

Programação 2024/2025 – 2º Semestre**Aula de Laboratório 5**

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* representando todos os dados do ficheiro como um vector de ponteiros `char *linhas[]`. Cada linha do ficheiro deve ser representada por uma *string*. O objectivo é ordenar os resultados por escalão, segundo a seguinte ordem: F20, F40, F45, M20, M35, M40, M45, M50, M55, M60 e escrevendo outro ficheiro com o nome *RunResults_out.txt* com os dados ordenados. Considere o seguinte:
 - A. Utilize alocação dinâmica de memória para cada uma das linhas, isto é, cada linha deve ocupar apenas o espaço necessário para a representar. Pode assumir que o vector `linhas` tem o tamanho de 140. Ignore a primeira linha.
 - B. Deve carregar todo o ficheiro para a memória, criar outro vector `char *linha_ord[]` com as linhas ordenadas e de seguida, escrever o ficheiro de saída. Cada uma destas tarefas deve corresponder a uma função e note que não é necessário alocar memória para cada linha do vector `linha_ord`.
 - C. Verifique o correcto funcionamento do programa com o *valgrind*.

2. Pretende-se saber em que posição ficou um determinado participante e para isso o programa deve receber o ficheiro com os resultados da prova no formato acima descrito e um nome. O programa deve imprimir a posição geral do participante bem como a posição no escalão a que pertence.
- A. Os dois parâmetros são introduzidos como argumentos de execução do programa. Deve recorrer à seguinte definição da função main:

```
int main(int argc, char *argv[])
```

argc é o número de argumentos (o nome do programa também conta como argumento), argv[0] aponta para o nome do programa e as restantes posições para cada um dos argumentos.

- B. Utilize a função de carregar o ficheiro para a memória, escrita na questão anterior. Sugere-se o uso das funções *strstr* ou *strcmp* ou *sscanf* ou *strtok* para realizar a procura e a comparação de *strings*.
- C. Verifique o correcto funcionamento do programa com o *valgrind*.