Prática 03 – Funções

Desenvolva os seguintes algoritmos em linguagem de programação C:

1. Escreva uma função em C para verificar se um número é par. Retornar 1 se for par e 0 (zero) se for ímpar.

2. Escreva uma função em C para verificar se um número é primo. Retornar 1 se for primo e 0 (zero) se não for.

3. Escreva uma função em C para verificar se um número é múltiplo de outro. Retornar 1 caso afirmativo e 0 (zero) se não for.

4. Reescreva todas as funções anteriores considerando que as variáveis usadas como parâmetros e como retorno são agora **globais**. O que mudará?

5. Escreva uma função em C para calcular o fatorial de um número recebido como parâmetro. A função deve retornar o resultado do cálculo.

6. Escreva uma função em C para calcular o máximo entre dois números recebidos como parâmetro. Retornar o resultado.

7. Escreva uma função em C para identificar se um caracter é um dígito entre ´0´ (inclusive) e ´9´ (inclusive). A função recebe como parâmetro um caracter, retornando 1 se o caracter for dígito e 0 caso contrário.

8. Escreva uma função em C para verificar se um caracter é vogal. A função recebe como parâmetro um caracter, retornando 1 se o caracter for vogal e 0 caso contrário.

9. Escreva um programa de adivinhação de número. O programa deve "sortear" um número de 0 a 10 como descrito abaixo e solicitar ao usuário que o adivinhe. Ele terá 3 chances de acertar. Caso não acerte, informar ao final qual era o número.

   Como sortear:

   srand (time(NULL)); // "liga" o gerador de números usando como base a hora atual em segundos

   ```
   x = rand ();  // cria um número qualquer, sem limite
   y = x%11;    // para limitar os numeros de 0 a 10
   → o usuário deve acertar o número armazenado em y
   ```

   Estruture o programa nas seguintes funções, usando variáveis globais:
   cria_numero() – sorteia o valor de y
   le_usr() – lê valor digitado pelo usuário
   mostra() – escreve mensagem para usuário: se conseguiu ou não adivinhar

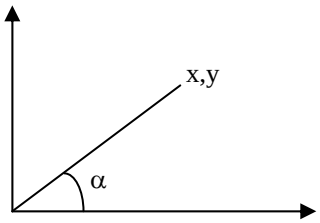
10. Modifique o programa 9 de forma que todas as variáveis sejam criadas no main. Veja as declarações das funções abaixo para essa modificação:

    int cria_numero () – sorteia o valor de y e retorna
    int le_usr () – lê valor digitado pelo usuário e retorna
    mostra (int result) – escreve mensagem para usuário: se conseguiu ou não adivinhar

    result terá o valor 0, se o usuário não conseguiu adivinhar; ou 1 se conseguiu

11. Seja um circuito resistivo com **n** resistores em série. Escreva um programa para calcular e mostrar a resistência equivalente e a corrente total. Escrever para esse programa as seguintes funções:
    - calcReq – lê todos os resistores, calcula e retorna a resistência equivalente
    - le_tensao – lê o valor da tensão na fonte e retorna esse valor
    - calcCorr – recebe os valores de tensão na fonte e resistência equivalente, calcula e retorna a corrente total
    - mostra – recebe a resistência equivalente e a corrente total e mostra na tela
        i. R_{eq} = Resistência equivalente = $\sum R_i$
        ii. $I = V_{fonte}/R_{eq}$; onde V_{fonte} = Tensão na fonte
        iii. I = corrente total

12. Robôs móveis autônomos são robôs que percorrem um ambiente com uma determinada finalidade e normalmente recebem a coordenada ***x,y*** de um ponto a ser alcançado. Diferentes abordagens podem ser utilizadas no cálculo de rotas e normalmente se utiliza o cálculo vetorial para isso. Escreva um programa para determinar em quanto tempo um robô alcança uma coordenada [x,y], tendo partido do ponto 0,0 em uma trajetória formando um ângulo α com o eixo x (ver figura abaixo):



O usuário digita: v (velocidade), α, x e y. O programa deve funcionar em loop até que o usuário decida encerrar. Crie sua condição de saída.

Estruture nas funções abaixo, usando apenas variáveis locais (declaradas no main):

? Le_dados(?) – lê v, x, y, α
? Calcula_t (?) – calcula o valor de t e retorna
? Mostra(?) – mostra valor de t calculado

Obs.: Sintaxe básica para funções: tipo nome da função (declaração de parâmetros) {
Seqüência de comandos;
}

Detalhes: A declaração dos parâmetros é feita no momento da listagem dos mesmos, pelo padrão atual ANSI. O nome da função deve respeitar as regras de nomenclatura já utilizadas para variáveis (não pode começar com números, deve ter até 32 caracteres sem espaço entre eles, existindo diferença entre caracteres maiúsculos e minúsculos). As funções retornam um valor de acordo com o tipo indicado na sintaxe da função, através do comando *return* (que pode ser um valor ou uma variável).