

No fim do exame deverá confirmar que submeteu corretamente os ficheiros no SIGEX, com o nome indicado no enunciado. [VERSÃO A]

1. [8 valores] Pretende-se criar um programa para analisar a informação da temperatura recolhida ao longo do tempo. A temperatura recolhida (em graus), de hora a hora, ao longo de muitos dias, está guardada num vetor temperatura de float alocado dinamicamente. **Uma sequência de valores constantes consecutivos chama-se planalto.**

1.1 [3 valores] Implemente a função

```
int maior_planalto (float * temperatura, int n)
```

Esta função deve calcular de forma eficiente o número máximo de horas consecutivas com a temperatura constante no vetor temperatura, sendo n o número de elementos no vetor. Por exemplo, para este vetor temperatura

[24.1 24.1 25.0 26.0 26.0 26.0 23.7], o resultado deverá ser 3.

1.2 [2 valores] Qual a complexidade da função anterior? Justifique em 2 linhas no ficheiro **prob1.c**.

1.3 [3 valores] Pretende-se imprimir de forma ordenada, do de menor largura para o de maior largura, todos os planaltos do vetor (em caso de planaltos com igual largura, a ordem é irrelevante). Para cada planalto, deve imprimir a largura e a temperatura correspondente.

Implemente a função

```
void impressao_ordenada_planaltos(float * temperatura, int n)
```

A função deve começar por guardar a informação dos planaltos numa minheap e depois imprimir. Toda a memória associada à minheap deve ser libertada após a impressão na consola.

Depois de implementada a função, o programa deverá apresentar:

```
Larg: 1; temp: 25.0  
Larg: 1; temp: 23.7  
Larg: 2; temp: 24.1  
Larg: 3; temp: 26.0
```

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

No fim do exame deverá confirmar que submeteu corretamente os ficheiros no SIGEX, com o nome indicado no enunciado. [VERSÃO B]

1. [8 valores] Pretende-se criar um programa para analisar a informação da temperatura recolhida ao longo do tempo. A temperatura recolhida (em graus), de hora a hora, ao longo de muitos dias, está guardada num vetor temperatura de float alocado dinamicamente.

1.1 [3 valores] Implemente de forma eficiente a função

```
float * realce_temperaturas (float * temperatura, int n)
```

Esta função deve retornar um novo vetor de temperaturas alocado dinamicamente, com o mesmo tamanho n do vetor original. Em caso de erro a função retorna NULL. Cada valor no vetor de retorno é a diferença entre o valor original de temperatura e a média local das temperaturas nesse instante. Para uma dada hora, a média local é a média das temperaturas na hora anterior, atual e seguinte. (Se um destes três valores não existir – início e fim do vetor – deve fazer a média apenas dos valores existentes.)

Se o vetor original for [24.1 24.1 25.0 26.0 26.0 26.0 23.7], o resultado deverá ser

[0.00 -0.30 -0.03 0.33 0.00 0.77 -1.15].

1.2 [2 valores] Qual a complexidade da função anterior? Justifique em 2 linhas no ficheiro **prob1.c**.

1.3 [3 valores] Pretende-se imprimir de forma ordenada o vetor de valores realçados, do de menor valor realçado para o de maior valor realçado (em caso de igualdade, a ordem é irrelevante). Para cada valor, deve imprimir o valor realçado e a temperatura correspondente.

Implemente a função que recebe as **temperaturas originais**

```
void impressao_ordenada_realcada (float * temperatura, int n)
```

A função deve começar por guardar a informação relevante numa minheap e depois imprimir. Toda a memória associada à minheap deve ser libertada após a impressão na consola.

Depois de implementada a função, o programa deverá apresentar:

```
realcado: -1.15; temp: 23.7
realcado: -0.30; temp: 24.1
realcado: -0.03; temp: 25.0
realcado: 0.00; temp: 26.0
realcado: 0.00; temp: 24.1
realcado: 0.33; temp: 26.0
realcado: 0.77; temp: 26.0
```

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

[VERSAO A]

2 [7 valores] Tendo por base as bibliotecas de estruturas de dados apresentadas em Programação 2, implemente as funcionalidades pedidas nas duas alíneas seguintes no ficheiro **prob2.c** fornecido. Sempre que conveniente utilize as funções disponíveis.

Uma rede social é mapeada num **grafo direcionado**, onde um vértice representa um utilizador e a presença da aresta do vértice A para o vértice B indica que o utilizador A influencia o utilizador B. Também se diz que A influencia B indiretamente se houver um caminho (dirigido) de A para B.

2.1 [4 valores] Implemente a função

```
int * ZonaInfluencia (grafo *g, int utl)
```

que permite obter todos os utilizadores influenciados direta ou indiretamente por utl. A função devolve um vetor alocado dinamicamente; deve colocar o valor 1 na posição i do vetor se o utilizador i for influenciado por utl e o valor 0 em caso contrário. A função retorna NULL em caso de erro.

Depois de implementada a função, o programa deverá apresentar:

```
Influencia do utilizador 0
0 1 0 1 0 1 1
Influencia do utilizador 3
1 1 0 0 0 1 1
```

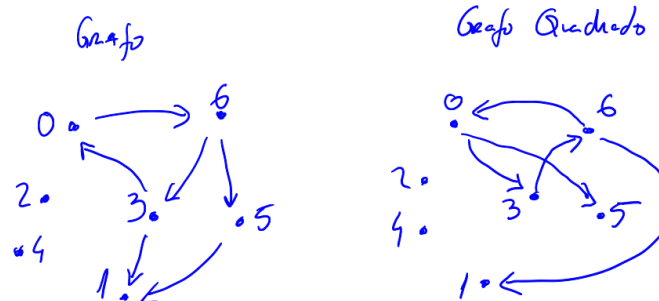
2.2 [3 valores] Suponha que quer usar uma árvore AVL para ordenar um conjunto de valores. Descreva como procederia. Qual a complexidade para este processo de ordenação? Justifique de forma simples e objetiva no local indicado no ficheiro **prob2.c**.

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

[VERSAO B]

2 [7 valores] Tendo por base as bibliotecas de estruturas de dados apresentadas em Programação 2, implemente as funcionalidades pedidas nas duas alíneas seguintes no ficheiro **prob2.c** fornecido. Sempre que conveniente utilize as funções disponíveis.

O quadrado de um grafo orientado $G=(V,A)$ é o grafo orientado $G^2=(V,A')$, onde A' é o conjunto das arestas (x,y) para as quais existe um vértice $z \in V$ tal que (x,z) e (z,y) são arestas de A ; ou seja, (x,y) é uma aresta de G^2 se e só se existe um vértice $z \in V$ tal que (x,z,y) é um caminho de G .



2.1 [4 valores] Implemente a função

`grafo * grafo_quadrado (grafo *g)`

que calcula e retorna o grafo correspondente ao quadrado do grafo g . A função deverá retornar NULL em caso de erro.

Depois de implementada a função, o programa deverá apresentar:

```
Grafo Quadrado
0: 5->3
1:
2:
3: 6
4:
5:
6: 1->0
```

2.2 [3 valores] Se tivesse um grafo g representado pela sua matriz de adjacência m , como poderia obter a matriz de adjacência do quadrado do grafo g ? Qual a complexidade desse cálculo, em função de N , onde N é o número de nós do grafo? Justifique de forma simples e objetiva no local indicado no ficheiro **prob2.c**.

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

Parte prática [15/20 valores]

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