

Arquitetura de Computadores 2009/10

Aula prática 2 – Tipos de dados em C

1. Implemente uma função que apresente no ecrã os códigos ASCII de todos os caracteres de uma *string* dada como parâmetro.

Uma vez implementada a função, faça o programa principal que lê a *string* e apresenta os códigos ASCII, utilizando a função previamente desenvolvida.

Compile e teste o seu programa.

2. Implemente uma função que inverte uma *string*.

Dada a *string* (vector de caracteres), que a função recebe como parâmetro, esta deve proceder à inversão da posição de todos os caracteres até ao terminador '`\0`'. Note que o vector original será alterado automaticamente.

Uma vez implementada a função, faça o programa que lê uma *string* e a apresenta invertida no ecrã.

Compile e teste o seu programa

3. Considere as seguintes estruturas e vectores:

```
struct a {
    char s1[5];
    int i;
    char s2[5];
};

struct b {
    int i;
    char s1[5];
    char s2[5];
};

struct a va[10];
struct b vb[10];
```

Determine, através de um pequeno programa em que utiliza as funções `printf` e `sizeof`, o número de bytes que ocupa cada uma das estruturas e os vectores definidos.

Analise os resultados obtidos.

Arquitetura de Computadores 2009/10

Aula prática 2 – Tipos de dados em C

4. Construa uma expressão, utilizando **sizeof**, que dê o número de elementos de um dado vector **V**.
 - a. Assuma que os elementos do vector são do tipo **T**;
 - b. Modifique a expressão para suportar vectores com qualquer tipo de elementos.
5. Faça um programa que lê um número em base **B** (uma *string*) e apresenta no ecrã o seu valor em decimal. Desenvolva o seu programa de forma incremental, seguindo os passos, aqui enumerados:
 1. considera **B** = 2 (binário);
 2. considera **B** ≤ 10, valor introduzido pelo utilizador;
 3. considera **B** ≤ 16, valor introduzido pelo utilizador;
 4. complete o seu programa verificando se a *string* introduzida é realmente um número na base **B**.

Não se esqueça que deve estruturar sempre o seu programa utilizando funções.

6. Faça o programa para decifrar a cifra de César com N=3. Decifre a seguinte frase — Vdprv surjudpdu hp C!
7. Faça uma função que dado um vector de inteiros e o seu número de elementos, retorna o menor elemento do vector.
Teste a sua função com um programa completo.
8. Considere que uma cadeia de caracteres representa um número em binário, apenas contendo '0's e '1's. Programe a função

`and(char n1[], char n2[], char n3[])`

que preenche `n3` com o “e” lógico dos números representados por `n1` e `n2`. Por exemplo se `n1` contiver "01" e `n2` contiver "11" `n3` deve ficar com "01".

Numa primeira fase considere `n1` e `n2` do mesmo tamanho, depois considere que podem ter comprimentos diferentes, devendo neste caso considerarem-se alinhados à direita. Assuma sempre que `n3` tem espaço suficiente para guardar o resultado.

Teste a sua função com um programa completo.