

## Miniteste 1 – Exemplo

1. Um sistema de gestão de dados sobre cinema está em desenvolvimento e é pedido nesta fase que se implementem algumas funções. Este sistema utiliza a biblioteca de vetores e a biblioteca de listas, que se encontram disponíveis nos ficheiros `vetor.h`, `vetor.c`, `lista.h` e `lista.c`. As funções devem ser implementadas no ficheiro `prob1.c` fornecido.

1.1 Implemente uma função permita retirar os elementos duplicados de um vetor.

```
void retira_duplicados(vetor *vec)
```

Depois de implementada a função, o programa deverá apresentar:

```
Total inicial: 146
Unicos: 30 (esperado: 30)
```

1.2 Implemente uma função que, dadas duas listas, retorne um vetor com os elementos comuns às duas listas:

```
vetor* interseta(lista *l1, lista *l2)
```

Por exemplo, dadas as listas {"prog1", "prog3", "prog2"} e {"prog3", "prog5", "prog4"} pretende-se que o resultado seja o vetor {"prog3"}. No caso de não haver elementos comuns, a função deve retornar NULL.

Depois de implementada a função, o programa deverá apresentar:

```
Interseccao: 75 (esperado: 75)
```

1.3 Implemente uma função que, dada uma lista de filmes, retorna um vetor com os filmes anteriores a um ano indicado:

```
vetor* anteriores(lista *lst, const int ano)
```

Depois de implementada a função, o programa deverá apresentar:

```
Anteriores: 106 (esperado: 106)
```

2. Pretende-se criar um programa para contabilizar o número de ocorrências de cada palavra distinta encontrada num ficheiro de entrada e apresentar um relatório. Considera-se como palavra qualquer sequência de caracteres não brancos, com um comprimento máximo de 100 caracteres. As funções devem ser implementadas no ficheiro `prob2.c` fornecido.

2.1 Implementa a função `analisaFicheiro` que lê um determinado ficheiro e devolve um vetor de registos e o número de palavras encontradas.

```
registro* analisaFicheiro(FILE *ficheiro, int *num_palavras)
```

O vetor deverá ser alocado dinamicamente e conter um registo para cada palavra distinta encontrada. O número de palavras distintas encontradas deverá ser indicado por `num_palavras`. Não é necessário qualquer tipo de ordenação.

**2.2** Implementa a função `gerarRelatorio` que escreve para um ficheiro de saída um relatório dos resultados.

```
void gerarRelatorio(FILE *ficheiro, registo *resultados, int num_palavras)
```

A função deve escrever no ficheiro um cabeçalho e uma listagem (sem ordem específica) das diferentes palavras encontradas e o respectivo número de ocorrências (vetor `resultados`). O tamanho do vetor `resultados` é indicado pelo argumento `num_palavras`. Não é necessário qualquer tipo de ordenação.

Para o ficheiro de entrada ***dialogo.txt***, o ficheiro de saída ***relatório.txt*** deverá apresentar o seguinte resultado:

Palavra encontradas:

```
-----  
1:  Maria: (2)  
2:  Eu (3)  
3:  adoro (2)  
4:  Prog2. (3)  
5:  E (1)  
6:  tu? (1)  
7:  Manel: (2)  
8:  +/-... (1)  
9:  Prog2 (1)  
10: porque (1)  
11: eu (1)  
12: aprendi (2)  
13: muito (2)  
14: com (2)  
15: tambem (1)
```