

## TP 1 : variables, Instructions conditionnelles et Boucles

1. Afficher les 20 premiers nombres pairs avec une boucle while
2. Afficher les 20 premiers nombres impairs avec une boucle while
3. Ecrire un programme qui calcule puis affiche la somme des entiers impairs inférieurs à 100
4. Ecrire un programme qui calcule puis affiche le produit des entiers pairs inférieurs à 100
5. Ecrire un programme qui renvoie le reste de la division euclidienne de n par b. (utiliser uniquement la soustraction)
6. Ecrire un programme qui renvoie le résultat de la multiplication de deux nombre entier n et b. (utiliser uniquement l'addition)
7. Sont l'utilisation de math écrire un programme qui calcule puis affiche  $(x^n - 1) \cdot (x \text{ réel}, n \text{ entier})$
8. Sont l'utilisation de math écrire un programme qui calcule puis affiche  $(x + (-1)^n) \cdot (x \text{ réel}, n \text{ entier})$
9. Sont l'utilisation de math écrire un programme qui calcule puis affiche la factorielle  $N! = 1 * 2 * 3 \dots (N-1) * N$  d'un entier naturel N en respectant que  $0! = 1$ .
10. Donnez le programme qui utilise une boucle for pour afficher les dizaines entre 0 et 100.  
Le résultat affiché sera :  
10 20 30 40 50 60 70 80 90 100
11. Donnez le programme qui utilise une boucle for pour afficher les dizaines entre 100 et 0.  
Le résultat affiché sera :  
100 90 80 70 60 50 40 30 20 10
12. Ecrire un programme qui saisit des entiers positifs. Le programme s'arrête dès qu'un entier négatif est saisi.
13. Ecrire un programme qui saisit des entiers positifs. Le programme s'arrête dès qu'un entier positif est saisi.
14. Ecrire un programme qui saisit des entiers. Le programme s'arrête dès qu'un entier nul est saisi.
15. Ecrire un programme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne

16. Ecrire un programme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivant.( par exemple, si l'utilisateur entre le nombre 20 , le programme affichera les nombres de 21 à 30.
17. Ecrire un programme qui saisit 2 entiers et affiche successivement la somme, la différence, le produit
18. Ecrire un programme qui saisit 3 entiers et affiche successivement la somme et la moyenne
19. Ecrire un programme qui demande à l'utilisateur d'entre la largeur et la longueur et afficher la surface d'un rectangle
20. Ecrire un programme pour lire le PHT, TVA et afficher le P TTC  
Remarque :  $P TTC = PHT * (1 + TVA)$
21. Ecrire un programme qui demande un nombre à l'utilisateur, puis qui calcule et affiche le carré de ce nombre

## TP 2 : Les pointeurs

### Exercice 1 : Affichage des adresses mémoire

Écrire un programme qui déclare trois variables (un entier, un flottant et un caractère). Utilisez des pointeurs pour afficher l'adresse mémoire de chaque variable.

### Exercice 2 : Manipulation de variables via pointeurs

Écrire un programme qui déclare un entier et un flottant. Utilisez des pointeurs pour modifier leurs valeurs et afficher les nouvelles valeurs.

### Exercice 3 : Double pointeur

Écrire un programme qui déclare une variable entière et un double pointeur (pointeur de pointeur). Modifiez la valeur de l'entier en utilisant le double pointeur.

### Exercice 4 : Inversion de deux entiers via pointeur

Écrire un programme qui déclare deux entiers, puis utilise un pointeur pour échanger leurs valeurs.

### Exercice 5 : Calcul de la somme de deux nombres via pointeur

Écrire un programme qui déclare deux entiers, puis utilise des pointeurs pour calculer et afficher leur somme.

### Exercice 6 : Affichage et modification d'un caractère

Écrire un programme qui déclare une variable de type char, utilise un pointeur pour afficher et modifier la valeur de cette variable.

### Exercice 7 : Comparaison d'adresses de deux variables

Écrire un programme qui déclare deux entiers et affiche laquelle des deux variables a l'adresse mémoire la plus élevée.

