	Lycée de l'Hyrôme - Chemillé	2012 - 2013
	<b>Procédures et fonctions</b>	<b>Langage C++</b>
<b>BTS IRIS</b>		<b>TP3</b>

## Objectifs

- Réaliser des procédures et des fonctions.
- Mettre en œuvre les différents types de paramètres.
- Réaliser des méthodes.
- Déclarer des attributs.

## Ressources disponibles

- Un PC.
- Un EDI C++.

## 1. Tableau d'entiers avec des fonctions

### 1.1. Présentation

Nous allons reprendre le TP2 (partie 1) et réaliser les 5 fonctionnalités : remplir le tableau, rechercher le plus petit entier, rechercher le plus grand entier, faire la moyenne des entiers et affichage des résultats sous forme de procédure ou de fonction.

#### **Remarque :**

La correction du TP précédent est fournie en annexe.

Les prototypes des différentes fonctions seront écrits dans le fichier "application.h" avant la déclaration de la classe "Application".

### 1.2. fonction "saisieTableau"

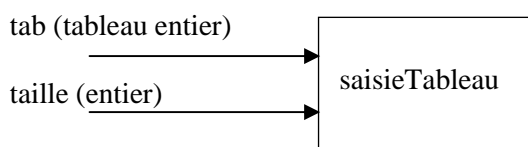
Fonctionnalité :

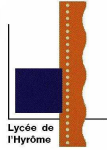
Rempli le tableau.

Paramètres :

tab : tableau à remplir

taille : taille du tableau



	<b>Lycée de l'Hyrôme - Chemillé</b>	2012 - 2013
	<b>Procédures et fonctions</b>	<b>Langage C++</b>
		<b>TP3</b>

Ecrire le prototype, définir la fonction et réaliser son appel (il remplacera le bloc "fonctionnalité 1" de la correction).

### 1.3. fonction " plusPetit"

Fonctionnalité :

Recherche le plus petit entier dans un tableau.

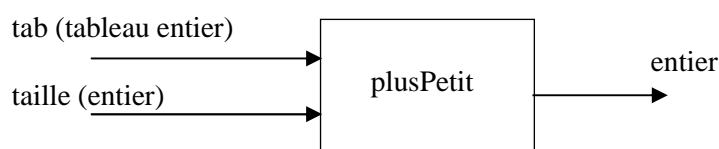
Paramètres :

tab : tableau à traiter

taille : taille du tableau

Retour :

plus petit entier



Ecrire le prototype, définir la fonction et réaliser son appel (il remplacera le bloc "fonctionnalité 2" de la correction).

### 1.4. fonction "plusGrand"

Fonctionnalité :

Recherche le plus grand entier dans un tableau.

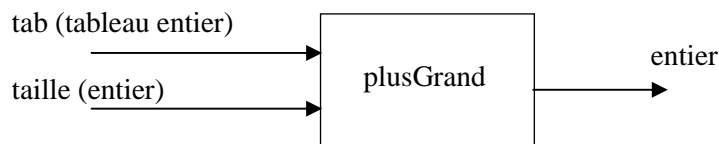
Paramètres :

tab : tableau à traiter

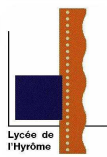
taille : taille du tableau

Retour :

plus grand entier



Ecrire le prototype, définir la fonction et réaliser son appel (il remplacera le bloc "fonctionnalité 3" de la correction).

	Lycée de l'Hyrôme - Chemillé	2012 - 2013
	<b>Procédures et fonctions</b>	<b>Langage C++</b>
<b>BTS IRIS</b>		<b>TP3</b>

### 1.5. fonction "moyenne"

Fonctionnalité :

Calcul la moyenne des entiers d'un tableau.

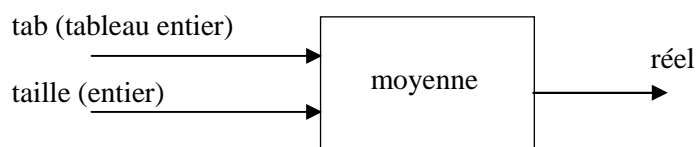
Paramètres :

tab : tableau à traiter

taille : taille du tableau

Retour :

moyenne des entiers



Ecrire le prototype, définir la fonction et réaliser son appel (il remplacera le bloc "fonctionnalité 4" de la correction).

### 1.6. fonction "afficher"

Fonctionnalité :

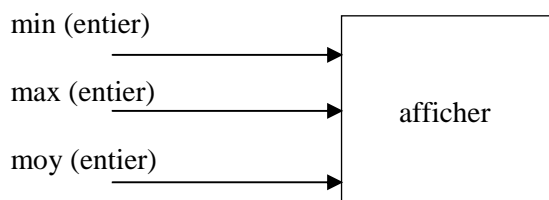
Affiche les résultats.

Paramètres :

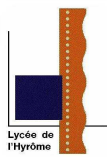
min : entier le plus petit

max : entier le plus grand

moy : moyenne des entiers



Ecrire le prototype, définir la fonction et réaliser son appel (il remplacera le bloc "fonctionnalité 5" de la correction).

	Lycée de l'Hyrôme - Chemillé	2012 - 2013
	<b>Procédures et fonctions</b>	<b>Langage C++</b>
<b>BTS IRIS</b>		<b>TP3</b>

### 1.7. "main"

Réaliser le programme principal permettant de tester les différentes fonctions.

## 2. Tableau d'entiers avec des méthodes

### 2.1. Présentation

Des méthodes sont des fonctions qui appartiennent à une classe.  
Comme pour les fonctions il faudra les déclarer et les définir.

### 2.2. Déclaration des méthodes

Les différentes méthodes que nous allons réaliser (correspondant aux fonctions définies ci-dessus) seront déclarer dans le fichier "Application.h" dans la classe "Application" dans un bloc privé.

```
class Application
{
    public:
        void lancer();

    private:

        // déclarer ici les différentes méthodes
};
```

Déclarer les 5 méthodes.

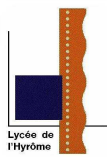
#### Remarque :

Le mot "private" indique que les méthodes ne sont pas accessibles en dehors de la classe.

### 2.3. Définir des méthodes

Les 5 méthodes seront définies en respectant le squelette suivant.

```
<retour> Application::<nom de la méthode>(<paramètres>)
{
}
}
```

	Lycée de l'Hyrôme - Chemillé	2012 - 2013
	<b>Procédures et fonctions</b>	<b>Langage C++</b>
<b>BTS IRIS</b>		<b>TP3</b>

## 2.4. "main"

Réaliser le programme principal permettant de tester les différentes fonctions.

## 3. Tableau d'entiers avec des méthodes et des attributs

### 3.1. Présentation

Les attributs sont des variables qui sont globales au niveau d'une classe, c'est à dire que l'on peut y accéder à partir d'une méthode sans avoir à les passer en paramètre.

### 3.2. Déclaration des attributs

Nous avons 5 données : MAXI, tab, min, max, et moy.

La constante MAXI sera déclarée en dehors de la classe (avant "class Application").

Les 4 autres données seront des attributs et sont déclarés dans la classe "Application" dans un bloc "privé".

```
class Application
{
    private:

        // déclarer ici les différentes attributs

    public:
        void lancer();
    private:

        // déclarer ici les différentes méthodes
};
```

### 3.3. Modifications des méthodes

Les attributs étant accessibles par n'importe quelle méthode, les différents paramètres et retours, que nous avons définis, précédemment, pour les différentes méthodes, ne sont plus nécessaires.

Modifier, en conséquence, les déclarations et les définitions des méthodes précédentes et tester le programme.