



Algorithmique et langage C  
Sujets des TP  
Première série



**IUT TOULON VAR**

Département Génie Électrique et Informatique Industrielle

**Algorithmique et langage C**  
**Premier Semestre**

**Sujets des Travaux pratiques**  
**Série n° 1**

**Nom :** .....

**Prénom :** .....

**Groupe :** .....

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

☀ : *application directe du cours*

☀☀ : *niveau minimum exigé (à savoir refaire très rapidement)*

☀☀☀ : *niveau souhaité*

☀☀☀☀ : *difficile*

ŠŠŠŠ : *exercice supplémentaire*

## TP1: Outils de développement, Quelques programmes simples.

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

Pour l'environnement **Windows / Dev-CPP/GCC** :

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

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

Compiler par : **Exécuter->Compiler**

Exécuter par : **Exécuter->Exécuter**

Attention : La console ("fenêtre invite de commande") 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.

Pour l'installation voir le document:

[http://arlotto.univ-tln.fr/cours\\_de\\_c/compilateurs/98\\_2000\\_XP\\_vista/Installation\\_Dev-Cpp-1.0.pdf](http://arlotto.univ-tln.fr/cours_de_c/compilateurs/98_2000_XP_vista/Installation_Dev-Cpp-1.0.pdf)

Sous GNU/Linux avec gcc en ligne de commande :

installer gcc : `$ sudo apt-get install build-essential`

puis éditer le source avec gedit par exemple : `$ gedit tp1.c &`

puis sauvegarder et compiler `$ gcc -Wall tp1.c -otpl`

enfin exécuter le programme avec : `$/tpl`

Sous GNU/Linux avec l'environnement graphique geany :

installer geany : `$sudo apt-get install geany`

Dans le menu "Construire->Définir les includes et les options" mettre la ligne `gcc -Wall -lm -o "%e" "%f"`

dans le dialogue **compiler** (identique à construire).

Pour démarrer à faire vos tp sous linux voir le document :

[http://arlotto.univ-tln.fr/cours\\_de\\_c/compilateurs/linux/tpc\\_avec\\_linux.1.2.pdf](http://arlotto.univ-tln.fr/cours_de_c/compilateurs/linux/tpc_avec_linux.1.2.pdf)

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

### La surface du disque ☀

Écrire un programme qui calcule et affiche la surface d'un disque dont le rayon (type double) en mètres est entré au clavier (on prendra  $\pi=3.1415$ ).

Modifier le programme précédant pour utiliser la constante nommée `M_PI` définie dans `math.h`.

Rechercher la définition de cette constante dans le fichier `math.h`.

### Quotient et reste ☀

Écrire 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.

### Échanges ☀

Écrire un programme qui permute et affiche les valeurs de trois variables `a`, `b`, `c` de type entier qui sont entrées au clavier :  $a \rightarrow b$ ,  $b \rightarrow c$ ,  $c \rightarrow a$  Attention, c'est le contenu des variables qui doit être modifié.

La ligne d'affichage doit rester identique avant et après l'échange : `printf("a=%d,b=%d,c=%d\n",a,b,c);`

## TP1: Outils de développement, Quelques programmes simples.

### Résistances ☀

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

- résistances câblées en série:

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

- les résistances câblées en parallèle:

b) Afficher la résistance équivalente série en ne se servant seulement que de *deux variables* pour tout le programme (perte des valeurs entrées).

c) Afficher la résistance équivalente parallèle en ne se servant seulement que de *deux variables* pour tout le programme (perte des valeurs entrées).

