

Manipulação duma lista telefónica

- Dados:
 - nome: variável indexada de caracteres ("string");
 - telefone: inteiro ou melhor variável indexada de caracteres que representam os dígitos (porquê?)

- Representada por duas variáveis indexadas: char nome[MAX][MAXN], tel[MAX][MAXT];
- Um elemento é nome[i] e tel[i]
- Mas melhor, é agrupar cada elemento numa estrutura:

```
struct elemento{  
    char nome[MAXN];  
    char tel[MAXT];  
}  
lista[MAX];
```

```
strcpy(lista[0].nome, "Ricardo Reis");  
strcpy(lista[0].tel, "0930191912");  
i=6;  
printf("Nome:%s, Tel:%s, lista[i].nome, lista[i].tel);
```

Funcionalidades

- Consultar a lista telefónica: dado o nome encontrar o telefone.
- Função de leitura de dados.
- Modificar a função anterior de modo a que os elementos na lista fiquem ordenados por ordem crescente de nomes.
- Modificar a função procura para usar pesquisa binária.
- Guardar a lista de telefones num ficheiro.

Leitura de dados

- Solução I: Usar a função `scanf()` para ler os dados.
- Desvantagem: não se pode ter espaços entre os nomes
- Solução II: Supor que o ficheiro de dados tem um formato determinado. Por exemplo:
- cada campo dum registo é separado por uma vírgula
- cada registo é separado pelo caracter mudança de linha

joaquina afonso, 016002222

ana catarina, 0671445566

joao coelho, 8442222

- Divisão do problema em funções
- char nao _ch(char ch)
 - lê caracteres iguais a ch e retorna o primeiro diferente.
- campo(char ch,char t,char a[])
 - em a[] fica a sequência de caracteres que começa em ch e termina em t.

Acesso a ficheiros

- `FILE *fopen(char nome_de_ficheiro[],char modo[])`: onde, `nome_de_ficheiro` é uma sequência de caracteres (string) e `modo` pode ser: "r", "w", "a", etc.
- `int fclose(FILE *f)` retorna EOF se ocorreu algum erro senão 0.

- `int getc(FILE *f)` retorna o caracter seguinte no ficheiro. (`getchar()` é equivalente a `getc(stdin)`)
- `int putc(int c,FILE *f)` escreve o caracter `c` no ficheiro, na posição seguinte. (`putchar(c)` é equivalente a `putc(c,stdin)`)
- Leitura de strings: `fgets(char s[],int n, FILE *f)`
- Escrita de strings: `fputs(char s[], FILE *f)`
- Leitura formatada: `fscanf(FILE *f,char formato[],...)`
- Escrita formatada: `fprintf(FILE *f,char formato[],...)`

Passagem de argumentos para um programa em C

- Ao `main()` são passados dois argumentos:
- `argc`: número de argumentos passados na linha de comandos
- `argv`: variável indexada de strings que contém os argumentos com que foi chamado o comando. `argv[0]` é sempre o nome do próprio comando.

Interface para a manipulação dum lista telefónica.

- Menu de opções:
 - Introduzir novo elemento
 - Procurar por nome
 - Procurar por telefone
 - Retirar elemento
 - Lista de elementos
 - Terminar