```
/**********************
Capteur.h
**********************/
```
3/5

```
#pragma once
#include <string>

using namespace std;

class Capteur
{
private :
        static int nombreDeCapteurs;
        string type;
        string noSerie;

public:
        Capteur(string type = "TEMP", string noSerie = "33");
        string getType() { return this->type; }
        string getNoSerie() { return this->noSerie; }
        static int getNombreCapteurs() { return nombreDeCapteurs; }
        ~Capteur();
};



/**********************
CapteurTemperature.h
**********************/
```
2/2

```
#pragma once
#include "Capteur.h"
class CapteurTemperature :
    public Capteur
{
protected :
    double temperatureMinAutorisee;
    double temperatureMaxAutorisee;
public:
    CapteurTemperature(string numeroSerie, double min, double max);
    double getTemperature();
};



/**********************
CapteurTemperatureExterieure.h
**********************/
```
2/2

```
#pragma once
#include "CapteurTemperature.h"
class CapteurTemperatureExterieure :
    public CapteurTemperature
{
public:
    CapteurTemperatureExterieure(string derniersChiffresNumeroSerie, double min, double max);
    bool verifieConformite();
};



/**********************
Capteur.cpp
**********************/
```
4/4

```
#include "Capteur.h"

int nombreDeCapteurs = 0;

Capteur::Capteur(string type, string noSerie)
{
        this->type = type;
        this->noSerie = noSerie;
        nombreDeCapteurs++;
}

Capteur::~Capteur()
{
        nombreDeCapteurs--;
}


/**********************
CapteurTemperature.cpp
**********************/
```
5/7

```
#include "CapteurTemperature.h"

CapteurTemperature::CapteurTemperature(string numeroSerie, double min, double max)
        :Capteur("TEMP", numeroSerie)
{
```

```
        this->temperatureMinAutorisee = min;
        this->temperatureMaxAutorisee = max;
}

double CapteurTemperature::getTemperature()
{
        double alea = rand(), min=0, max=1;
        alea / 32767;
        alea = alea * (max - min);
        alea = (max - min) + min;
        return alea;
}

/**********************
CapteurTemperatureExterieure.cpp
**********************/
#include "CapteurTemperatureExterieure.h"

CapteurTemperatureExterieure::CapteurTemperatureExterieure(string derniersChiffresNumeroSerie, double min, doubl
e max)
        : CapteurTemperature(derniersChiffresNumeroSerie,min,max)
{
}

bool CapteurTemperatureExterieure::verifieConformite()
{
        if (temperatureMinAutorisee < -50 && temperatureMaxAutorisee > 60)
        {
                return false;
        }
        if (temperatureMaxAutorisee < 60 && temperatureMinAutorisee > -50)
        {
                return true;
        }
}

/**********************
TestCapteurs_A_Completer.cpp
**********************/
/***********************************************
Programme principal à compléter
Au final, l'exécution du programme doit
produire un résultat similaire à l'exécutable fourni
***********************************************/
#define _CRT_SECURE_NO_WARNINGS

#include <iostream> // bibliothèque de gestion des E/S
#include <conio.h>      // gestion de la console (ici _getch())
#include <windows.h>
#include <vector>
#include "CapteurTemperatureExterieure.h"

using namespace std; // utilisation de l'espace de nommage standard


/*=============================================
Fonction principale
=============================================*/
int main()
{
        // Prise en compte des accents
        // Il faudra choisir la police de caractères Consolas ou Lucida
        SetConsoleOutputCP(1252);

        //==== Déclaration d'éventuelles variables locales ====
        Capteur* leCapteur;
        int i, j, c = 1, val = 1;
        string strDebut;
        string strFin;

        // Affichage du nombre de capteurs
