



# IUT TOULON VAR

Département Génie Electrique et Informatique Industrielle

```
#include <stdio.h>
#include <time.h>

void Getdate (char Date[])
{
    time_t temps=time(0);
    struct tm *t1=localtime(&temps);
    strftime(Date, 50, "%d %m %Y %H:%M",t1);
}

void wait (int n)
{
    int Sec;
    Sec=time(0);
    while (difftime (time(0),Sec)<n)
        ;
}

float Temperature (void)
{
    return 20 - ((rand()%100)/5.0) + ((rand()%100) /5.0);
}
```

## Semestre 1

# Algorithmique et langage C

# Sujets des Travaux pratiques

**Nom :**

**Prénom :**

**Groupe :**



## Sujets des TP Première série

**Codage du niveau de difficulté des exercices :**

☀ : *application directe du cours* ☀☀ : *niveau minimum exigé* ☀☀☀☀ : *niveau souhaité*

---

### Le premier programme : "Hello world" ☀

Pour le couple logiciel **PCGRASP/GCC** :, lancer une fenêtre d'édition (new, C file). Y écrire le programme permettant d'afficher « Hello World » à l'écran. Sauvegardez le fichier dans votre répertoire personnel (File, Save as). Compiler et linker (Compiler, compile & link). Corriger les erreurs éventuelles et recompiler.  
Exécuter votre programme (Run, Run captured).

Pour l'environnement DevCPP/GCC :

Créer le programme par : Fichier->Nouveau->Fichier Source

Le sauver avec une extension .c (et pas .cpp comme proposé par défaut)

Compiler par : Exécuter->Compiler

Exécuter par : Exécuter->Exécuter

Attention : La console ("fenêtre MS-DOS") se referme immédiatement après

L'exécution du programme. Pour la laisser ouverte il suffit de rajouter

system("pause"); avant l'instruction return 0 ;

Il suffira d'appuyer sur une touche pour la fermer.

Sous linux avec gcc en ligne de commande : \$gcc -Wall tp1.c -otp1 ; ./tp1

```
#include <stdio.h>
#include <stdlib.h> /* pour la fonction system() (devcpp uniquement) */
int main(void)
{
    printf("Hello World\n");
    system("pause"); // <- pour devcpp uniquement
    return 0 ;
}
```

---

### La surface du cercle ☀

Ecrire un programme qui calcule et affiche la surface d'un cercle dont le rayon (type double) en mètres est entré au clavier.

---

### Quotient et reste ☀

Ecrire un programme qui affiche le quotient et le reste de la division entière de deux nombres entiers entrés au clavier ainsi que le quotient rationnel de ces nombres.

---

### Echanges ☀

Ecrire un programme qui permute et affiche les valeurs de trois variables a, b, c de type entier qui sont entrées au clavier : a ==> b , b ==> c , c ==> a

---

## Résistances ☀

a) Ecrire un programme qui affiche simultanément la résistance équivalente à trois résistances R1, R2, R3 (type **double**) pour les deux cas :

- résistances câblées en série:

$$R_{\text{ser}} = R1 + R2 + R3$$

$$R_{\text{par}} = \frac{R1 \cdot R2 \cdot R3}{R1 \cdot R2 + R1 \cdot R3 + R2 \cdot R3}$$

- les résistances câblées en parallèle:

b) en se servant de 2 variables (perte des valeurs entrées)

---

## TVA ☀☀

a) Ecrire un programme qui calcule le prix TTC (type **double**) d'un article à partir du prix net (type **int**) et du pourcentage de TVA (type **int**) à ajouter. Utilisez la formule suivante en faisant attention aux priorités et aux conversions automatiques de type:

$$PTTC = PNET + PNET \cdot \frac{TVA}{100}$$

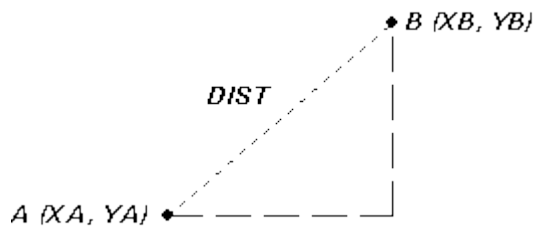
b) Ecrire un programme qui calcule le prix net d'un article (type **double**) à partir du prix TTC (type **double**) et du pourcentage de TVA (type **int**) qui a été ajoutée.