### Exercice 8 : Pointeur void

Écrire un programme qui déclare une variable entière et une variable flottante. Utilisez un pointeur de type void pour stocker et afficher les valeurs des deux variables.

## TP 3 : Les Fonctions

**Exercice 1 :** Fonction simple de somme

Écrire une fonction qui prend deux entiers en paramètres et retourne leur somme.

**Exercice 2 :** Fonction de calcul du maximum

Écrire une fonction qui retourne le plus grand de deux entiers passés en paramètres.

**Exercice 3 :** Fonction de factorielle

Écrire une fonction récursive qui calcule la factorielle d'un entier donné.

**Exercice 4 :** Fonction pour vérifier si un nombre est pair ou impair

Écrire une fonction qui prend un entier en paramètre et retourne `true` si le nombre est pair et `false` s'il est impair.

**Exercice 5 :** Fonction de permutation de deux nombres

Écrire une fonction qui prend deux références d'entiers et échange leurs valeurs.

**Exercice 6 :** Fonction pour calculer l'exponentiation

Écrire une fonction qui prend deux entiers en paramètres (base et exposant) et retourne la base élevée à la puissance de l'exposant.

**Exercice 7 :** Fonction pour déterminer si un nombre est premier

Écrire une fonction qui vérifie si un nombre donné est un nombre premier.



## TP 5 :

### **Exercice 1 : Passage par valeur : Somme de deux nombres**

Créez une fonction qui prend deux nombres en paramètre par valeur et retourne leur somme.

### **Exercice 2 : Échange de deux nombres**

**Passage par valeur :** Créez une fonction qui prend deux nombres par valeur et essaie d'échanger leurs valeurs (ce qui échouera car c'est un passage par valeur). Affichez les valeurs avant et après l'appel de la fonction.

**Passage par pointeur :** Créez une fonction qui prend deux pointeurs d'entiers et échange les valeurs des deux entiers pointés.

**Passage par référence :** Créez une fonction qui prend deux entiers par référence et échange leurs valeurs.

### **Exercice 3 : Passage par pointeur : Incrémenter une variable**

Écrivez une fonction qui prend un pointeur d'entier et incrémente la valeur pointée.

### **Exercice 4 : Passage par référence : Inverser deux éléments d'un tableau**

Créez une fonction qui prend deux références sur des éléments d'un tableau et les échange.

### **Exercice 5 : Passage de tableau par pointeur : Somme des éléments d'un tableau**

Écrivez une fonction qui prend un tableau et sa taille en paramètre (par pointeur) et retourne la somme des éléments du tableau.

### **Exercice 6 : Passage de tableau par référence : Trouver le minimum**

Créez une fonction qui prend un tableau par référence et retourne la valeur minimale du tableau.

### **Exercice 7 : Passage de tableau de pointeurs : Affichage d'un tableau de chaînes**

Écrivez une fonction qui prend un tableau de pointeurs de chaînes de caractères et affiche chaque chaîne.

### **Exercice 8 : Passage de tableau par référence : Modification d'un tableau**

Écrivez une fonction qui prend un tableau par référence et modifie ses éléments en les multipliant par 2.

## TP 6 : Classe en C++

### Exercice 1 :

- Créez une classe **Rectangle** avec deux attributs : **longueur** et **largeur**.
- Ajoutez des méthodes pour calculer l'aire et le périmètre du rectangle.
- Implémentez le programme avec des fichiers séparés : un fichier d'en-tête (**Rectangle.h**), un fichier source pour la classe (**Rectangle.cpp**), et un fichier principal (**main.cpp**).