### TVA ☀☀

a) Écrire un programme qui calcule le prix TTC (type **float**) 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 types :

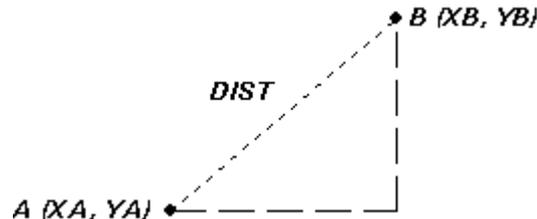
$$PTTC = PNET + PNET \cdot \frac{TVA}{100}$$

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

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

## TP1: Outils de développement, Quelques programmes simples.

### Distance entre deux points ☀☀



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

La racine carrée est obtenu grâce à la fonction c standard `sqrt()` dont le prototype se trouve dans `math.h`.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main(void) {
    float x, y ;
    printf("Entrez un nombre ") ;
    scanf("%f",&y);
    x = sqrt(y) ;
    printf("La racine carré de %f vaut : %f\n",y,x) ;
    return 0 ;
}
```

Rq : Pour gcc il faut rajouter `-lm` dans les options de compilation pour utiliser la librairie mathématique.

```
$gcc -Wall -lm -odistance distance.c
```

### Échange de valeurs numériques avec deux variables ☀☀☀☀

Écrire un programme qui échange la valeur de deux variables de type `int` en utilisant seulement deux variables pour tout le programme. Discuter des avantages/inconvénients de cet algorithme par rapport à l'algorithme classique utilisant une variable supplémentaire.

## TP2 : Variations sur if/else

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

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

### 2/ Le plus grand ☀

1.1/ Écrire 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.

Votre programme ne devra pas utiliser les opérateurs && ni ||. Vérifier que votre programme fonctionne correctement lorsque 2 ou 3 nombres sont égaux.

### 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 livrée 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 :

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

Votre programme devra être factorisé, en particulier il ne faut pas répéter les affichages ni les calculs.

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

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

Proposer un programme permettant de résoudre cette équation, a, b et c étant des valeurs réelles (double) saisies au clavier. Votre programme devra passer en revue toutes les possibilités. On pourra s'inspirer de l'organigramme de la page suivante.

### 5/ Le jour de la semaine ☀☀☀

Écrivez un programme qui calcule le jour de la semaine correspondant à une date donnée, exprimée sous la forme de trois nombres  $j$  (jour),  $m$  (mois),  $a$  (année).

On utilisera la méthode suivante :