(Déduez la formule du calcul de celle indiquée ci-dessus)

---

## Distance entre deux points ☀☀

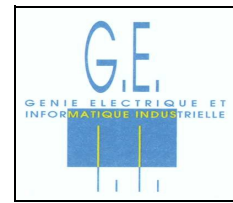
Ecrire un programme qui calcule et affiche la distance DIST (type **double**) entre deux points A et B du plan dont les coordonnées (XA, YA) et (XB, YB) sont entrées au clavier comme entiers.





# Algorithmique et langage C

## TP n°2: L'alternative simple



---

### 1/ Le plus grand ☀

1.1/ Ecrire un programme qui effectue, la saisie de trois entiers et affiche le plus grand des trois nombres.

1.2/ Modifier votre précédent programme de telle sorte qu'il affiche aussi le plus petit.

---

### 2/ Petit, moyen, grand, très grand ☀

Ecrire un programme qui affiche petit, moyen, grand ou très grand selon que le nombre entré est inférieur 10, compris entre 10 et 20, compris entre 20 et 40 ou supérieur à 40.

---

### 3/ La facture de câble électrique ☀

Sachant que le prix hors taxe d'un mètre de câble est 0.19€, écrire le programme permettant d'afficher une facture. La longueur sera saisie au clavier. Pour un montant HT supérieur à 1000€ une remise sur le prix hors taxe de 5% est appliquée. Si le montant est compris entre 500 et 1000€ la remise est de 3%.

Sur la facture devront apparaître :

- Le prix unitaire HT et TTC (hors remise)
- La longueur livrée
- Le prix total HT (tenant compte de la remise)
- La remise
- Le coût de la TVA (19.6%)
- Le prix total TTC

---

### 4/ Résolution d'une équation du second degré ☀☀

Soit l'équation du second degré à coefficients constants :  $Y=ax^2+bx+c=0$

Proposer un programme permettant de résoudre cette équation, a, b et c étant saisis au clavier. Votre programme devra passer en revue toutes les possibilités. On pourra s'inspirer de l'organigramme fourni en annexe.

