QUESTÃO 1 (2 valores)

Faça um programa que apresenta a seguinte saída, perguntando ao utilizador o número máximo (no exemplo, 9). Este número deve ser sempre ímpar.

```
1 2 3 4 5 6 7 8 9
  2 3 4 5 6 7 8
    3 4 5 6 7
      4 5 6
        5
```

Solução:

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
 int resp, n;
 while (true)
 {
  cout << "Dê o número máximo de elementos (ímpar):" << endl;
  cin >> resp;
  n = resp/2;
  if(n*2 == resp) cout << "Este numero é par..." << endl;
  else break;
 }
 for (int i = 0; i <= n; ++i)
 {
  for (int k = 0; k <= i; ++k) cout << " ";
  for (int j = i + 1; j <= resp - i; ++ j) cout << j << " ";
  cout << endl;
 }
 return 0;
}
```

QUESTÃO 2 (2 valores)

Escreva um programa que leia 3 notas de um aluno e a média das notas dos exercícios realizados por ele. Calcular a média de aproveitamento, usando a fórmula: MA = (N1 + N2*2 + N3*3 + ME)/7. A partir da média, informar o conceito de acordo com a tabela:

maior ou igual a 9	A
maior ou igual a 7.5 e menor que 9	B
maior ou igual a 6 e menor que 7.5	C
maior ou igual a 4 e menor que 6	D
menor que 4	E

Solução:

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
 float n1, n2, n3, media, ma;
 cout << "Dê a primeira nota:" << endl;
 cin >> n1;
 cout << "Dê a segunda nota:" << endl;
 cin >> n2;
 cout << "Dê a terceira nota:" << endl;
 cin >> n3;
```

```
 cout << "Dê a média dos exercícios:" << endl;
 cin >> media;
 ma = (n1 + n2*2 + n3*3 + media)/7;
 cout << ma;
 if (ma >= 9) cout << "A" << endl;
 else if (ma >= 7.5) cout << "B" << endl;
   else if (ma >= 6) cout << "C" << endl;
     else if (ma >= 4) cout << "D" << endl;
         else cout << "E" << endl;
 return 0;
}
```

QUESTÃO 3 (6 valores)

Um animal contém **nome**, **comprimento**, número de **patas** (o padrão é 4), uma **cor**, **ambiente** e uma **velocidade** (em m/s).

Um peixe é um animal, tem 0 patas, o seu ambiente é o mar (padrão), cor cinzenta (padrão). Além disso, o peixe tem como **características**: barbatana e cauda;

Um mamífero é um animal, o seu **ambiente** é a terra (padrão);

Um urso é um mamífero, cor castanha e o seu **alimento** preferido é o mel.

Codifique as classes **animal**, **peixe** e **mamífero**.

Para a classe **Animal**, codifique os métodos que permitam:

- Criar o animal de forma personalizada;
- Alterar o seu nome;
- Alterar o número de patas;
- Alterar a cor;
- Alterar o ambiente em que ele vive;
- Alterar a velocidade de deslocamento;
- Imprimir todos os seus dados;

Para a classe **Peixe**, codifique métodos que permitam:

- Criar o peixe de forma personalizada;
- Alterar as suas características;
- Imprimir seus dados;

Para a classe **Mamifero**, codifique métodos que permitam:

- Criar o mamífero de forma personalizada;
- Alterar o seu alimento;
- Imprimir seus dados;