### Exercice 2 :

Écrire un programme qui permet de :

Créer une classe de base nommée **Vehicule**. Un **Vehicule** possède trois membres protégés :

- **Annee\_achat** de type **int** ;
- **prix\_achat** et **prix\_courant** de type **double** ;

Et les méthodes publiques :

- **Vehicule(int Annee, double prix)** qui joue le rôle du constructeur de la classe en question.
- **affiche()** permet d'afficher les trois membres : *Annee\_achat, prix\_achat* et *prix\_courant*.
- **calculePrix()** permet de calculer le membre *prix\_courant* d'un **Vehicule** à l'aide de la formule suivante :

$$\text{prix\_courant} = (1 - ((2024 - \text{Annee\_achat}) * 0.01)) * \text{prix\_achat}$$

### Exercice 3 :

Écrire un programme qui permet de :

Créer une classe de base nommée **Personne**. Une **Personne** possède quatre membres protégés :

- **nom** de type **string** ;
- **prenom** de type **string** ;
- **age** de type **int** ;
- **adresse** de type **string** ;

Et les méthodes publiques :

- **Personne(string nom, string prenom, int age, string adresse)** qui joue le rôle du constructeur de la classe en question.
- **affiche()** permet d'afficher les quatre membres : *nom, prenom, age* et *adresse*.
- **saisir()** permet de saisir une personne par le clavier



## TP 4 : Tableaux et Chaînes de caractères

### Exercice 1 : Somme des éléments d'un tableau

Écrivez une fonction qui prend un tableau d'entiers et sa taille en paramètres, puis retourne la somme de tous les éléments du tableau.

*int sommeTableau(int arr[], int taille);*

### Exercice 2 : Moyenne des notes

Créez un programme qui demande à l'utilisateur de saisir les notes de 5 étudiants dans un tableau, puis calcule et affiche la moyenne des notes en utilisant une fonction.

*double moyenneNotes(int notes[], int taille);*

### Exercice 3 : Trouver un élément dans un tableau

Écrivez une fonction qui prend un tableau, sa taille, et un élément à rechercher. Si l'élément est trouvé, la fonction renvoie l'indice de l'élément, sinon elle renvoie -1.

*int chercherElement(int arr[], int taille, int element);*

### Exercice 4 : Longueur d'une chaîne

Écrivez une fonction qui prend une chaîne de caractères en paramètre et renvoie sa longueur (sans utiliser la fonction strlen).

*int longueurChaine(const char\* str);*

### Exercice 5 : Comparaison de chaînes

Créez une fonction qui compare deux chaînes de caractères et renvoie true si elles sont égales, sinon false (sans utiliser la fonction strcmp).

*bool comparerChaines(const char\* str1, const char\* str2);*

### Exercice 6 : Réverser un tableau

Créez une fonction qui prend un tableau d'entiers et renverse son contenu (le premier devient le dernier, etc.). Utilisez cette fonction dans un programme principal pour tester sur un tableau de 10 éléments.

*void inverserTableau(int arr[], int taille);*

### Exercice 7 : Concaténation de deux chaînes

Écrivez une fonction qui prend deux chaînes en paramètres et retourne une nouvelle chaîne qui est la concaténation des deux (sans utiliser strcat).

*char\* concatenerChaines(const char\* str1, const char\* str2);*

### Exercice 8 : Trouver un mot dans une phrase

Créez une fonction qui prend une phrase et un mot en paramètres et renvoie true si le mot existe dans la phrase, sinon false (sans utiliser strstr).

*bool trouverMot(const char\* phrase, const char\* mot);*

### Exercice 9 : Recherche du maximum dans un tableau





16. Ecrire un programme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivant. ( par exemple, si l'utilisateur entre le nombre 20 , le programme affichera les nombres de 21 à 30.
17. Ecrire un programme qui saisit 2 entiers et affiche successivement la somme, la différence, le produit
18. Ecrire un programme qui saisit 3 entiers et affiche successivement la somme et la moyenne
19. Ecrire un programme qui demande à l'utilisateur d'entre la largeur et la longueur et afficher la surface d'un rectangle
20. Ecrire un programme pour lire le PHT, TVA et afficher le PTTC  
Remarque :  $PTTC = PHT * (1 + TVA)$
21. Ecrire un programme qui demande un nombre à l'utilisateur, puis qui calcule et affiche le carré de ce nombre .

de chaînes

qui utilise un tableau de chaînes de caractères pour stocker les prénoms de 5  
e fonction pour afficher tous les prénoms et une autre pour rechercher un prénom