

MC102— Algoritmos e Programação de Computadores

28 de Março de 2012

Lista de Exercícios 02

1. Escreva um programa que leia um número inteiro n fornecido pelo usuário e gere um ‘quadrado’ de n linhas e n colunas que tenha caracteres ‘x’ nas posições da diagonal principal e caracteres ‘+’ nas demais.

ENTRADA: 4**SAÍDA:**

x+++

+x++

++x+

+++x

DICA: O programa deve utilizar dois comandos de repetição aninhados.

2. A série de *Fibonacci* é dada por 1, 1, 2, 3, 5, 8, 13, 21, 34, ..., onde um termo da sequência é dado pela soma dos dois termos anteriores, exceto pelos dois primeiros termos 1 e 1.

$$F(1) = 1$$

$$F(2) = 1$$

$$F(n) = F(n - 1) + F(n - 2)$$

Faça um programa que receba um número n e imprime o n -ésimo termo da sequência de *Fibonacci* na tela.

ENTRADA: 15**SAÍDA:** 610**DICA:** Neste caso a série é 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, ... Como exercício extra, tente imprimir “1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610” na tela.

3. Um número inteiro n é dito subnúmero de um outro número m , se os dígitos de n aparecem na mesma sequência em m . Escreva um programa em C que leia dois números, n e m , e imprima “SIM” se n é um subnúmero de m e “NAO” caso contrário.

ENTRADA: 21 3212**SAÍDA:** SIM

DICA: Neste caso, como 21 aparece em **3212**, 21 é um subnúmero de 3212.

ENTRADA: 21 3231

SAÍDA: NAO

DICA: Como 21 não está contido em 3231, ele não é subnúmero de 3231.

4. Um automóvel possui um medidor que informa de tempos em tempos a um velocímetro digital a distância percorrida (em **metros**) e o tempo gasto (em **segundos**). O velocímetro acumula algumas medições e eventualmente realiza cálculos para atualizar o valor da velocidade (em **km/h**) mostrada no display do automóvel. Desenvolva um programa que:

- Leia uma sequência de pares de valores na forma: distância percorrida<espaço>tempo gasto<enter>;
- Acumule os valores e quando recebe um par de zeros, imprime na saída a velocidade média em **km/h**, zerando os acumuladores;
- O final da leitura se dá quando o programa recebe um par de -1;

Um exemplo de entrada e sua respectiva saída é apresentado a seguir:

ENTRADA: 20 0.5

50 0.5

0 0

20 0.5

120.5 2.0

0 0

-1 -1

SAÍDA: 252.00

202.32

DICA: —

5. Faça um programa que recebe quatro números l , c , i e j e gere um “retângulo” de l linhas e c colunas que tenha caracteres ‘-’ em todas as posições, exceto na posição (i, j) , onde deve haver o valor $i \times j$. Além disso o programa deve imprimir na tela uma mensagem de erro “Entrada incorreta!” se a entrada for inválida. (Caso alguma das dimensões do retângulo seja não-positiva ou caso a posição (i, j) especificada esteja “fora” do quadrado)

ENTRADA: 4 3 2 2

SAÍDA: - - -

- 4 -

- - -

- - -

DICA: A posição (2, 2) contém um 2×2 .

ENTRADA: 5 5 1 6

SAÍDA: Entrada incorreta!

DICA: Não existe a posição (1, 6) num quadrado de dimensões (5, 5).

6. Elabore um programa em C para calcular a raiz quadrada de um número positivo, usando o roteiro abaixo, baseado no método de aproximações sucessivas de Newton. O algoritmo deverá prover 20 aproximações.

Seja Y o número:

