

Aula prática 3

Para cada exercício, consulte a respetiva pasta incluída em **P03.zip**, disponível no Moodle.

Os exercícios 1 e 2 podem fazer no moodle, com o CodeRunner.

1 – Utilize o ficheiro **vector.cpp** que cria e manipula vectores de números inteiros e implemente as seguintes funções:

a) **int imprime_vetor(vector<int> vec)**

Imprime os elementos do vetor;

Imprime o tamanho do vetor.

b) **vector<int> ordena_euromilhoes(vector<int> vec)**

Aceita a chave do euromilhões não ordenada em que os últimos dois números correspondem ao número de estrelas.

Ordena o vector, tendo em atenção que deverá aparecer primeiramente os números da chave e só depois os números das estrelas.

Exemplo: o vector com os elementos:

43 | 3 | 27 | 33 | 15 | 5 | 2

deverá tornar-se em

3 | 15 | 27 | 33 | 43 | 2 | 5

Depois de implementar as alíneas, o programa deverá apresentar um resultado semelhante ao exemplo:

```
Elementos do vetor: { 1 99 37 40 11 }
Tamanho do vector: 5

Chave 1 do Euromilhões:
Elementos do vetor: { 2 13 7 26 15 20 10 }
Tamanho do vector: 7

Chave 1 do Euromilhões ordenada:
Elementos do vetor: { 2 7 13 15 26 10 20 }
Tamanho do vector: 7

Chave 2 do Euromilhões:
Elementos do vetor: { 14 5 33 21 2 44 20 }
Tamanho do vector: 7

Chave 2 do Euromilhões ordenada:
Elementos do vetor: { 2 5 14 21 33 20 44 }
Tamanho do vector: 7
```

2 - Considere o ficheiro **auto_parts_price.cpp**. Uma empresa de automóveis guarda o seu catálogo de vendas (designação de peças e seu respectivo valor) no ficheiro de texto "auto_price_table.txt" (as linhas ímpares contêm a designação e as pares o valor das peças). Implemente as seguintes funções:

a) **vector<string> cria_vet_componentes()**

Lê o ficheiro "auto_price_table.txt", colocando informação referente aos nomes das peças num vetor.

b) **vector<float> cria_vet_custos()**

Lê o ficheiro "auto_price_table.txt", colocando informação referente aos respectivos valores das peças noutra vetor.

c) **int custo_componente(string str_componente,
vector<string> componentes,
vector<float> custos)**

Apresenta no ecrã o preço da peça str_componente.

Retorna 0 se conseguiu ou -1 se não conseguiu.

Depois de implementar as alíneas, o programa deverá apresentar um resultado semelhante ao exemplo:

```
O preço do componente Jante é: 200.5
Return de Jante é: 0

O preço do componente Retrovisor é: 34
Return de Retrovisor é: 0

Componente Roda não encontrado.
Return de Roda é: -1

O preço do componente Vidro traseiro é: 210
Return de Vidro traseiro é: 0
```

3 - Implemente a função **vector<int> encontrePadrao(vector<int> v, vector<int> vp);** que deve receber dois vectores, o 2º vector representa uma sequência de números que representará um padrão a ser procurado no primeiro vector. Como retorno, a função retornará um vector com as posições iniciais onde o padrão foi encontrado.

Exemplo: Imagine o vector v[1,2,1,2,1,2,1] e o vector padrão vp[1,2,1].

Como resultado, deve-se encontrar 3 padrões, conforme destacados abaixo.

- 1,2,1,2,1,2,1
- 1,2,1,2,1,2,1
- 1,2,1,2,1,2,1

O vector de retorno será vr[0,2,4]

Depois de implementar a função, o programa deverá apresentar um resultado semelhante ao exemplo:

```
V1=[ 1 2 1 2 1 2 1 ]
Vpadrao=[ 1 2 1 ]
Vresultado=[ 0 2 4 ]
V2=[ 1 4 5 1 5 5 4 5 2 1 ]
Vpadrao2=[ 4 5 ]
Vresultado=[ 1 6 ]
```

4 - Implemente a função **vector<Person*> findperson(vector<Person>& vperson, int min, int max)** que deve receber um vector de pessoas, e dois inteiros correspondendo a um intervalo de valores. Como retorno, a função retornará um vector com as pessoas com idade compreendida entre **min** e **max**.

```
class Person {  
private:  
    string name;  
    int age;  
public:  
    Person(string name,int ag) {}  
    void setName(string name) {}  
    void setAge(int age) {}  
    string getName() const {}  
    int getAge() const {}  
}
```

Depois de implementar a função, o programa deverá apresentar um resultado semelhante ao exemplo:

```
Vresultado=[ Ana - 41 ; Tiago - 40 ; Ferando - 44 ; ]  
Vresultado=[ Pedro - 23 ; Susana - 22 ; Teresa - 22 ; ]
```