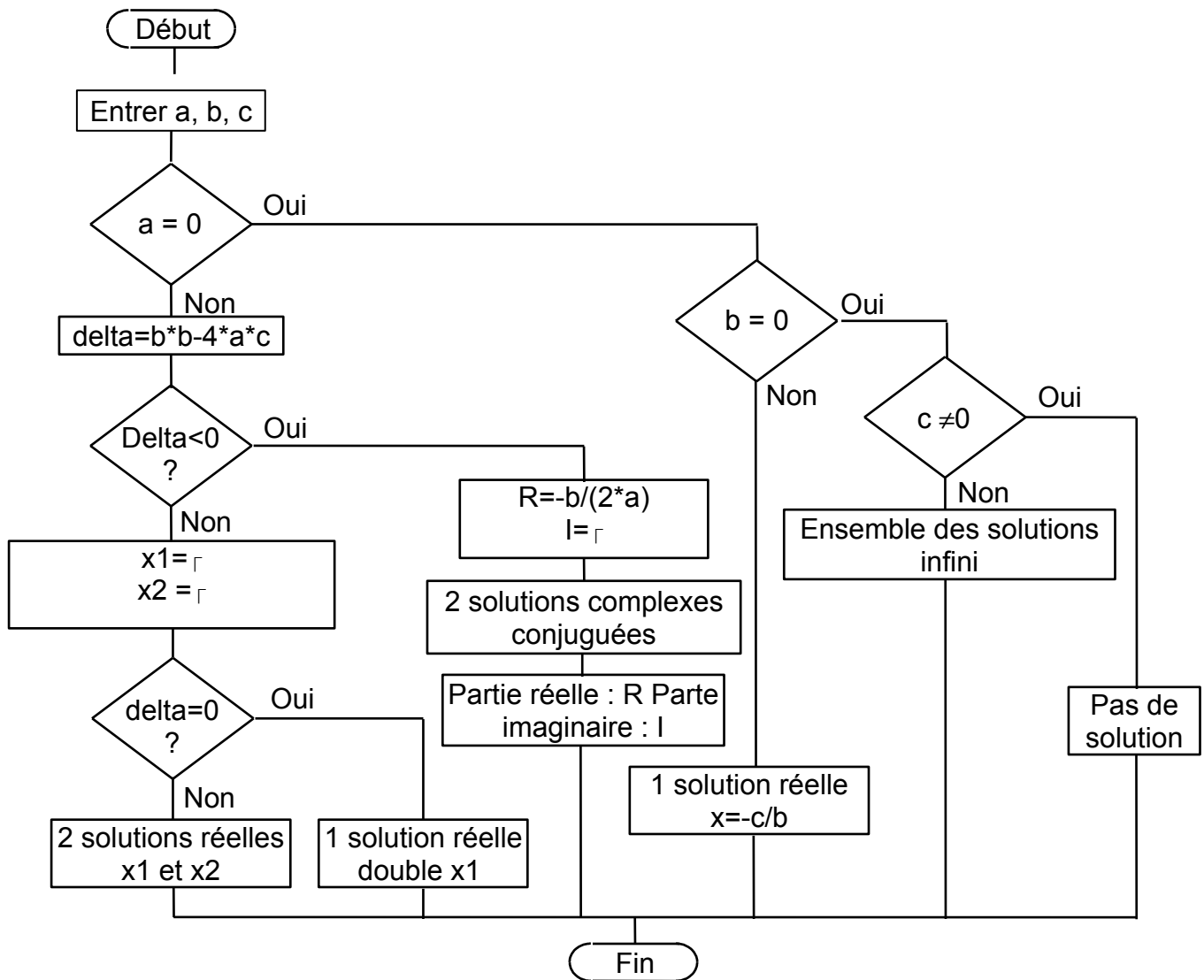


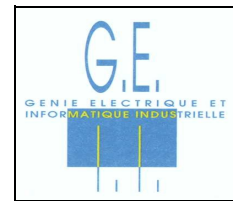
# Algorithmique et langage C

## TP n°2: L'alternative simple



### ANNEXE :





---

## 1/ Une boucle ☀

-Ecrire le programme qui affiche N fois « Boucle » à l'écran. N sera saisi au clavier.

- Ecrire le programme qui affiche la table de multiplication par N entré au clavier.

Par exemple pour N = 3 votre programme affichera :

3 x 0 = 0

3 x 1 = 1

.....

3 x 10 = 30

---

## 2/ La fonction puissance ☀

2.1/ Ecrire le programme qui calcule X à la puissance N, avec N entier positif.

2.2/ Idem 2.1/ avec N entier quelconque.

---

## 3/ Un petit dessin ☀

3.1/ Afficher à l'aide d'une boucle un rectangle à l'écran (le nombre de ligne est saisi au clavier):

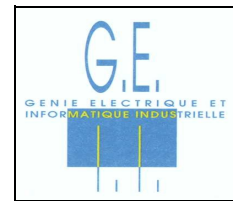
```
*****  
*****  
*****
```

3.2/ Afficher à l'aide d'une boucle ce demi sapin de Noël (le nombre de ligne est saisi au clavier):

```
*  
***  
*****  
*****  
*****
```

3.3/ Afficher à l'aide d'une boucle ce sapin de Noël (le nombre de ligne est saisi au clavier): ☀☀☀

```
*  
***  
*****  
*****  
*****
```



### 4/ Ce nombre est-il premier ? ☀☀

Ecrire un programme qui indique si un nombre entré au clavier est premier. On implémentera l'algorithme simpliste qui consiste à vérifier la divisibilité par les entiers successifs de 2 à la moitié du nombre à tester.

