

Lista de Exercícios 4 - Funções

Bacharelado em Ciência da Computação

Linguagens de Programação II

Professor: Dany Sanchez Dominguez

1. Escreva uma função `DecimalToBinario` que receba um número inteiro e retorne uma cadeia de caracteres com o número em representação binária. Exemplo: se a função receber o numero 17 deverá retornar 100001.
2. Escreva uma função `BinarioToDecimal` que receba uma cadeia de caracteres que represente um número binário (i.e. a cadeia pode ser composta apenas por 0s e 1s) e retorne o número em notação decimal. Exemplo: se a função receber 10101 deverá retornar 22.
3. Escreva uma função que receba um número inteiro e retorne o mesmo número em ordem inversa. Exemplo: se a função receber 1975 ela deverá retornar 5719.
4. Defina uma função `Hipotenusa` que calcule o comprimento da hipotenusa de um triângulo retângulo, ao serem fornecidos os catetos. Use a função em um programa para determinar o comprimento da hipotenusa onde o comprimento dos catetos são lidos pelo programa. A função deve receber dois argumentos `float` e retornar um valor `float`.
5. Crie um programa que leia a *base* e o *expoente* e imprima $base^{expoente}$. Admita que o expoente é um valor inteiro (positivo, negativo ou zero). Escreva uma função `PotenciaInt`, que utilize um `for` para controlar o calculo. Não use a função `power()`.
6. Escreva um programa que receba n duplas de números inteiros e determine se o segundo é múltiplo do primeiro. Escreva uma função `Multiplo` que receba dois números inteiros e retorne 1 se o segundo for múltiplo do primeiro ou 0 em caso contrário.
7. Diz-se que um número é perfeito se a soma de seus fatores, incluindo 1 (mas não o número em si), resulta no próprio número. Por exemplo 6, é um número perfeito porque $6 = 1 + 2 + 3$. Escreva uma função `perfeito` que determine se o parâmetro `numero` é um numero perfeito. Use essa função em um programa que determine e imprima todos os números perfeitos entre 1 e 100. Imprima os fatores de cada número encontrado para confirmar que ele é realmente perfeito.
8. Diz-se que um número inteiro é primo se for divisível apenas por 1 por se mesmo. Por exemplo, 2, 3, 5, e 7 são números primos, mas 4, 6, 8 e 9, não são.
 - a) Escreva uma função que determine se um número é primo.
 - b) Use essa função em um programa que determine e imprima todos os números primos entre 1 e 1000.
9. Escreva um programa que imprima os primeiros n números primos onde n é um valor informado pelo usuário. Considere utilizar alguma das funções implementadas no programa anterior.

10. Escreva um programa que simule o lançamento de uma moeda. Para cada lançamento da moeda o programa deve imprimir **Cara** ou **Coroa**. Deixe o programa lançar a moeda 100 vezes e conte o número de vezes que cada lado da moeda aparece. Imprima os resultados. O programa deve chamar uma função separada *jogada* que não utiliza argumentos e retorna 0 para coroa e 1 para cara.
11. Escreva uma função que receba a hora do dia em segundos (a quantidade de segundos desde a medianoite) e imprima a hora no formato *hh:mm:ss*. Por exemplo se receber 0 imprime 00:00:00, se receber 3725 imprime 01:02:05. A função deve verificar que recebe uma quantidade de segundos válida, entre 0 e a quantidade de segundos que tem um dia.
12. Escreva uma função que calcule o fatorial de um numero inteiro. Utilize essa função num programa que imprime os fatoriais dos números entre 1 e n , onde n e fornecido via teclado.
13. Escreva uma função que receba as coordenadas de dois pontos (x_0, y_0) e (x_1, y_1) e retorne a distancia entre esses dois pontos. Use a função num programa que recebe uma lista de pares de pontos e imprime a distancia entre eles, e a distancia média. A quantidade de pares de pontos é desconhecida. Utilize uma estrutura para representar os pontos.
14. Escreva um programa que gere n números inteiros aleatórios entre 1 e 100. O valor de n deve ser lido pelo teclado. Seu programa deve informar:
 - a) O maior numero,
 - b) O menor numero,
 - c) O produtório dos números pares,
 - d) O somatório dos números ímpares,
 - f) Imprimir a lista de números,
15. Escreva uma função para desenhar quadrados na tela. A função recebe dois parâmetros; o primeiro toma valores 0 ou 1, 0 indica quadrado vazado e 1 quadrado cheio; o segundo parâmetro a quantidade de asteriscos no lado do quadrado. Utilize a função num programa que imprima o seguinte padrão:

```
*****
*****
*****
*****

*****
*   *
*   *
*   *
*   *
*****
```