$$\text{on pose } m_1 = \begin{cases} m-2 & \text{si } m \geq 3 \\ m+10 & \text{si } m < 3 \end{cases} \quad a_1 = \begin{cases} a & \text{si } m \geq 3 \\ a-1 & \text{si } m < 3 \end{cases} \quad \begin{array}{l} n_s = \text{deux premiers chiffres de } a_1 \\ a_s = \text{deux derniers chiffres de } a_1 \end{array}$$

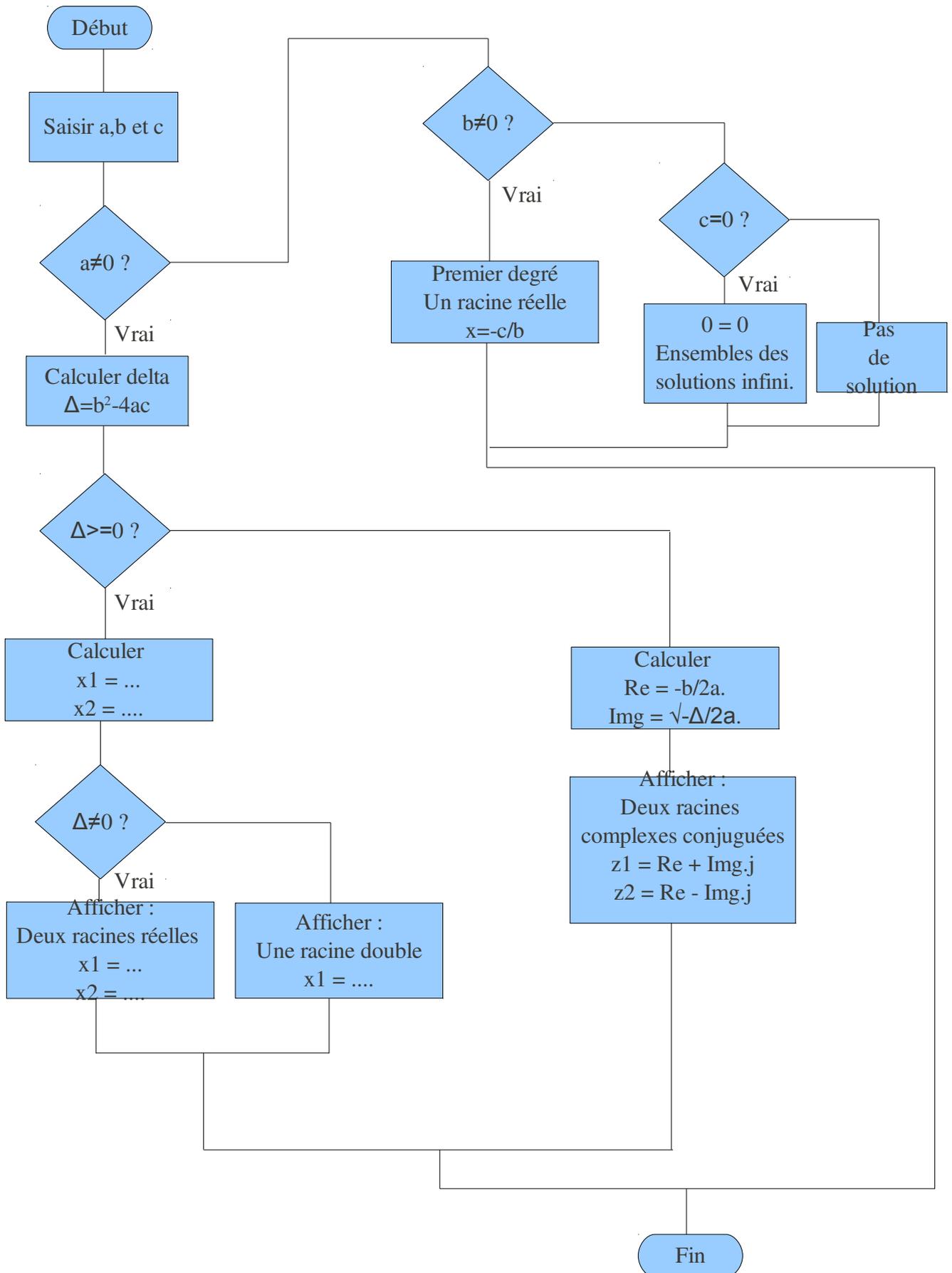
$$f = j + a_s + \frac{a_s}{4} - 2 \cdot n_s + \frac{n_s}{4} + \frac{26 \cdot m_1 - 2}{10} \quad \text{Les barres de fractions indiquent des "quotients par défauts".}$$

Le jour de la semaine est donné par le reste de la division de  $f$  par 7. Avec la convention 0 = dimanche, 1 = lundi, etc... (Attention la valeur de  $f$  peut être négative ☀☀☀☀).

### 6/ Date du Lendemain ☀☀☀☀

Écrivez un programme qui lit la date d'un jour, exprimée sous la forme de trois nombres entiers  $j$  (jour),  $m$  (mois),  $a$  (année) et qui calcule et affiche la date du lendemain. On supposera que la date donnée est correcte.

## TP2 : Variations sur if/else



## TP2 : Variations sur if/else

### 4/ Les Frais d'eBay<sup>®</sup> ☀☀☀☀

Le site d'enchères en ligne eBay<sup>®</sup> se rémunère en facturant ses services au vendeur d'un objet.

A l'aide du principe simplifié de facturation du site eBay<sup>®</sup> ci dessous, on vous demande de créer un programme qui calcule la somme à payer à eBay<sup>®</sup> en fonction du prix de départ et du prix final d'un objet vendu aux enchères.

Pour les annonces Enchères ou à Prix Fixe, les frais eBay<sup>®</sup> se décomposent en 2 parties :

1. Les **frais d'insertion** qui sont dus à la mise en ligne de votre annonce
2. La **commission sur le prix final** qui ne s'applique que si vous trouvez un acquéreur

Frais de base au 9 juillet 2008

Frais d'insertion		Commission sur le prix final	
Prix de départ	Frais d'insertion	Prix final	Commission sur le prix final
		Objet non vendu	Pas de commission
0,01 EUR - 1,00 EUR	0,20 EUR	0 EUR - 50 EUR	<b>5,25%</b> du prix final
1,01 EUR - 9,99 EUR	0,35 EUR	50 EUR - 1000 EUR	<b>3,5%</b> x (prix final - 50 EUR) + 2,63 EUR
10,00 EUR - 24,99 EUR	0,60 EUR	1000 EUR et plus	<b>1,5%</b> x (prix final - 1000 EUR) + 35,88 EUR
25,00 EUR - 49,99 EUR	1,15 EUR		
50,00 EUR - 99,99 EUR	1,80 EUR		
100,00 EUR - 249,99 EUR	2,85 EUR		
250 EUR et plus	3,90 EUR		

## TP3 : Les itérations

### 1/ Une boucle ☀

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

### 2/ les racines ☀

- Écrire un programme qui affiche les racines carrées des nombres entiers entre 0 et 10.

### 3/ les diviseurs ☀

- Écrire un programme qui affiche tous les diviseurs d'un nombre entier entré au clavier.

### 3/ 3 fois 1..3 , 3 fois 2..6 , .... ☀

- Écrire 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
```