- (a) a primeira aproximação para a raiz quadrada de Y é $X_1 = \frac{Y}{2}$;
- (b) as sucessivas aproximações serão $X_{n+1} = \frac{X_n^2 + Y}{2X_n}$.
7. Um número inteiro n é dito ser alternante se a sequência de dígitos que o forma alterna entre dígitos pares e ímpares. Exemplo de números alternantes são: 5, 12, 21, 12345 e 252. Faça um programa que leia um inteiro e imprima se o número é alternante ou não.
8. Supondo que a população de um país A seja da ordem de 90.000.000 de habitantes com uma taxa anual de crescimento de 3% e que a população de um país B seja de, aproximadamente, 200.000.000 habitantes com uma taxa anual de crescimento de 1.5%, escreva um programa que calcule e imprima o número de anos necessários para que a população do país A ultrapasse ou iguale a população do país B, mantidas essas taxas de crescimento.
9. Aponte os erros de implementação existentes no código em C, a seguir, desenvolvido com o intuito de calcular e imprimir o *fatorial* de um número inteiro não-negativo.

```
#include <stdio.h>

int main(){
    int valor;
    scanf("%d", &valor);

    int n = valor;
    float fatorial = 1;

    if (n > -1)
        while (n > 0)
        {
            fatorial *= n;
            n--;
        }
        print("O fatorial de %d e igual a %d\n", valor, fatorial);
    else
        print("Nao existe fatorial de %d\n", n);

    return 0;
}
```

10. Dadas duas seqüências, uma de n e a outra de m números inteiros entre 0 e 9, escrever um programa que calcule a seqüência de números inteiros entre 0 e 9 que represente a soma.

Exemplo:

primeira seqüência:	3	4	5	1	8	0	5
segunda seqüência:			7	3	1	1	8
resultado:	3	5	2	4	9	2	3

11. Dadas duas seqüências de n e m valores inteiros, onde $n \leq m$, escrever um programa que verifique quantas vezes a primeira seqüência ocorre na segunda.

Exemplo:

primeira seqüência: 1 0 1
segunda seqüência: 1 1 0 1 0 1 0 0 1 1 0 1 0

Resultado: 3

12. Escreva um procedimento para um programa que intercala os valores de dois vetores inteiros crescentes de mesmo tamanho em um terceiro vetor, em ordem crescente.

Exemplo:

v1 = [1, 3, 5, 5, 7, 9, 10]
v2 = [2, 2, 4, 6, 8, 8, 10]
v3 = [1, 2, 2, 3, 4, 5, 5, 6, 7, 8, 8, 9, 10, 10]

13. Historicamente César foi o primeiro a codificar mensagens. Ele reorganizava o texto de suas mensagens de maneira que o texto parecia não ter sentido. Cada mensagem sempre possuía uma contagem de letras cujo total equivalia a um quadrado perfeito, dependendo de quanto César tivesse que escrever. Assim, uma mensagem com 16 caracteres usava um quadrado de quatro por quatro; se fossem 25 caracteres, seria cinco por cinco; 100 caracteres requeriam um quadrado de dez por dez, etc. Seus oficiais sabiam que deviam transcrever o texto preenchendo as casas do quadrado sempre que uma mensagem aleatória chegasse. Ao fazerem isso, podiam ler a mensagem na vertical e seu sentido se tornaria claro.

Escreva um programa que lê o tamanho de uma string e a string e escreve a mensagem decifrada. Você deve programar o tratamento de erro caso os arquivos não estejam disponíveis.

Exemplo:

36

MEEUMOCSHMSC1T*AGUOA***L2****T*****A

Esta mensagem pode ser transcrita em um quadrado perfeito 6x6.

M	E	E	U	M	O
C	S	H	M	S	C
1	T	*	A	G	U
0	A	*	*	*	L
2	*	*	*	*	T
	*	*	*	*	A

Lendo cada coluna da matriz (desconsiderando o char '*'), a saída deverá conter:

MC102 ESTA EH UMA MSG OCULTA.

14. Sudoku é jogado numa malha de 9x9 quadrados, dividida em sub-malhas de 3x3 quadrados, chamada "quadrantes". O objetivo do jogo é preencher os quadrados com números entre 1 e 9 de acordo com as seguintes regras:

- Cada número pode aparecer apenas uma vez em cada linha.
- Cada número pode aparecer apenas uma vez em cada coluna.

- Cada número pode aparecer apenas uma vez em cada quadrante.

Exemplo:

9	5	3	4	8	6	2	7	1
1	2	7	9	3	5	8	4	6
6	8	4	7	1	2	9	3	5
5	6	8	3	9	1	4	2	7
4	9	1	2	6	7	3	5	8
3	7	2	8	5	4	1	6	9
7	4	9	5	2	8	6	1	3
2	3	6	1	7	9	5	8	4
8	1	5	6	4	3	7	9	2

Escreva um programa que lê um jogo de Sudoku (matriz 9x9, toda preenchida com números de 1 a 9) e verifica se é um jogo válido ou não. Um jogo válido respeita as três regras acima.