

**No final deverá 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 contactos e és responsável por implementar algumas funções. Este sistema utiliza as bibliotecas de listas ligadas e de filas estudadas nas aulas práticas, que se encontram disponíveis nos ficheiros `lista.h/.c`, e `fila.h/.c`. As funções devem ser implementadas no ficheiro **prob1.c** fornecido.

1.1 [25 pontos] Implementa a função `junta_nomes`, que agrega numa nova lista todos os nomes contidos em duas listas (que não devem ser alteradas). A nova lista fica portanto com comprimento igual à soma dos comprimentos das listas de entrada.

```
lista* junta_nomes(lista *lst1, lista *lst2)
```

Quando ocorrer um erro o retorno deverá ser `NULL`.

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:

```
Lista resultante contem 354 nomes.
```

1.2 [20 pontos] Implementa a função `lista_remove_duplicados`, que remove todos os duplicados existentes na lista e os coloca numa nova lista. Quando ocorrer um erro ou quando o tamanho da lista de entrada for 0, o retorno deverá ser `NULL`.

```
lista* lista_remove_duplicados(lista *lst)
```

Depois de implementada a função, o programa deverá apresentar:

```
Nomes duplicados (total de 50):  
ACÁCIA MARIA CARDOSO SANTOS  
ACI TERESINHA FERRAZ  
...  
ZENARA CAMPOS DOS SANTOS
```

1.3 [20 pontos] Implementa a função `pesquisa_nome`, que permite obter os nomes com correspondência com o nome indicado (*string* definida pelo parâmetro `nome`) e os guarda numa fila.

Quando ocorrer um erro ou quando o tamanho da lista de entrada for 0, o retorno deverá ser `NULL`.

```
fila* pesquisa_nome(lista *lst, char *nome)
```

Depois de implementada a função, o programa deverá apresentar (para o nome "SILVA"):

```
Nomes encontrados (41):  
ADAILTON MARTINS SILVA  
ADENILSA JOANA DA SILVA AGUIAR  
..  
WELLINGTON DA COSTA SILVA
```

***** Submete o ficheiro prob1.c desenvolvido no SIGEX *****

2. [15 pontos] Estás a desenvolver um programa para manipular expressões matemáticas. Este programa utiliza as bibliotecas de vetores e de pilhas estudadas nas aulas práticas, que se encontram disponíveis nos ficheiros `vetor.h/.c`, e `pilha.h/.c`.

Implementa no ficheiro **prob2.c** fornecido a função `avalia_expressoes`:

```
int avalia_expressoes(FILE *ficheiro, vetor *expressoes)
```

A função deve escrever no ficheiro que recebe como parâmetro: as expressões guardadas no vetor e se estão sintaticamente corretas, no que diz respeito ao número de parênteses. O resultado da avaliação deverá ser "OK" ou "NOK" (expressão incorreta); a função deve retornar 1 se for bem sucedida ou 0 em caso contrário.

Sugestão: utiliza uma pilha para verificar se o número de parênteses está balanceado, colocando um valor na pilha sempre que um parêntesis é aberto e retirando sempre que um parêntesis é fechado.

Depois de implementada a função, o ficheiro de saída **avaliacao.txt** deverá apresentar o seguinte resultado:

```
(3 + 4) + 7 * 2 - 1 - 9 -> OK  
(8 - 1 + 3) * 6 - ((3 + 7) * 2) -> NOK  
3 * 6 - 7 + 2 -> OK  
(3 + 4)) + 7 * 2 - 1 - 9 -> NOK  
5 - 2 + 4 * (8 - (5 + 1)) + 9 -> OK  
6 * 2 + (5 - 3) * 3 - 8 -> OK  
5 - 2 + 4 * (8 - (5 + 1)) + 9 -> OK  
(8 - 1 + 3) * 6 - ((3 + 7) * 2) -> OK  
3 * 6 - 7 + 2 -> OK  
6 * 2 + (5 - 3) * 3 - 8 -> OK  
4 + 2 -> OK  
4 + 2) -> NOK
```

***** Submete o ficheiro prob2.c desenvolvido no SIGEX *****