        cout << "ETAPE1 : NOMBRE DE CAPTEURS = " << leCapteur->getNombreCapteurs() << " (attendu 0)" << endl << endl;

        // Créer un tableau de 20 capteurs de température (modèles de base)
        CapteurTemperature* lesCapteurs[20];
        // Ces  capteurs de temperature  ont des numéros de séries allant de "111101" à "111120"
        // de temperature mini -20.5 et temperature maxi +80.5

        for (i = 0; i < 10; i++)
        {
                strDebut = "11110";
                strFin = to_string(val);
                lesCapteurs[i] = new CapteurTemperature(strDebut += strFin, -20.5, 80.5);
                val = val + 1;
        }
```

```
        for (i = 10; i < 20; i++)
        {
                strDebut = "1111";
                strFin = to_string(val);
                lesCapteurs[i] = new CapteurTemperature(strDebut += strFin, -20.5, 80.5);
                val = val + 1;
        }

        // Affichage du nombre de capteurs
        cout << "ETAPE2 : NOMBRE DE CAPTEURS = " << lesCapteurs[20]->getNombreCapteurs() << " (attendu 20)" << endl <
< endl;

        // Afficher 8 mesures par capteurs (20 lignes de 8 valeurs)
        cout << "CAPTEURS DE TEMPERATURE (MODELE DE BASE) " << endl;
        cout.precision(4);      // 4 digits de précision
        for (i = 0; i < 20; i++)
        {
                for (j = 0; j < 8; j++)
                {
                        cout << "Les valeurs sont pour le capteur " << c << " de " << lesCapteurs[i]->getTemperature() << en
dl;
                }
                c++;
        }

        // Supprimer de la mémoire 10 capteurs
        for (i = 0; i < 10; i++)
        {
                delete lesCapteurs[i];
        }

        // Affichage du nombre de capteurs
        cout << "ETAPE3 : NOMBRE DE CAPTEURS = " << lesCapteurs[10]->getNombreCapteurs() << " (attendu 10)" << endl <
< endl;

        // Créer 12 capteurs de température extérieure dont les numéros de séries finissent par "5501" à "5512"
        // Les 11 premiers capteurs ont une temperature mini -85.5 et temperature maxi +300.5
        // le dernier capteur a une température mini de -45.5 et maxi de 60.6
        val = 1;
        for (i = 0; i < 10; i++)
        {
                strDebut = "550";
                strFin = to_string(val);
                lesCapteurs[i] = new CapteurTemperature(strDebut += strFin, -85.5, 300.5);
                val = val + 1;
        }
        for (i = 10; i < 12; i++)
        {
                strDebut = "55";
                strFin = to_string(val);
                lesCapteurs[i] = new CapteurTemperature(strDebut += strFin, -85.5, 300.5);
                val = val + 1;
        }

        lesCapteurs[12] = new CapteurTemperature(strDebut += strFin, -45.5, 60.6);

        // Afficher 8 mesures par capteurs (12 lignes de 8 valeurs) et afficher pour chaque capteur s'il est con
forme ou non
        cout << "\n\nCAPTEURS DE TEMPERATURE (MODELE EXTERIEUR) " << endl;
        c = 1;
        for (i = 0; i < 12; i++)
        {
                for (j = 0; j < 8; j++)
                {
                        cout << "Les valeurs sont pour le capteur " << c << " de " << lesCapteurs[i]->getTemperature() << en
dl;
                }
                c++;
        }

        // Affichage du nombre de capteurs
        cout << "ETAPE4 : NOMBRE DE CAPTEURS = " << lesCapteurs[i]->getNombreCapteurs() << " (attendu 22)" << endl <<
 endl;

        // Supprimer tous les capteurs
        delete lesCapteurs;

        // Affichage du nombre de capteurs
        cout << "ETAPE5 : NOMBRE DE CAPTEURS = " << lesCapteurs[i]->getNombreCapteurs() << " (attendu 0)" << endl <<
endl;

        _getch(); // attente d'appui sur une touche
        return 0; // sortie du programme
}
```

4

2

1

2,5

0,5