

MC102—Algoritmos e Programação de Computadores

M e N

Segundo semestre de 2009

Lista de exercícios 03

Os exercícios marcados com uma estrela (★) são aqueles que podem ser cobrados no teste do dia 3 de Novembro de 2009

1. Escreva um programa em C que solicita ao usuário duas cadeias de caracteres e informa se elas são iguais ou diferentes. Como a sua função pode retornar esta informação usando um inteiro?
2. ★ Escreva uma função que leia um número inteiro positivo e escreva **na tela** esse mesmo número utilizando algarismos romanos. Você pode escrever um número em algarismos romanos da seguinte forma: pegue cada algarismo decimal (centena, dezena e unidade) e converta para o formato romano e depois escreva na mesma ordem em que os algarismos estavam no número original.

Ex: 579 pode ser decomposto em 5 (centena), 7 (dezena) e 9 (unidade). A centena 5 é escrita como D, a dezena 7 é escrita como LXX e a unidade 9 é escrita como IX. Logo, 579 é escrito como DLXXIX.

Você pode utilizar, em seu programa, as seguintes funções:

- `char *centena(int algarismo)` — dado um algarismo que represente a centena de um número decimal, retorna uma *string* que representa aquela centena em algarismos romanos.
- `char *dezena(int algarismo)` — dado um algarismo que represente a dezena de um número decimal, retorna uma *string* que representa aquela dezena em algarismos romanos.
- `char *unidade(int algarismo)` — dado um algarismo que represente a unidade de um número decimal, retorna uma *string* que representa aquela unidade em algarismos romanos.

3. Escreva uma função em C que recebe como parâmetro uma cadeia e um caracter, e retorna como resultado o número de ocorrências desse caracter na cadeia passada como parâmetro. O cabeçalho da função deve ser `int conta(char s[], char c);`
4. ★ Escreva a função `int fim_cadeia(char s[], char t[])`, que retorna 1 se a cadeia *t* ocorrer no fim da cadeia *s*, e 0 caso contrário.
5. Determine a saída dos programas abaixo para as entradas fornecidas. **Dica:** a saída é sempre uma palavra em português.

(a) A entrada é 10 25 7 7 25 19 21 13

```

#include <stdio.h>
main () {
    char s[9];
    int i, v;
    s[8] = '\0';
    for (i = 0; i < 8; i++) {
        scanf ("%d", &v);
        s[i] = 'z' - v;
    }
    printf ("%s\n", s);
}

```

(b) A entrada é rvoatii

```

#include <stdio.h>
#include <string.h>
main () {
    int i = 4, j = 0;
    char a[8], b[8];
    scanf("%s", a);
    b[7] = '\0';
    do {
        b[i] = a[j];
        i += 3;
        i %= 7;
        j++;
    } while (i != 4);
    printf ("%s\n", b);
}

```

6. ★ Um grande político romano decidiu cobrar por seus discursos, famosos em todo o mundo antigo pela sua beleza. Para contabilizar o que lhe era devido, ele colocou um preço de meia moeda de bronze por letra. Imediatamente todos reclamaram que o preço era muito alto e que ninguém o convidaria mais para discursar. Para agradar a sua clientela, nosso político resolveu adotar uma política de preços mais realista: ele cobraria uma moeda de ouro por vogal, limitado ao maior número de vogais *consecutivas* presentes no discurso. Isso agradou a todos, até o momento que começou o primeiro discurso sob a nova regra: poucos sabiam que era possível colocar tantas vogais juntas em latim...

Você faz parte da equipe de um museu que trabalha para contabilizar quanto dinheiro este político ganhou com seus discursos. Como parte deste trabalho, escreva uma função que receba como parâmetro uma cadeia e retorne o maior número de vogais consecutivas presentes nesta cadeia. Por exemplo, para a cadeia "Alea jacta est." a sua função deve retornar 2 como resposta. O cabeçalho da sua função deve ser:

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

Como o político falava em latim, você deve utilizar a função `vogal` definida abaixo (e implementada por outro membro da equipe do museu) para determinar se uma caracter é uma vogal:

```
int vogal(char c) — Retorna 1 se c for uma vogal, ou 0 caso contrário.
```

7. Escreva uma função que, dado uma cadeia de caracteres, retorne 1 se a cadeia for um palíndromo ou 0 caso contrário. Uma cadeia é um palíndromo se, quando lida ao contrário, ela possui exatamente o mesmo conteúdo. Considere que a cadeia só contém letras minúsculas sem acentos e números (ou seja, não contém espaços, vírgulas e outros símbolos que podem complicar o problema). O cabeçalho da função é `int palindromo(char cadeia[])`;