Inspirez vous du programme précédent pour afficher la liste des nombres premiers inférieurs à 1000.

Rappel : Un nombre est premier si il n'est divisible que par lui même ou par 1.

### 5/ Essence ou diesel ? ☀☀

Un loueur de voiture affiche les tarifs suivants :

Véhicule diesel : 30€ par jour + 0.15€/km au delà des 500 premiers kilomètres.  
Véhicule essence : 20€ par jour + 0.35€/km au delà des 300 premiers kilomètres.

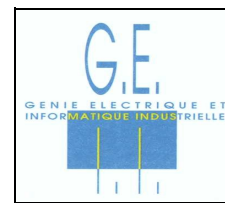
5.1/ Ecrire un programme qui propose le type de véhicule le plus économique à partir de la durée de la location et du nombre de kilomètres parcourus fournis au clavier par l'utilisateur.

5.2/ Reprendre l'exercice 5.1 pour afficher un tableau qui indique la meilleure option en fonction de la durée et de la distance. Un E sera affiché si l'essence est plus économique, un D dans le cas contraire. La distance sera limitée à 900 km par tranche de 100 km. La distance et la durée sont entrées au clavier par l'utilisateur

**Exemple d'exécution :**

Nombre de jour maximum : 15									
Distance maximum : 900									
J/K	200	300	400	500	600	700	800	900	
01	E	E	D	D	D	D	D	D	
02	E	E	D	D	D	D	D	D	
03	E	E	D	D	D	D	D	D	
04	E	E	E	D	D	D	D	D	
05	E	E	E	D	D	D	D	D	
06	E	E	E	D	D	D	D	D	
07	E	E	E	E	D	D	D	D	
08	E	E	E	E	D	D	D	D	
09	E	E	E	E	E	D	D	D	
10	E	E	E	E	E	D	D	D	
11	E	E	E	E	E	E	D	D	
12	E	E	E	E	E	E	D	D	
13	E	E	E	E	E	E	E	D	
14	E	E	E	E	E	E	E	D	
15	E	E	E	E	E	E	E	E	

**Conseils :** Il faut d'abord afficher la première ligne , puis utiliser une boucle pour répéter le calcul de chaque jour. Pour chaque jour il faut, à l'aide d'une boucle imbriquée dans la première, calculer les prix des deux types de véhicule par incrément de 100 km et afficher E ou D. Cet exercice porte sur la notion de boucle imbriquée et ne nécessite pas l'utilisation de tableau en C.



---

### 1/ La somme des N premiers entiers ☀

Ecrire le programme qui calcule avec une boucle la somme des N premiers entiers.

Ex : si N vaut 4, Somme=1+2+3+4

On vérifiera avec la formule suivante : Somme=N\*(N+1)/2

---

### 2/ Le loto Flash ☀

Le principe du loto Flash est un tirage aléatoire de 7 entiers compris entre 1 et 49.

Réaliser le programme effectuant un tirage Flash.

Pour l'instant ce programme ne pourra pas détecter les éventuels doublons dans cette grille.

Conseils :

- La fonction `int rand(void)` ; retourne un entier pseudo aléatoire compris entre 0 et INT\_MAX (le plus grand entier possible dans l'environnement considéré). La fonction `void srand(int)` ; permet d'initialiser le générateur utilisé par `rand()`. Les prototypes se trouvent dans `stdlib.h`.

- Lorsqu'on divise un nombre entier par un nombre entier non nul n, on obtient un reste strictement inférieur à n.

- Une valeur relativement imprévisible au moment du lancement d'un programme est le nombre de secondes écoulées depuis le 1<sup>er</sup> janvier 1970. Ce nombre s est fourni par l'appel `s=time(NULL)` ; le prototype de cette fonction se trouve dans `time.h`.

---

### 3/ Une suite récurrente ☀☀

Ecrire le programme qui calcule le nième terme de la suite de Fibonacci.

