



Disciplina: Algoritmos e Estruturas de Dados (AEDA)

1º Semestre de 2024

Professor: Anderson Grandi Pires (agpires@cefetmg.br)

Cada programa abaixo deverá conter uma mudança de linha como última instrução.

Exercícios disponibilizados em 05.03.2024

1. Crie uma função para calcular a soma dos primeiros 100 números naturais. Considere o número 1 como primeiro número natural. Faça um programa para usar a função e apresentar a soma para o usuário.
2. Implemente uma função para calcular a soma dos primeiros 50 números naturais ímpares. Considere o número 1 como primeiro número natural. Faça um programa para usar a função e apresentar a soma para o usuário.
3. Implemente uma função para calcular o produto dos primeiros N números naturais ímpares. Leia um número natural N e calcule o produto dos números naturais ímpares, menores ou iguais a N . Considere o número 1 como primeiro número natural. Faça um programa para usar a função e apresentar o resultado para o usuário.
4. Desenvolver um programa para efetuar o cálculo do valor de S , sendo $S = 1 - 2 + 3 - 4 + \dots$. O número n de termos deverá ser solicitado ao usuário e o resultado da soma deverá ser apresentado.
5. Desenvolver um programa para efetuar o cálculo do valor de S utilizando os n primeiros termos (n é informado pelo usuário), sendo $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$. O resultado da soma deverá ser apresentado.
6. Desenvolver um programa para efetuar o cálculo do valor de S , sendo $S = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots$. O número n de termos deverá ser solicitado ao usuário e o resultado da soma deverá ser apresentado.
7. Desenvolver um programa para efetuar o cálculo do valor de S , sendo $S = 1 + \frac{2}{3} + \frac{3}{5} + \frac{4}{7} + \dots$. O número n de termos deverá ser solicitado ao usuário e o resultado da soma deverá ser apresentado.
8. Desenvolver **um programa para cada padrão abaixo**. Cada programa deve criar o padrão utilizando uma única instrução para imprimir os asteriscos como, por exemplo, `printf("*");`. Logo, estruturas de repetição precisarão ser utilizadas. Instruções como `printf("\n");` podem ser utilizadas para efetuar mudanças de linha.

*
**

**
*

9. Desenvolver uma função que calcule a soma de dois números passados por parâmetro e utilize um terceiro parâmetro para retornar o resultado da soma. Segue uma sugestão de protótipo da função: `void soma(int a, int b, int *s);`. Implemente a função `main` para testar a função criada.
10. Escreva uma função denominada `primo` que receba como parâmetro um inteiro m e dois outros parâmetros `primo_menor` e `primo_maior` passados por referência. A função deve retornar em `primo_menor` o maior número primo que é menor do que m e deve retornar em `primo_maior` o menor número primo que é maior do que m . Crie um programa para testar a função criada.
11. Escreva uma função que receba um número real n (passagem de parâmetros por valor) e retorne a parte inteira e a fracionária por meio de passagem de parâmetros por referência. Crie um programa para testar a função criada.
12. Escreva uma função que receba um vetor de inteiros de tamanho n e retorne o maior elemento presente no vetor. Implemente a função `main` para testar a função criada.
13. Escreva uma função que receba um vetor de inteiros de tamanho n e preenche esse vetor com valores aleatórios no intervalo $[-15, 20]$. Crie um programa para testar a função criada, preenchendo o vetor e imprimindo seu conteúdo.
14. Escreva uma função que receba um vetor de inteiros de tamanho n e um valor v . A função deve retornar a quantidade de valores iguais a v existente no vetor. Crie um programa para testar a função criada gerando números aleatórios para o vetor no intervalo $[10, 100]$.
15. Escreva uma função que receba um vetor de tamanho n e retorne **por referência** o maior e o menor elementos do vetor do vetor. Implemente a função `main` para testar a função criada.
16. Escreva uma função que receba um vetor de tamanho n e retorne **por referência** o maior elemento do vetor e o número de vezes que esse elemento ocorreu no vetor. Implemente a função `main` para testar a função criada.
17. Desenvolver um programa para cada padrão abaixo. Cada programa deve criar o padrão utilizando uma **única instrução para imprimir os asteriscos** como, por exemplo, `printf("*");`. Logo, estruturas de repetição precisarão ser utilizadas. Instruções como `printf("\n");` podem ser utilizadas para efetuar mudanças de linha.

*	*
**	***
***	*****
****	*****
*****	*****

Os exercícios 18 a 23 foram retirado do livro texto (p. 117).

18. Implemente uma função que receba uma string como parâmetro e retorne como resultado o número de vogais nessa string. Essa função deve obedecer ao protótipo:

```
int conta_vogais(char *str);
```

19. Implemente uma função que receba uma string e um caractere e retorne como resultado o número de ocorrências desse caractere na string. Essa função deve obedecer ao protótipo:

```
int conta_char(char *str, char c);
```

20. Implemente uma função que receba uma string e altere nela as ocorrências de caracteres maiúsculos para minúsculos. Essa função deve obedecer ao protótipo:

```
void minusculo(char *str);
```

21. Implemente uma função que receba uma string e substitua todas as letras por suas sucessoras no alfabeto. Por exemplo, a string "Casa" seria alterada para "Dbtb". Essa função deve obedecer ao protótipo:

```
void shift_string(char *str);
```

Observação: A letra z deve ser substituída pela letra a (e Z por A). Caracteres que não forem letras devem permanecer inalterados.

