DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA - UFRN - LISTA DE EXERCÍCIOS
DIM0320 - ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES Prof: Gabriel Vasiljevic

Para cada um dos problemas, tente primeiro descrever os dados de entrada e de saída. Em seguida, crie variáveis para os dados de entrada e pense em como obter os dados de saída a partir deles. Todos os exercícios devem ser implementados em C. Tente nomear as funções de forma que fique claro o que cada uma delas faz.

1. Implemente uma função que recebe dois números inteiros como parâmetro e retorna o MDC entre os dois.
2. Implemente uma função que recebe três números reais como parâmetro, e retorna `verdadeiro` caso seja possível construir um triângulo usando estes três valores como lados, e `falso` caso contrário.
3. Escreva uma função que recebe um número inteiro N como parâmetro e retorna `verdadeiro`, caso o número seja primo, e `falso` caso contrário. Utilize esta função para exibir todos os números primos entre 1 e 100.
4. Escreva uma função que recebe um número inteiro e positivo N como parâmetro e retorna o fatorial de N. Considere o tipo de retorno da função como `unsigned long long`.
5. Escreva uma função que recebe um número inteiro e retorna o inverso deste número (e.g., para 12345, a função retorna 54321).
6. Escreva uma função que recebe como argumento uma string e um caractere, e retorna quantas vezes este caractere aparece na string.
7. Escreva uma função que receba três números inteiros como parâmetros, representando horas, minutos e segundos, e retorna um inteiro correspondente aos valores convertidos em segundos. Por exemplo: 2h, 40min e 10seg correspondem a 9610 segundos.
8. Escreva uma função que receba um número inteiro como parâmetro, representando um número de segundos, e imprime o valor correspondente no formato "hh:mm:ss". Por exemplo, para a entrada 9610, a função imprime "02:40:10".
9. A Cifra de César é um dos métodos de criptografia mais antigos conhecidos. Neste método, cada caractere de uma frase é substituído pelo caractere N posições de distância à frente. Por exemplo, para N = 3, a palavra "zebra" se torna "cheud", com cada letra sendo substituída pela letra 3 posições à frente no alfabeto. Para decifrar uma palavra cifrada, basta fazer a operação inversa. Crie uma função que recebe uma string como entrada e um número inteiro N e criptografa essa string usando a Cifra de César.
10. Uma forma mais avançada de criptografia consiste em usar uma chave de criptografia. Neste tipo de criptografia, realiza-se uma operação XOR (OU lógico exclusivo, ou disjunção exclusiva, que é `verdadeiro` somente se um dos dois operandos for `verdadeiro`, mas não ambos) entre os bits de cada caractere da string e cada caractere da chave de criptografia. Em C, a operação XOR entre bits pode ser realizada utilizando o operador '^'. Assim, um pseudocódigo para realizar esta criptografia seria o seguinte:

    ```
    chave: string de tamanho L
    frase: string de tamanho N

    para i de 0 até N-1:
        frase[i] = frase[i]^chave[i%L]
    ```

    Para descriptografar uma mensagem criptografada com este método, é necessário possuir a mesma chave utilizada para realizar a criptografia. Então, basta realizar a mesma operação novamente e a frase será decifrada. Escreva uma função que recebe duas strings como entrada, uma representando uma frase e a outra uma chave de criptografia, e criptografa a frase usando a chave. Por exemplo, com a chave sendo "DIM0320" e a frase "algoritmos", a cifra gerada é "%%*_A[D)&>". Com a frase sendo "%%*_A[D)&>" e usando a mesma chave, a cifra gerada é "algoritmos".
11. Escreva uma função que recebe um nome de usuário (como uma string) e valida este nome. A função retorna `verdadeiro` se o nome for válido, e `falso` caso contrário. Um nome de usuário é válido se só possui letras, números ou *underline* e não começa com um número.
12. Escreva uma função que recebe uma senha (como uma string) como parâmetro e valida esta senha. A função retorna 0, se a senha é válida; -1, caso a senha não tenha nenhuma letra maiúscula; -2, caso a senha não tenha nenhum número; e -3 caso a senha não tenha nenhum caractere especial (!, @, #, $, %, &, (, ou)).