Le nième terme de la suite de Fibonacci est donné par la relation de récurrence:

$$U_1=1, U_2=1, U_n = U_{n-1} + U_{n-2} \text{ (pour } n>2)$$

---

### 4/ Un petit développement limité ☀☀☀☀

Le but de cet exercice est de calculer le développement limité de la fonction exponentielle au voisinage de zéro et de comparer la valeur obtenue avec la fonction `exp` de la bibliothèque mathématique.

Développement limité au voisinage de zéro de `exp(x)`:

$$\text{Exp}(x) = 1 + x + x^2/2! + x^3/3! + \dots$$

On pourra améliorer le programme en saisissant l'ordre du développement limité en mode conversationnel

---

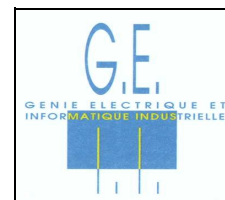
### 5/ Equation du troisième degré ☀☀☀☀

Résolution d'une équation du troisième degré par dichotomie

L'équation  $F(x) = x^3 - 3x^2 + 2x + 1 = 0$  admet au moins une solution entre -1 et +1 .

Ecrire le programme qui calcule cette solution, c'est à dire la valeur de x pour  $f(x) = 0$  à 0.0001 près.





---

### 1/ Recopie d'un tableau ☀

Soit les déclarations suivantes :

```
int Tabint[10]={65,21,-9,23,-1,32,5,-69,12,-3}, TabPos[10]={0} ;
```

Ecrire le programme qui recopie dans *TabPos* les valeurs positives contenues dans *Tabint* et affiche les deux tableaux.

---

### 2/ Calculs sur un tableau ☀

Ecrire le programme permettant de :

- Réaliser la saisie au clavier d'un tableau de 10 entiers positifs. La fin de la saisie s'effectue sur l'occurrence du dixième entier.
- Calculer la moyenne des différentes valeurs contenues dans le tableau.
- D'afficher le plus grand entier contenu dans le tableau.

---

### 3/ Le loto Flash (suite) ☀☀

Le principe du loto Flash est un tirage aléatoire de 7 entiers compris entre 1 et 49.

Réaliser le programme effectuant un tirage Flash.

Pour chaque entier tiré, votre programme vérifiera qu'il ne constitue pas un doublon.

---

### 4/ Le manchot ☀☀

Le but de cette question est de créer une machine à sous **virtuelle** communément appelée « manchot ».

Votre programme devra acquérir la combinaison d'un joueur composée de trois chiffres compris entre 1 et 3 (pomme, banane et poire), puis il effectuera un tirage aléatoire. Le joueur gagnant empochera la somme des mises (**symboliques**) versées par tous les précédents joueurs depuis que le programme du jeu est lancé.

---

### 5/ Tri d'un tableau d'entiers ☀☀☀

On désire mettre en oeuvre l'algorithme du tri à bulles. Ce tri s'effectuant sur un tableau d'entiers.

Cette méthode consiste à comparer les éléments du tableau deux à deux, à partir du bas, et à les permuter si l'élément du bas est inférieur à l'autre. A la fin, le tableau sera donc classé par ordre croissant.



ex: Soit le tableau correspondant à l'initialisation suivante:

`int TableauEntier [5]={10,23,41,3,30};`

10	10	10	10	3	3	3	3	3	3
23	23	23	3	10	10	10	10	10	10
41	41	3	23	23	23	23	23	23	23
3	3	41	41	41	30	30	30	30	30
30	30	30	30	30	41	41	41	41	41

Ce qui donne l'algorithme suivant:

Entier i,j, *TableauEntier* [5]

```
Pour j variant de 1 à N-1
  Pour i variant de N à j
    Si TableauEntier [j] < TableauEntier [i-1]
      Permuter TableauEntier [j] avec TableauEntier [i-1]
    Fsi
  Finpour
Finpour
```

On remarque que cet algorithme peut être amélioré car, si au cours d'un parcours du tableau aucune valeur n'est permutée le tableau est trié, il n'est donc plus nécessaire de le parcourir d'autres fois.