

## Programação 2024/2025 – 2º Semestre

### Aula de Laboratório 6

Ao desenvolver os seguintes programas tenha em atenção o bom uso dos comentários, de indentação e o correto nome das variáveis!!!! Bom trabalho!

Considere um ficheiro de dados de texto que contém os resultados de uma prova de atletismo. Cada linha do ficheiro representa um participante e contém a seguinte informação:

- Posição: Posição geral em que acabou a prova.
- Escalão: Caracter seguido de um número. O caracter pode ser M (masculino) ou F (feminino) e o número corresponde à idade mínima do escalão.
- Posição no escalão: Posição dentro do seu escalão quando acabou a prova.
- Dorsal: Número do dorsal. Atribuído durante a inscrição
- Nome: Nome do participante.
- Sexo: M (masculino) ou F (feminino)
- Tempo: Tempo que demorou a completar a prova.

O ficheiro *RunResults.txt* fornecido no FENIX tem os resultados da prova de atletismo. Apenas são fornecidos os resultados para os 140 participantes que acabaram a prova com o menor tempo. Os campos do ficheiro encontram-se delimitados por um tab (`'\t'`).

1. Leia o ficheiro *RunResults.txt* para uma lista simplesmente ligada de estruturas que guardam a informação de cada participante lida de uma linha do ficheiro.
  - A. Defina uma estrutura para um elemento da lista. Deve guardar os sete diferentes tipos de informação guardada no ficheiro em diferentes campos da estrutura. Utilize alocação dinâmica de memória para o nome de cada participante, isto é, o nome deve ocupar apenas o espaço necessário para o representar.
  - B. Deve carregar todo o ficheiro para a memória para a lista. Ignore a primeira linha.
2. O programa deve permitir fazer pesquisas na lista em memória consoante os parâmetros da linha de comando. O resultado da pesquisa deve ser apresentado no écran. Os parâmetros da linha de comando podem aparecer por qualquer ordem e devem ser (alguns aparecem aos pares):
  - h mostra os parâmetros possíveis e as correspondentes funcionalidades
  - i ficheiro usar ficheiro de entrada “ficheiro” em vez de “RunResults.txt”, por omissão
  - e escalão pesquisar o “escalão”, mostrando a informação dos participantes desse escalão
  - n nome pesquisar o “nome” do participante, mostrando a informação dos participantes que tenham esse “nome” ou partes desse “nome”. Note que, se quiser pesquisar um nome e apelido, deve pôr o nome dentro de aspas, para que o programa receba apenas uma única string (sem as aspas).

A. Exemplos de comandos válidos:

```
./programa -e F20
```

```
./programa -i RunResults_out.txt -e M35
```

```
./programa -n "Bruno Carvalho"
```

B. Sugere-se a utilização de `getopt()` para processar os parâmetros da linha de comando. Confira a página do manual com “`man 3 getopt`”.

C. Faça funções para os diferentes tipos de pesquisa na lista. Não se esqueça de libertar a memória no fim.

D. Basta fazer uma pesquisa por invocação do programa, sendo opção do programador o que acontece se o utilizador puser múltiplos comandos semelhantes na linha de comandos: pode processar só o último, por exemplo. Ou seja, o comando seguinte poderia pesquisar apenas os participantes do escalão F20 no ficheiro `RunResults2.txt`:

```
./programa -i RunResults_out.txt -e M35 -i RunResults2.txt -e F20
```

E. Verifique o correcto funcionamento do programa com o *valgrind*.