

- O nome do projeto deve ser “EN\_(Número aluno)”.
- A compilação do código vale 0,5 valores.

### Parte I

1. Implemente as seguintes funções globais no ficheiro *main.cpp*.
  - (a) (1,5 valores) A função `Multiplica` que devolve a multiplicação de todos os elementos de um vetor de inteiros.
  - (b) (1,5 valores) A função `Multiplos5` que devolve quantos elementos de um vetor de inteiros são múltiplos de 5.
  - (c) (2,5 valores) A função `Completo` do tipo `bool` que recebe um vetor de inteiros e verifica se esse vetor é completo. Um vetor de dimensão  $n$  é completo se tiver todos os números entre 1 e  $n$ . Por exemplo, o vetor (2, 1, 4, 3) é completo.
  - (d) (2 valores) A função `EscreveVetor` que recebe um vetor de inteiros e o escreve num ficheiro *Vetor.txt* localizado na pasta *build* do projeto com o formato “( $v_1, \dots, v_n$ )”.

### Parte II

2. Pretende-se criar a classe `SalaCinema` para gerir uma sala de cinema retangular. A classe deve estar preparada para lidar com todo o tipo de erros que possam ocorrer, lançando exceções com mensagens de erro específicas. A classe deve ter:
  - (a) (1,5 valores) Como atributos o preço de um bilhete *standard*, o título do filme em exibição e uma matriz que represente a ocupação da sala. Os lugares ocupados serão representados na matriz com o valor 1 enquanto os lugares livres serão representados pelo valor 0.
  - (b) (1,5 valores) Um construtor que receba o número de filas e o número de colunas da sala, o preço de um bilhete *standard* e o título do filme em exibição. Todos os lugares da sala devem estar inicialmente livres.
  - (c) (1,5 valores) Um método `Compra` que, dada a fila e a coluna, permite comprar um lugar.
  - (d) (1,5 valores) Um método `Imprime` que imprima para o ecrã a informação associada à sala de cinema no formato seguinte:

```
Filme: Nome do Filme
Ocupacao:
(Ocupação da sala - ver output)
```

- (e) (2 valores) Sabendo que os bilhetes que ficam na metade da sala mais afastada da tela têm uma valorização de 10% comparativamente aos bilhetes *standard*, crie um método *Receita* que calcule a receita que a sala irá gerar com a ocupação atual. Caso o número de filas da sala seja ímpar, assume-se que o preço dos bilhetes na fila do meio é o preço *standard*. Considere que a fila 0 é a que está mais próxima da tela.
- (f) (2 valores) Um método *TaxaTS* que indique a taxa de ocupação da submatriz triangular superior da ocupação da sala. Note que este conceito só existe para matrizes quadradas e que a entrada  $(i, j)$  da matriz se encontra na submatriz triangular superior se  $i < j$ .
- (g) (2 valores) O operador `==` que verifique se duas salas de cinema têm o mesmo número de lugares. Este operador deve manter o seu significado original, isto é, deve devolver *true* se o número de lugares nas salas for igual.

O programa deve funcionar com a seguinte função *main* após implementar a condição do `if`.

```
//Implementar funções globais

int main(){
    //Parte I
    vector<int> v = {2, 5, 10, -1, -3};
    cout << "Multiplicacao: " << Multiplica(v) << endl;
    cout << "Multiplos5: " << Multiplos5(v) << endl;
    cout << "Completo? " << Completo(v) << endl << endl;
    EscreveVetor(v);

    //Parte II
    SalaCinema s1(5, 3, 10, "Barbie");
    s1.Compra(2, 1);
    s1.Compra(0, 2);
    s1.Compra(4, 0);
    s1.Compra(4, 2);
    s1.Imprime();
    cout << "Receita: " << s1.Receita() << endl << endl;

    SalaCinema s2(4, 4, 7, "Oppenheimer");
    s2.Compra(1, 2);
    s2.Compra(1, 3);
    s2.Imprime();
    cout << "TaxaTS: " << 100*s2.TaxaTS() << endl;

    if(/*Condicao a preencher pelo aluno*/)
        cout << "s1 e s2 sao diferentes\n";
    else
        cout << "s1 e s2 sao iguais\n";

    return 0;
}
```

O output esperado é o seguinte:

Multiplicacao: 300

Multiplos5: 2

Completo? 0

Titulo: Barbie

Ocupacao:

0 0 1

0 0 0

0 1 0

0 0 0

1 0 1

Receita: 42

Titulo: Oppenheimer

Ocupacao:

0 0 0 0

0 0 1 1

0 0 0 0

0 0 0 0

TaxaTS: 33.3333

s1 e s2 sao diferentes