13. Escreva uma função que recebe um número inteiro positivo N e retorna um valor aproximado de e (base do lagarítmo natural), calculado a partir da seguinte equação:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots$$

Sendo N o número de termos da série a serem somados (no exemplo acima, N = 4).

14. Escreva uma função que recebe um número inteiro positivo N e retorna um valor aproximado de π, calculado a partir da seguinte equação:

$$\pi = 3 + \frac{4}{2 \times 3 \times 4} - \frac{4}{4 \times 5 \times 6} + \frac{4}{6 \times 7 \times 8} - \frac{4}{8 \times 9 \times 10} + \dots$$

Sendo N o número de termos da série a serem somados (no exemplo acima, N = 5).

15. Escreva uma função que recebe dois números inteiros, M e N, e uma matriz de tamanho MxN. A função retorna `verdadeiro` se a matriz é uma matriz de permutação, e `falso` caso contrário. Uma matriz NxN é uma matriz de permutação de ordem N se, além de ser uma matriz quadrada, cada linha e coluna da matriz possuir apenas um elemento igual a 1, enquanto todos os outros são iguais a zero. Exemplo:

1	0	0	0
0	0	0	1
0	1	0	0
0	0	1	0

16. Escreva uma função que recebe um vetor e seu tamanho como parâmetro, e retorna o baricentro do vetor. O baricentro de um vetor v de tamanho N é o índice k do vetor que minimiza a soma:

$$S(k) = |\sum_{i=0}^{k} v[i] - \sum_{j=k+1}^{N-1} v[j]|$$

17. Uma matriz A quadrada de dimensão $N \times N$ é um quadrado latino de ordem N se em cada linha e em cada coluna aparecem todos os inteiros $1, 2, 3, 4, \dots N$. Exemplo:

1	2	3	4
2	3	4	1
4	1	2	3
3	4	1	2

Crie uma função que recebe um índice i e verifica se todos os inteiros de 1 a N ocorrem na linha i de A. Em seguida, crie uma função similar para uma coluna j. Utilize estas duas funções para verificar se uma dada matriz é um quadrado latino de ordem N.

18. No Jogo-da-Velha, um jogador vence ao conseguir juntar três casas iguais consecutivas em qualquer direção, dado um tabuleiro 3x3. Crie uma função que recebe como argumento uma matriz de caracteres de tamanho 3x3, preenchida com os caracteres 'X', 'O' ou ' ' (espaço em branco) e retorna `verdadeiro` caso seja possível vencer o jogo na próxima jogada, e `falso` caso contrário. Por exemplo, no tabuleiro a seguir:

O		
X	*X*	*O*
O	*X*	

É possível vencer na próxima jogada, dado que é a vez do jogador 'X' ('X' sempre faz a primeira jogada) e pode-se alinhar três 'X' na vertical.

19. No Jogo da Vida de Conway, cada célula sobrevive ou muda de estado dependendo do seu número de vizinhos. Caso tenha mais de 3 vizinhos vivos, a célula morre de superlotação. Caso tenha menos de 2 vizinhos vivos, a célula morre de solidão. Caso tenha 2 ou 3 vizinhos vivos, a célula permanece viva para a próxima geração, e caso tenha exatamente 3 vizinhos vivos, uma célula morta se torna viva.

Escreva um programa que declara uma matriz global de caracteres, de tamanho 15x15, e inicializa esta matriz na função main com valores quaisquer (asterisco ou espaço em branco em cada posição). Em seguida, crie uma função que recebe dois inteiros, `i` e `j`, e retorna o número de vizinhos vivos da célula na posição `ij` da matriz.