

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

1 [8 valores] Tendo por base a biblioteca de pilhas estudadas nas aulas práticas, e disponíveis nos ficheiros `pilha.h/.c`, implemente as funcionalidades pedidas nas duas alíneas seguintes no ficheiro **prob1.c** fornecido. Sempre que conveniente utilize as funções disponíveis.

Pretende-se criar um programa para analisar os cromos de uma coleção. A lista de cromos está guardada num vetor de inteiros alocado dinamicamente. Os cromos são representados por números. Cada posição do vetor tem o número do cromo e estes não estão ordenados.

1.1 [4 valores] Implemente a função

```
int* cromos_em_falta(int *cromos, int n, int ntotal, int *tam)
```

Esta função recebe um vetor contendo um conjunto de cromos e retorna um vetor alocado dinamicamente com os cromos que faltam na coleção ou NULL em caso de erro. Os restantes parâmetros da função têm a seguinte utilização: `n` é o número total de cromos do vetor `cromos`; `ntotal` é o número total da coleção; e `tam` é um apontador para o tamanho do vetor retornado. Os cromos em falta devem estar ordenados por ordem crescente.

Dado [2,4,5,6,5,4,10,11,10,10,3,12,14,16,18], de uma coleção com o número total de 20 cromos, o resultado será [1,7,8,9,13,15,17,19,20]. Depois de implementada a função, o programa deverá apresentar:

Número de cromos em falta: 9 (Certo)

Lista correta dos cromos em falta: [1,7,8,9,13,15,17,19,20]

1.2 [4 valores] Imagine que tem uma coleção armazenada numa pilha e pretende retirar da pilha os cromos de um dado intervalo de números. Implemente a função

```
void retirar_cromos_intervalo(pilha *cromos, int inicio, int fim)
```

A função recebe a pilha `cromos`, e os valores `inicio` e `fim`, que representam os valores de início e de fim do intervalo de cromos a retirar (o valor `inicio` e o valor `fim` também se retiram). A função deve retornar a pilha sem os cromos a retirar e os cromos sobrantes devem estar na mesma ordem. Deve apenas recorrer às funções da pilha (não serão consideradas soluções em que os elementos da pilha são percorridos diretamente).

Sugestão: utilize uma pilha auxiliar para manter a ordem.

Prova com consulta (Moodle). Duração: 70min.

Parte prática [15/20 valores]

2
5
6
11
34
25
35
5
24

Dado com intervalo entre 10 e 25 o resultado será

2
5
6
34
35
5

Os cromos foram retirados corretamente (Certo)

Cromos: [2,5,6,34,35,5]

**** Submeta o ficheiro prob1.c no SIGEX ****

2 [7 valores] Está a desenvolver um programa para gestão de correspondência numa estação de correios. Este programa utiliza as bibliotecas de vetores e listas ligadas estudadas nas aulas práticas, que se encontram disponíveis no ficheiro `vetor.h/.c` e `lista.h/.c`. As bibliotecas foram adaptadas para guardar os dados necessários para este programa. Os dados constantes do ficheiro `correspondencia.txt` são carregados em três vetores: um contendo os nomes dos remetentes, outro contendo os nomes dos destinatários e um terceiro contendo os códigos postais do endereço de destino.

2.1 [4 valores] Implemente no ficheiro **prob2.c** fornecido a função `pesquisar_destinatarios`:

```
vetor* pesquisar_destinatarios(vetor *vrem, vetor *vdest, const char  
                               *remetente)
```

A função aceita como argumentos dois vetores com informação relativa ao nome dos remetentes da correspondência (`vrem`), e ao nome dos destinatários (`vdest`). O terceiro argumento (`remetente`) é o nome do remetente em relação ao qual se pretende efetuar uma pesquisa. A função retorna um vetor ordenado e sem elementos repetidos dos nomes dos destinatários para quem o remetente enviou correspondência. Em caso de erro ou argumentos inválidos, a função deve retornar `NULL`.

Depois de implementada a função, o programa deverá apresentar:

Para o remetente Carolina Costa...

Tamanho do vetor OK: 4

Conteúdo do vetor de destinatários:

Bianca Esteves

Camila Anjos

Diogo Abreu

Goncalo Assuncao

2.2 [3 valores] Implemente no ficheiro **prob2.c** fornecido a função `contar_correspondencia`:

```
lista* contar_correspondencia(vetor *vcp_dest, int *cpdistintos)
```

A função aceita como argumento um vetor com os códigos postais dos destinatários da correspondência depositada na estação (`vcp_dest`). A função efetua a contagem do número de cartas existentes para os diferentes destinos (código postal do destinatário) e coloca essa informação na lista ligada a devolver. A função deve ainda preencher o número de códigos postais distintos que foram encontrados, usando para o efeito o parâmetro `cpdistintos`. Em caso de erro ou argumentos inválidos, a função deve retornar `NULL`.

Prova com consulta (Moodle). Duração: 70min.

Parte prática [15/20 valores]

Depois de implementada a função, o programa deverá apresentar, não necessariamente pela mesma ordem:

Tamanho da lista OK: 11

Numero de codigos postais distintos retornado OK: 11

2440-000 BATALHA : 2 cartas
4411-801 ARCOZELO VNG : 7 cartas
2440-445 BATALHA : 2 cartas
2495-013 S. MAMEDE : 2 cartas
4150-740 PORTO : 7 cartas
4415-350 PEDROSO : 5 cartas
5400-582 CHAVES : 7 cartas
4615-019 FREIXO DE CIMA : 7 cartas
2125-998 MUGE : 7 cartas
7940-311 FARO DO ALENTEJO : 6 cartas
4935-204 VIANA DO CASTELO : 9 cartas

**** Submeta o ficheiro prob2.c no SIGEX ****