

**No final deverás confirmar que submetes corretamente no SIGEX o código fonte dos teus programas utilizando o nome indicado no enunciado.
Quaisquer cópias detetadas serão penalizadas com anulação da prova.**

1. Integras a equipa que está a desenvolver um sistema de gestão de informação geográfica e és responsável por implementar algumas funções. Este sistema utiliza a biblioteca de listas ligadas e a biblioteca de vetores estudadas nas aulas práticas, que se encontram disponíveis nos ficheiros `vetor.h/.c`, e `lista.h/.c`. As funções devem ser implementadas no ficheiro **prob1.c** fornecido.

1.1 [20 pontos] Implementa a função `comprimento_medio` que permite obter o comprimento médio das *strings* que estão guardadas numa lista. Quando ocorrer um erro ou quando o tamanho da lista de entrada for 0, o retorno deverá ser 0.

```
float comprimento_medio(lista *lst)
```

Indica ainda num comentário no início do código da função qual a complexidade do algoritmo que implementaste.

Depois de implementada a função, o programa deverá apresentar:

```
Comprimento medio das palavras: 10.7
```

1.2 [20 pontos] Implementa a função `filtra_nomes` que permite obter uma nova lista com os nomes que contenham a *string* definida pelo parâmetro `existe`.

```
lista* filtra_nomes(lista *lst, char *existe)
```

Adicionalmente, esses nomes devem ser removidos da lista de entrada `lst`. Quando ocorrer um erro o retorno deverá ser `NULL`. Sugestão: utilize a função `strstr` da biblioteca `string.h`.

Depois de implementada a função, o programa deverá apresentar:

```
Nomes filtrados (total de 11):
Sao Mamede de Ribatua
Sao Mamede de Infesta
...
Azinh. Barros/S Mamede Sadao
```

1.3 [15 pontos] Implementa a função `mais_palavras` que permite obter a *string* com mais palavras no nome entre as *strings* que estão guardadas num vetor. Quando ocorrer um erro ou quando o tamanho da lista de entrada for 0, o retorno deverá ser `NULL`.

```
const char* mais_palavras(vetor *vec)
```

Depois de implementada a função, o programa deverá apresentar:

```
Nome com mais palavras: N Sra da Graca Pavoia e Meada
```

***** Submeta o ficheiro que desenvolveu no SIGEX com o nome prob1.c *****

2. Estás a desenvolver um programa para analisar informação sobre os municípios de Portugal. Essa informação é armazenada em registos com os seguintes campos: nome (1 palavra), população em 2001 (inteiro) e população em 2012 (inteiro). As funções devem ser implementadas no ficheiro **prob2.c** fornecido.

2.1 [15 pontos] Implementa a função `lerFicheiro` que lê um determinado ficheiro e devolve um vetor de municípios e o número de municípios existentes. Trata-se de um ficheiro de texto, cada linha contendo o nome do município e as respetivas populações.

```
municipio* lerFicheiro(FILE *ficheiro, int *num_municipios)
```

O vetor deverá ser alocado dinamicamente e conter um registo por cada município existente no ficheiro. O número de municípios deverá ser indicado em `num_municipios`.

2.2 [15 pontos] Implementa a função `gerarRelatorio` que escreve para um ficheiro de saída um relatório dos resultados.

```
void gerarRelatorio(FILE *ficheiro, municipio *resultados,  
int num_municipios)
```

A função deve escrever no ficheiro a lista dos dez municípios que perderam mais população entre 2001 e 2012, entre os municípios guardados no vetor `resultados`. O tamanho do vetor `resultados` é indicado pelo argumento `num_municipios`. A listagem deve incluir o nome do município e a população perdida.

Para o ficheiro de entrada **municipios.txt**, o ficheiro de saída **relatorio.txt** deverá apresentar o seguinte resultado:

```
Lisboa, 32465  
Porto, 31715  
Coimbra, 8004  
Seia, 3720  
Abrantes, 3633  
Amarante, 3619  
Covilha, 3548  
Tomar, 2978  
Valpacos, 2862  
Fundao, 2649
```

***** Submeta o ficheiro que desenvolveu no SIGEX com o nome prob2.c *****