16. Escreva um programa para processar os resultados de uma maratona, para cada atleta seu programa deverá considerar (nome, idade, tempo (-1 se o atleta não termino a maratona), e sexo). A quantidade de atletas que participam da maratona é informada pelo usuário. Seu programa deverá:
- a) Ler os dados de cada atleta.
 - b) Informar o pódio (primeiro, segundo e terceiro) das modalidades femininas e masculina.
 - c) Informar a quantidade de atletas que participaram e a porcentagem dos que terminaram a corrida.

Crie as funções: `LeAtleta()` para ler os dados de um atleta e `PrnAtleta()` para imprimir os dados de um atleta.

17. Crie um programa para simular um jogo de dados com as seguintes regras: *três dados são lançados se todos os dados tiveram o mesmo valor, o jogador fez uma "trinca" e vence. Se a soma dos dados for par o jogador ganha, em todos os outros casos a banca ganha.* Crie uma função chamada `jogada` para realizar o processamento, o resultado do lançamento deve ser mostrado na tela.
18. Escreva um programa que simule um jogo de dardos, três dardos são lançados a um alvo de 20 cm de diâmetro, se dois deles ficam a menos de 1 cm do centro do alvo o jogador recebe um premio. Se os três dados ficam fora do alvo o jogador recebe uma penalidade. Utilize uma estrutura `Ponto` para representar o ponto do alvo onde o dardo acerta, os valores de x e y serão gerados aleatoriamente.
19. Escreva um programa que leia dois vetores de tamanho arbitrário e faça a multiplicação dos elementos de mesmo índice, colocando o resultado em um terceiro vetor. Mostre o vetor resultante. Crie funções para fazer a leitura de um vetor, imprimir um vetor e fazer a multiplicação ponto a ponto.
20. Escreva um programa que leia um vetor G de 10 elementos caracter que representa o gabarito de uma prova. A seguir, para cada um dos n alunos de uma turma, leia o vetor de respostas (R) do aluno e conte o número de acertos. Mostre o nº de acertos do aluno e uma mensagem APROVADO, se a nota for maior ou igual a 6; e mostre uma mensagem de REPROVADO, caso contrário. Cada questão correta vale 1,0 ponto.
21. Faça um programa que leia vetor de n posições de números reais e mostre um menu com as opções: terminar o programa, mostrar o vetor na ordem direta e mostre o vetor na ordem inversa. Seu programa dever ser executado até que a opção terminar o programa seja escolhida.
22. Escrever um programa que lê 2 vetores X e Y de m e n elementos respectivamente. Crie, a seguir, um vetor Z que seja
- a) a união de X com Y ;
 - b) a intersecção entre X e Y .

Mostre o vetor Z em cada caso.

23. Escreva uma função que receba uma string e retorne o caractere que mais se repete na string e a quantidade de vezes que se repete.

24. Escrever um programa que lê uma matriz $A(m, n)$ de elementos inteiros e a escreva. Verifique, a seguir, quais os elementos de A que estão repetidos e quantas vezes cada um está repetido. Escrever cada elemento repetido com uma mensagem dizendo que o elemento aparece x vezes em A .
25. Na teoria dos sistemas, define-se como elemento *minimax* de uma matriz o menor elemento da linha onde se encontra o maior elemento da matriz. Escreva um programa que gere automaticamente uma matriz $m \times n$ de inteiros e encontre seu elemento *minimax*, mostrando também sua posição.
26. Escreva uma função que recebe um número inteiro M e gere uma matriz crescente até M . Exemplo para $M = 3$:

```
1 1 1 1 1 1
1 2 2 2 2 1
1 2 3 3 2 1
1 2 3 3 2 1
1 2 2 2 2 1
1 1 1 1 1 1
```