### 4/ La fonction puissance ☀

2.1/ Écrire le programme qui calcule x à la puissance n, avec x réel et n entier positif.

2.2/ Idem 2.1/ avec n entier quelconque.

### 5/ Un petit dessin ☀

5.1/ Afficher à l'aide d'une boucle un rectangle à l'écran : le nombre de lignes est saisi au clavier, le nombre de colonnes est fixe (10):

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

5.1bis/ Afficher à l'aide d'une boucle un rectangle à l'écran : les nombres de lignes et de colonnes sont saisis au clavier :

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

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

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

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

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

### 6/ Ce nombre est-il premier ? ☀☀☀☀

-Écrire 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 jusqu'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 et par 1. On peut tester la divisibilité grâce à l'opérateur c %.

## TP3 : Les itérations

### 5/ Résistances en série et en parallèle ☀☀

-Écrire un programme qui calcule la résistance équivalente à un nombre quelconque de résistances câblées en série. La saisie s'arrête et le résultat s'affiche lorsque l'on rentre une valeur négative.

Votre programme doit utiliser seulement deux variables.

Exemple : Tapez valeur de R : 10  
Tapez valeur de R : 20  
Tapez valeur de R : 5  
Tapez valeur de R : -1  
Résultat : 35

- Même problème pour des résistances câblées en parallèle.

### 6/ Essence ou diesel ? ☀☀

Un loueur de voiture affiche les tarifs (fantaisistes) 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/ Écrire 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 maximale et la durée maximale 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.**

## TP4 : Les itérations (suite)

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

Écrire 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/ Un nombre à deviner ☀☀

Le programme tire un nombre entier au hasard entre 1 et 100 et essaye de le faire deviner à l'utilisateur en 10 essais maximum. Après chaque essai l'ordinateur indique "trop grand" ou "trop petit" au joueur ainsi que le nombre d'essais restants. Si au bout de 10 essais, le joueur n'a pas trouvé il a perdu et l'ordinateur lui indique la solution.

Conseil : Pour la mise au point, affichez le nombre tiré un début du programme.

### 4/ Une suite récurrente ŠŠŠŠ

Écrire le programme qui calcule le énième terme de la suite de Fibonacci.

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

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

Les premiers termes sont donc :  $u_1=1, u_2=1, u_3=2, u_4=3, u_5=5, u_6=8, \dots$

### 5/ Un petit développement limité ŠŠŠŠ

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 à l'ordre n au voisinage de zéro de  $e^x$  :  $e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$

### 6/ Calcul de la racine carrée par la méthode de Newton ŠŠŠŠ

La suite définie par  $x_0=1, x_{n+1}=\frac{1}{2}\cdot(x_n+\frac{b}{x_n})$  avec  $b\geq 0$  converge vers la racine carré

de b.

Ecrire un programme qui calcule la racine carré d'un nombre à  $10^{-6}$  près en utilisant cette méthode (appelée méthode de Newton). Travailler avec des variables de type double.

On rappelle que  $|x|$  est donnée par `fabs(x)` dont le prototype se trouve dans `math.h`.