8. Transforme o algoritmo de busca sequencial visto em sala de aula em uma função de busca sequencial, definindo uma declaração de função apropriada. Qual devem ser os parâmetros da função? Qual o valor de retorno? Como essa função pode ser usada?
9. Transforme o algoritmo de ordenação por seleção visto em sala de aula em uma função de ordenação por seleção, definindo uma declaração de função apropriada. Qual devem ser os parâmetros da função? Qual o valor de retorno? Como essa função pode ser usada?
10. ★ Dado o vetor ordenado $v = \{1, 5, 6, 7, 12, 17, 22, 45, 65, 98, 100, 120, 150, 200, 231\}$. Mostre passo a passo a busca binária pelo elemento 100. Quantas comparações foram feitas? E se a busca fosse sequencial, quantas comparações seriam feitas?
11. Para o vetor acima, dê um exemplo onde a busca sequencial faria menos comparações que a busca binária.
12. ★ Ordene o vetor $v = \{20, 12, 28, 05, 10, 18\}$ usando os métodos de inserção, seleção e bolha. Mostre o vetor a cada passo do laço para cada um dos métodos.
13. Escreva um programa que leia um vetor de n valores (n informado pelo usuário) e determine o k -ésimo menor elemento do vetor (k também informado pelo usuário). Considere que o usuário nunca digitará n e k maiores que 100. **Dica:** ordene o vetor antes.
14. Escreva uma função recursiva que exiba, em ordem crescente, os números do intervalo $[1, n]$ na saída padrão.
15. Escreva uma função recursiva que exiba, em ordem decrescente, os números do intervalo $[1, n]$ na saída padrão.
16. ★ Os romanos descobriram uma forma interessante de realizar a potenciação de um número real por uma potência inteira.

$$x^n = \begin{cases} x^{\frac{n}{2}} \cdot x^{\frac{n}{2}} & \text{se } n \text{ for par e } n > 0; \\ x \cdot x^{\frac{n-1}{2}} \cdot x^{\frac{n-1}{2}} & \text{se } n \text{ for ímpar e } n > 0; \\ 1 & \text{se } n = 0; \\ \frac{1}{x^{-n}} & \text{se } n < 0; \end{cases}$$

Usando esta definição, você deve escrever a função **potencia** que calcula a potenciação de um número real por uma potência inteira. A sua função deve ter a assinatura abaixo e deve ser implementada de forma *recursiva*.

```
float potencia(double x, int n){
```

17. Escreva um procedimento recursivo que recebe uma string como parâmetro e a exibe invertida.
18. ★ Escreva uma função recursiva `int mdc(int x, int y)` para calcular o máximo divisor comum de dois números x e y ($x \geq y$). Lembre-se que o MDC é calculado da seguinte forma:

$$\text{MDC}(x, y) = \begin{cases} y & \text{caso } x \bmod y = 0 \\ \text{MDC}(y, x \bmod y) & \text{caso contrário} \end{cases}$$

19. ★ Considere a função abaixo e relacione a primeira coluna (com os parâmetros da função) com a segunda coluna (com os possíveis valores de retorno).

- | | |
|---------------------------|-------|
| 1. a = 12, b = 2, c = 100 | A. 44 |
| 2. a = 12, b = 3, c = 100 | B. 28 |
| 3. a = 12, b = 4, c = 100 | C. 36 |
| 4. a = 12, b = 5, c = 100 | D. 32 |
| 5. a = 11, b = 2, c = 100 | E. 21 |

```

int recursao (int a, int b, int c) {
    int x;
    if (b == 0) {return (1);}
    else if (b % 2 == 0) {
        x = recursao (a, b/2, c);
        x = (x * x) % c;
    } else {
        x = recursao (a, b/2, c);
        x = (x * x * a) % c;
    }
    return (x);
}

```

20. ★ Escreva uma implementação para a função $f(x)$ inteira, definida para valores de x inteiros. Sua implementação deve *obrigatoriamente* utilizar recursão. Você deve definir os parâmetros e o valor de retorno adequados para a sua função.

$$f(x) = \begin{cases} 0 & \text{se } x \leq 1; \\ f(3x + 1) + 1 & \text{se } x \text{ for ímpar e } x > 1; \\ f(x/2) + 1 & \text{se } x \text{ for par e } x > 1. \end{cases}$$

21. Escreva uma função recursiva que determine o valor máximo de um vetor de inteiros positivos.
22. ★ A sequência de Fibonacci é tal que cada elemento (com exceção dos dois primeiros que são 0 e 1), é igual a soma dos dois anteriores. Ex: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, . . .
A definição da função para o n-ésimo número de fibonacci é:

$$\text{Fibonacci}(n) = \begin{cases} 0 & \text{se } n = 0 \\ 1 & \text{se } n = 1 \\ \text{Fibonacci}(n - 1) + \text{Fibonacci}(n - 2) & \text{se } n > 1 \end{cases}$$

Escreva uma função recursiva para determinar a sequência de Fibonacci de um número n dado pelo usuário.