27. Crie uma função que recebe duas strings e retorna verdadeiro se elas formam um anagrama, caso contrário retorna falso. Temos um anagrama quando string é igual a outra lida em ordem reversa; exemplo: AMOR - ROMA.
28. Crie uma função que recebe uma string e retorna verdadeiro se a string for um palíndromo, caso contrário retorna falso. Um palíndromo é uma palavra que se lê da mesma maneira em ambos sentidos, exemplos: ovo, osso e radar.
29. Crie uma função que receba uma string e um caracter qualquer. A função deve remover todas as ocorrências do caractere da string.
30. Escreva um programa para ler e imprimir os resultados de uma turma de n alunos em uma disciplina (matrícula, nome, sobrenome, nota). O programa deverá imprimir todos os dados do melhor estudante, e do pior estudante.
31. Utilize uma estrutura para representar um ponto do plano (x, y) , escreva um programa que leia um ponto do teclado e informe a que quadrante pertencê (I, II, III ou IV). O programa ficará em execução até que o ponto $(0, 0)$ seja digitado. Escreva uma função para ler os pontos `Ponto le_ponto(void)`, e outra que recebe um ponto e informa o quadrante `int informa_quad(Ponto)`.
32. Utilize a estrutura do programa anterior para fazer um programa que calcule a distancia entre dois pontos P_1 e P_2 . Define-se a distancia entre dois pontos d , como

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

crie a função `float dist_pontos(Ponto, Ponto)`.

33. Crie uma estrutura para representar um ponto `TPonto` do espaço. Utilize a estrutura `TPonto` para criar uma estrutura `Tesfera` (centro e radio). Escreva um programa usando as estruturas anteriores que receba uma esfera e um ponto, seu programa deve informar se o ponto é interno ou externo a esfera, para isso crie três funções: `le_Ponto()`, `le_Esfera()` e `verif_Ponto_Interior()`.

34. **Gerenciamento de Biblioteca** Escreva um programa para o gerenciamento do acervo de uma biblioteca. Para um livro considere os seguintes dados: Título, Autor e Número de Páginas. Seu programa deverá mostrar um menu com as opções:

- 1 Adicionar livro
- 2 Buscar livro
- 3 Apagar livro
- 4 Listar acervo
- 5 Sair

Recomendamos utilizar funções para as opções 1, 2, 3 e 4. Crie um projeto com arquivos diferentes para o programa principal, os cabeçalhos das funções e as definições das funções.

35. **Jogo de Craps** Um dos jogos de azar mais populares é o jogo conhecido como “craps”. Escreva um programa que simule o jogo de “craps”. As regras do jogo são simples:

Um jogador joga dois dados. Cada dado tem seis faces. Essas faces contêm 1, 2, 3, 4, 5, ou 6 pontos. Depois dos dados pararem a soma das faces superiores do dado é calculada. Se a soma for 7 ou 11 no primeiro lançamento o jogador vence. Se a soma for 2, 3 ou 12 no primeiro lançamento (chamado “craps”), o jogador perde (i.e. a “casa” vence). Se a soma for 4, 5, 6, 8, 9 ou 10 no primeiro lançamento esta soma se torna o “ponto” do jogador. Para vencer, o jogador deve continuar lançando dados até “fazer o ponto”. O jogador perde se tirar um 7 antes de fazer o ponto.

Seu programa deve imprimir o resultado de cada lançamento dos dados.

36. **Agenda telefônica** Crie um programa que simule o funcionamento de uma agenda telefônica, considere a estrutura:

```
typedef struct item{
    char nome[255];
    char telefone[13];
    struct item *ptr;
}Titem;
```

para cada contato da agenda. Seu programa deve mostrar um menu com as opções:

- (a) Inserir um contato,
- (b) Apagar um contato,
- (c) Buscar um contato,
- (d) Listar todos em ordem alfabética,
- (e) Sair

Seu programa deve utilizar alocação dinâmica de memória e uma lista autoreferenciada para armazenar os dados. Implemente funções para cada opção do menu.