22. Implemente uma função que receba uma string como parâmetro e substitua as ocorrências de uma letra pelo seu oposto no alfabeto, isto é, $a \leftrightarrow z$, $b \leftrightarrow y$, etc. Caracteres que não forem letras devem permanecer inalterados. Essa função deve obedecer ao protótipo:

```
void string_oposta(char *str);
```

23. Implemente uma função que receba uma string como parâmetro e desloque os seus caracteres uma posição para a direita. Por exemplo, a string "casa" seria substituída para "acas". Repare que o último caractere vai para o início da string. Essa função deve obedecer ao protótipo:

```
void roda_string(char *str);
```

24. Desenvolver uma estrutura para representar um círculo no \mathbb{R}^2 . Utilize o comando `typedef` para oferecer um sinônimo (apelido) para a estrutura. Desenvolva um programa para criar dois círculos, atribuir a eles valores para o par ordenado que representa o centro do círculo e um valor para o raio. Apresente na tela os dados dos círculos usando o seguinte formato: `<(x, y) raio>`. Apresente cada círculo em uma linha.

25. Implemente uma função para apresentar as coordenadas de um círculo usando o formato `<(x, y) raio>`.

26. Implemente uma função para calcular a área de um círculo, definido conforme a especificação do exercício 31.

27. Implemente uma função para calcular o perímetro de um círculo, definido conforme a especificação do exercício 31.

28. Desenvolva um programa para criar um círculo. Atribua a suas coordenadas um valor real no intervalo $[-20, 20]$ e para o raio um valor no intervalo $[0, 10]$. Utilize as funções implementadas nos exercícios 25, 26 e 27 para apresentar as informações do círculo, sua área e seu perímetro.

29. Implemente uma função para mover um círculo no \mathbb{R}^2 com base nos deslocamentos em x e y . Crie um programa para mostrar o perfeito funcionamento da função.
30. Desenvolver uma estrutura para representar um quadrado no \mathbb{R}^2 . Utilize o comando `typedef` para oferecer um sinônimo (apelido) para a estrutura. Implemente uma função para calcular a área do quadrado e outra para seu perímetro. Desenvolva um programa para criar um quadrado, atribuir a ele valores e apresente na tela sua área e seu perímetro.
31. Desenvolver uma estrutura para representar um aluno contendo `nome`, `idade`, `sexo` e um vetor com 4 `notas`. Inicie cada informação de um aluno e apresente-as conforme formato abaixo:
Nome: ...
Idade: ...
Sexo: ...
Notas: `n1 n2 n3 n4`
No local de cada ... deve aparecer a informação usada na inicialização. Da mesma forma, no local de cada nota n_i deverá aparecer a i -ésima nota do aluno.
32. Implemente uma função para calcular a média das notas de um aluno. Desenvolva um programa para criar um aluno. Solicite ao usuário os dados do aluno e suas notas no intervalo $[1, 10]$. Calcule a média usando a função criada e mostre-a na tela.
33. Crie um programa para imprimir 30 números pseudo-aleatórios no intervalo $[0, 10]$. Utilize as funções `rand()` e `srand()` da biblioteca `stdlib.h`, além da função `time()` da biblioteca `time.h`.
34. Crie um programa para imprimir 50 números pseudo-aleatórios no intervalo $[0, 50]$. Utilize as funções `rand()` e `srand()` da biblioteca `stdlib.h`, além da função `time()` da biblioteca `time.h`.
35. Crie um programa para imprimir 100 números pseudo-aleatórios no intervalo $[5, 25]$. Utilize as funções `rand()` e `srand()` da biblioteca `stdlib.h`, além da função `time()` da biblioteca `time.h`.
36. Crie um programa para imprimir 100 números pseudo-aleatórios no intervalo $[-15, 15]$. Utilize as funções `rand()` e `srand()` da biblioteca `stdlib.h`, além da função `time()` da biblioteca `time.h`.
37. **(JOGO)** Crie um programa em C que sorteia um número entre 1 e 100. Um jogador (você ou outra pessoa) tenta adivinhar o número sorteado. Uma mensagem na tela deve ser apresentada ao usuário, solicitando-o um palpite entre 1 e 100. Para cada palpite, o programa deve informar se o palpite é maior, menor ou igual ao número sorteado. Caso seja igual, o jogo termina e o jogador vence. Caso contrário, o jogador faz um novo palpite para tentar acertar o número sorteado. Um jogador tem direito a fazer 10 palpites. Caso ele não acerte o número nesses 10 palpites, o computador ganha o jogo.
38. Implemente uma função recursiva para calcular e retornar a soma dos números naturais menores ou iguais a N .
39. Implemente uma função recursiva para calcular o resultado do cálculo de um número a elevado a um número natural b , ou seja, a^b .
40. Implemente uma função recursiva para calcular e retornar o produto dos números inteiros maiores que 0 e menores ou iguais a N .

41. Implemente uma função recursiva para apresentar o conteúdo de um vetor de inteiros, do último ao primeiro item.
42. Implemente uma função recursiva para apresentar o conteúdo de um vetor de inteiros, do primeiro ao último item.
43. Implemente uma função recursiva para somar o conteúdo de um vetor de números reais.