Solução:

```
#include <iostream>
namespace std {
class Animal {
private:
 string nome;
```

```
 float comprimento;
 int nPatas;
 string cor;
 string ambiente;
 float velocidade;
public:
 Animal();
 virtual ~Animal();
  string getAmbiente() const { return this->ambiente; }
  float getComprimento() const { return this->comprimento; }
  string getCor() const { return this->cor; }
  string getNome() const { return this->nome; }
  int getPatas() const { return this->nPatas; }
  float getVelocidade() const { return this->velocidade; }
  void setAmbiente(string ambiente) { this->alimento = alimento; }
  void setComprimento(float comprimento) { this->alimento = alimento; }
  void setCor(string cor) { this->cor = cor; }
  void setNome(string nome) { this->nome = nome; }
  void setPatas(int nPatas) { this->nPatas = nPatas; }
  void setVelocidade(float velocidade) { this->velocidade = velocidade; }
  virtual void imprimeInfo();
};
} /* namespace std */

#include "Animal.h"
namespace std {
class Peixe : public Animal
{
private:
  string caracteristicas;
public:
  Peixe();
  Peixe(string);
  virtual ~Peixe();
  string getCaracteristicas() const { return this->caracteristicas; }
  void setCaracteristicas(string caracteristicas) { this->caracteristicas =
caracteristicas; }
  void imprimeInfo();
};
} /* namespace std */

#include "Animal.h"
namespace std {
class Mamifero : public Animal
{
private:
  string alimento;
public:
  Mamifero();
  Mamifero(Animal, string);
  virtual ~Mamifero();
  string getAlimento() const { return this->alimento; }
  void setAlimento(string alimento) { this->alimento = alimento; }
  void imprimeInfo();
};
} /* namespace std */
```

```
#include "Animal.h"
namespace std {
   Animal::Animal() {
   cout << "Dê o nome:" << endl; cin >> this->nome;
   cout << "Dê o nº de patas:" << endl; cin >> this->nPatas;
   cout << "Dê a cor:" << endl; cin >> this->cor;
   cout << "Dê a velocidade (m/s):" << endl; cin >> this->velocidade;
   cout << "Dê o ambiente:" << endl; cin >> this->ambiente;
   cout << "Dê o comprimento:" << endl; cin >> this->comprimento;
   }
   void Animal::imprimeInfo()
   {
    cout << "Nome:" << nome << endl;
    cout << "nº de patas:" << nPatas << endl;
    cout << "Cor:" << cor << endl;
    cout << "Velocidade (m/s):" << velocidade << endl;
    cout << "Ambiente:" << ambiente << endl;
    cout << "Comprimento:" << comprimento << endl;
   }
   Animal::~Animal() {}
} /* namespace std */

#include "Mamifero.h"
namespace std {
   Mamifero::Mamifero() {
    cout << "Dê o alimento:" << endl; cin >> this->alimento;
   }
   void Mamifero::imprimeInfo()
   {
    this->Animal::imprimeInfo();
    cout << "Alimento:" << alimento << endl;
   }
   Mamifero::~Mamifero() {}
} /* namespace std */

#include "Peixe.h"
namespace std {
   Peixe::Peixe()
   {
      cout << "Dê as características:" << endl; cin >> this->caracteristicas;
   }
   Peixe::Peixe(string c)
   {
      caracteristicas = c;
   }
   void Peixe::imprimeInfo()
   {
      this->Animal::imprimeInfo();
      cout << "Características: " << caracteristicas << endl;
    }
   Peixe::~Peixe() {}
} /* namespace std */
```

QUESTÃO 4 (2 valores)

Para a questão 3, crie um ficheiro de teste com o nome TesteAnimais.java, de forma a ter um jardim zoológico com os seguintes animais: camelo, tubarão e urso-do-canadá.

Exemplo de execução:

Zoo:

Animal: Camelo
Comprimento: 150 cm
Patas: 4
Cor: Amarelo
Ambiente: Terra
Velocidade: 2.0 m/s

Animal: Tubarão
Comprimento: 300 cm
Patas: 0
Cor: Cinza
Ambiente: Mar
Velocidade: 1.5 m/s
Característica: Barbatanas e cauda

Animal: Urso-do-canadá
Comprimento. 180 cm
Patas: 4
Cor: Vermelho
Ambiente: Terra
Velocidade: 0.5 m/s
Alimento: mel

Solução:

Aos ficheiros .h seriam declaradas as respectivas assinaturas dos métodos abaixo e os mesmos incluídos nos respectivos ficheiros .cpp:

```
void Animal::gravaInfo()
{
       ofstream fout("testeanimais.txt");
       fout << "Nome:" << nome << endl;
       fout << "nº de patas:" << nPatas << endl;
       fout << "Cor:" << cor << endl;
       fout << "Velocidade (m/s):" << velocidade << endl;
       fout << "Ambiente:" << ambiente << endl;
       fout << "Comprimento:" << comprimento << endl;
       fout.close();
}

void Mamifero::gravaInfo()
{
     this->Animal::gravaInfo();
     fstream fout;
     fout.open("testeanimais.txt",ios::ate | ios::out | ios::in);
     fout.seekg(0,ios::end);
     fout << alimento << endl;
     fout.close();
 }
```

```
void Peixe::gravaInfo()
{
     this->Animal::gravaInfo();
     fstream fout;
     fout.open("testeanimais.txt",ios::ate | ios::out | ios::in);
     fout.seekg(0,ios::end);
     fout << carcateristicas << endl;
     fout.close();
 }
```

Por fim, adicionar a seguinte biblioteca no início da classe Animal:

```
#include <iostream>
#include <fstream>
```

FIM