levará para que a cidade A ultrapasse a cidade B em número de habitantes. Obs.: se o tempo for mais do que 100 anos o programa deve apresentar a mensagem: Mais de 1 século. Neste caso, acredito que seja melhor interromper o programa imediatamente após passar de 100 anos, caso contrário você poderá receber como resposta da submissão deste problema "Time Limit Exceeded".

Exemplo de Entrada	Exemplo de Saída
6 100 150 1.0 0 90000 120000 5.5 3.5 56700 72000 5.2 3.0 123 2000 3.0 2.0 100000 110000 1.5 0.5 62422 484317 3.1 1.0	51 anos. 16 anos. 12 anos. Mais de 1 século. 10 anos. 100 anos.

A última questão (23) é um desafio de modo que não é obrigatória!

Bom Exercício.

DICA: É IMPORTANTE QUE TODOS OS EXERCÍCIOS SEJAM REALIZADOS!

Para tirar a **raiz quadrada** de um número use a função **sqrt(x)** e para potenciação use a função **pow(x,y)**