

1 Guirlandes

Utiliser la surcharge d'opérateur pour la classe Guirlande, ajouter des fonctions membre pour allumer et éteindre la guirlande.

```
int main()
{ /*
    Guirlande MaGuirlande1(1)
    Guirlande MaGuirlande2(1)
    MaGuirlande2.toggle()
    MaGuirlande=MaGuirlande1+MaGuirlande2
    cout << MaGiurlande << end
    MaGuirlande.toggle()
    cout << MaGuirlande << endl
    MaGuirlande*=5
    cout << MaGiurlande << end
    MaGuirlande.toggle[]
    cout << maGuirlente << endl
    MaGuirlande1=MaGuirlande/3
    MaGuirlande1.Eteindre()
    cout << MaGiurlande1 << end
    MaGuirlande1.Allumer[]
    cout << MaGiurlande1 << end
    MaGuirlande=5*MaGuirlande1
    cout << MaGourlande << end
    MaGuirlande2=MaGuirlande1-MaGuirlande
    cout << Guirlande2 << end
    */
    return EXIT_SUCCESS;
}
```

Résultat attendu :

```
.o
o.
o.o.o.o.o.o.
.o.o.o.o.o.o
...
.o.
.o..o..o..o..o..o.
.o..o..o..o..o..o.
```

A NOTER :

Le main.cpp fourni par votre responsable de projet est buggée, illisible donc inutilisable.

Proposer à votre responsable de projet des sources avec les commentaires nécessaires à la compréhension du programme.

2 Classe Time

Le point de départ de cet exercice est la classe `Time` définie comme suit :

```
class Time {
private :
    unsigned short hour ;
    unsigned short minute ;
    unsigned short second ;

public :
    Time(unsigned short hour, unsigned short minute, unsigned short second);
};
```

L'objectif est de définir des fonctions de comparaison sur la classe `Time` en utilisant la possibilité de surcharger les opérateurs du C++. Par exemple, l'opérateur `<` peut être surchargé ainsi :

```
class Time {
    ...
    bool operator<(const Time & other) const ;
    // le temps est-il antérieur a l'autre temps ?
    ...
};
```

- ① Compléter la classe `Time` pour surcharger les 6 opérateurs de comparaison : `==`, `!=`, `<`, `>`, `<=`, `>=`. Écrire un programme de test qui permet de vérifier la classe `Time` ainsi écrite.

Q

- ② Ajouter les opérateurs de pre-incrémentation et de pre-décrémentation : `++`, `--`.

Q

- ③ Il est également possible de surcharger un opérateur à l'extérieur de la classe. La signature est alors de la forme :

```
bool operator< (const Time &t1, const Time &t2)
// t1 est-il antérieur a t2 ?
```

Que constatez-vous lorsque vous surchargez l'opérateur `<` de cette façon ?

Quelle fonction sera appelée, fonction membre ou fonction non-membre ?