

**No final deverá confirmar que submete 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. Pretende-se implementar um sistema para registo/atualização de informação relativa a ruas da cidade do Porto e seus estabelecimentos comerciais. Este sistema utiliza as bibliotecas de **filas de prioridades** (fila de prioridade de mínimo, baseadas em heap) e **tabelas de dispersão** (encadeamento fechado) estudadas nas aulas, que se encontram disponíveis nos ficheiros `heap.h/.c` e `tabdispersao.h/.c`

As funções devem ser implementadas no ficheiro **prob1.c** fornecido.

- 1.1 [20 pontos] Implemente a seguinte função:

```
int adiciona_comercio(char* designacao, char *atividade, tabela_dispersao*
                      comercio_zona, tabela_dispersao* atividades_zona)
```

A tabela `comercio_zona` contém informação sobre as casas comerciais existentes; a chave é a sua designação e o valor é a sua classe (a classe pode assumir dois valores: “top”, “normal”). A tabela `atividades_zona` contém informação sobre as atividades comerciais existentes; nesta tabela a chave e o valor são o nome da atividade comercial.

A função adiciona a casa comercial de nome `designacao` e atividade comercial `atividade` na tabela `comercio_zona`. Se a casa comercial a inserir possui atividade ainda não presente na zona, esta atividade deve ser inserida na tabela `atividades_zona` e a classe da casa comercial é “top”. Caso contrário, a classe é “normal”. A função retorna 0 se a casa comercial já existe ou 1 se não existia. Considere que a designação de uma casa comercial é única.

Depois de implementada a função, o programa deverá apresentar:

```
Casa comercial adicionada com sucesso: Casa comercial Desp1
Casa comercial adicionada com sucesso: Casa comercial Desp2
Casa comercial adicionada com sucesso: Casa comercial Ocul1
Tabela 'atividades_zona' tem 2 elementos - ok
Tabela 'comercio_zona' tem 3 elementos - ok
```

- 1.2 [25 pontos] As ruas estão armazenadas numa fila de prioridade de mínimo, onde o valor guardado é a string “nome\_da\_ rua\_nome\_da\_freguesia” (ex: “Rua do Amial\_Paranhos”) e a prioridade é o número de casas comerciais existentes. Implemente a seguinte função:

```
char* remove_ rua_menos_comercio(heap* ruas, char *freg)
```

A função remove da heap ruas a rua com menor número de casas comerciais na freguesia freg e retorna essa string. A função retorna NULL se ocorrer algum erro (ruas não existe ou freg é NULL). A fila de prioridade ruas é alterada de acordo com a remoção solicitada.

**Notas:** Na implementação da heap foram incluídas as funções heap\_num\_elementos e heap\_elemento\_topo. A função get\_freguesia está implementada no ficheiro prob1.c

Depois de implementada a função, o programa deverá apresentar:

```
Rua com menos comercio em Paranhos: Rua da Telheira_Paranhos
Rua com menos comercio em Bonfim: Rua Pinto Bessa_Bonfim
Nao existem ruas comerciais na freguesia da Se
RUAS:
                Rua do Amial_Paranhos (14)
        Rua Latino Coelho_Bonfim (12)
                Rua Sao Roque da Lameira_Campanha (17)
Rua de Augusto Lessa_Paranhos (10)
                Rua do Vale Formoso_Paranhos (18)
        Rua do Campo Lindo_Paranhos (13)
                Rua Dom Joao IV_Bonfim (20)
```

**1.3** [10 pontos] Indique a complexidade da solução implementada na alínea anterior e uma justificação clara e sucinta no comentário assinalado na questão 1.3

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

**2** Tendo por base as bibliotecas de estruturas de dados **grafo** e **árvore avl** estudadas nas aulas, que se encontram disponíveis nos ficheiros `grafo.h/.c` e `avl.h/.c`, responda às questões seguintes. As funções devem ser implementadas no ficheiro **prob2.c** fornecido.

**2.1** [25 pontos] Uma rede social é mapeada num grafo não direccionado, onde um vértice representa o id de um utilizador. Implemente a função:

```
arvore_avl* sugestao_amizade(grafo *g, int u)
```

A função sugere ao utilizador u um conjunto de amigos. Este conjunto é constituído pelos amigos dos amigos do utilizador u que ainda não são amigos deste. A função retorna as sugestões numa árvore avl. Em caso de erro (grafo não existe ou utilizador u superior a tamanho do grafo), a função retorna NULL.

**Notas:** Na implementação do grafo foi incluída a função grafo\_adjacentes(grafo \*g, int v, int\* n). A implementação da árvore avl, `avl.h/.c`, foi alterada para guardar informação do tipo int.

Depois de implementada a função, o programa deverá apresentar:

```
Sugestões de amizade para '2': 6 7
Utilizador '12' ou rede nao existe
```

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