

# RECUEIL D'EXERCICES DE PROGRAMMATION EN LANGAGE C

## 1) LES TYPES DE DONNEES

Type de donnée	Signification	Taille (en octets)	Plage de valeurs acceptée
char	Caractère	1	-128 à 127
unsigned char	Caractère non signé	1	0 à 255
Short int	Entier court	2	-32 768 à 32 767
unsigned short int	Entier court non signé	2	0 à 65 535
int	Entier	2 (sur processeur 16 bits)	-32 768 à 32 767
		4 (sur processeur 32 bits)	-2 147 483 648 à 2 147 483 647
unsigned int	Entier non signé	2 (sur processeur 16 bits)	0 à 65 535
		4 (sur processeur 32 bits)	0 à 4 294 967 295
long int	Entier long	4	-2 147 483 648 à 2 147 483 647
unsigned long int	Entier long non signé	4	0 à 4 294 967 295
float	Flottant (réel)	4	$-3.4 \cdot 10^{-38}$ à $3.4 \cdot 10^{38}$
double	Flottant double	8	$-1.7 \cdot 10^{-308}$ à $1.7 \cdot 10^{308}$
Long double	Flottant double long	10	$-3.4 \cdot 10^{-4932}$ à $3.4 \cdot 10^{4932}$
bool	Booléen	Même taille que le type <i>int</i> , parfois 1 sur quelques compilateurs	Prend deux valeurs : ' <i>true</i> ' et ' <i>false</i> ' mais une conversion implicite (valant 0 ou 1) est faite par le compilateur lorsque l'on affecte un entier (en réalité toute autre valeur que 0 est considérée comme égale à <i>true</i> ).

## 2) STRUCTURE D'UN PROGRAMME, VARIABLES, TYPES, OPERATEURS, ENTREES/SORTIES DE BASE

### EXERCICE 1:

Analyser le programme ci-dessous et écrivez précisément le résultat obtenu par son exécution pour chaque partie.

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

```

#include <stdlib.h>

int main (void)
{
    int i, j, k;
    float a, b, c;
    double p, q, r;

    printf("\n\n partie 1\n");
    i = 5;
    j = 25;
    k = 25%5;
    a = M_PI;          // le nombre pi
    r = 5.6;
    q = 2.3;
    p = q = r;
    printf("i=%d, j=%d, k=%d\n",i, j, k);
    printf("a=%f, b=%f, c=%f\n",a , b, c);
    printf("p=%e, q=%e, r=%e\n",p, q, r);
    system("pause");

    printf("\n\n partie 2\n");
    p = M_PI;          // le nombre pi
    q = p; a = p; i = p;
    b = p - a;
    r = p - q;
    c = a - p;
    //printf("i=%d, j=%d, k=%d\n",i, j, k);
    printf("b=%f, r=%f, c=%f, i=%d\n",b , r, c, i);
    printf("b=%e, r=%e, c=%e, i=%e\n",b , r, c, i);
    system("pause");

    printf("\n\n partie 3\n");
    p = M_PI;          // le nombre pi
    i = 10;
    j = i/3*3;
    k = i*3/3;
    printf("i=%d, j=%d, k=%d\n",i, j, k);
    system("pause");

    printf("\n\n partie 4\n");
    p = M_PI;          // le nombre pi
    i=10;
    a = i/3;
    b = a*3;
    c = i/a;
    printf("i=%d\n",i);
    printf("a=%f, b=%f, c=%f\n",a, b, c);
    system("pause");

    printf("\n\n partie 5\n");
    p = M_PI;          // le nombre pi
    i = 10;
    j = 0;
    printf("p=%f, i=%d, j=%d\n",p, j, i);
    system("pause");

    printf("\n\n partie 6\n");
    p = M_PI;          // le nombre pi
    q = sqrt(p)*sqrt(p);
    r = p - q;

```

```

a = p;
b = sqrt(a)*sqrt(a);
c = a - b;
printf("p=%f, q=%f, r=%f\n",p, q, r);
printf("a=%f, b=%f, c=%f\n",a, b, c);
printf("p=%e, q=%e, r=%e\n",p, q, r);
printf("a=%e, b=%e, c=%e\n",a, b, c);
system("pause");

printf("\n\n partie 7\n");
p = M_PI;          // le nombre pi
i = 14;
printf("i=%d",i);
printf("%f\n",p);
system("pause");

printf("\n\n partie 8\n");
p = M_PI*1.e3;
i = p;
j = i*i;
k = j*j;
printf("i=%d, j=%d, k=%d\n",i, j, k);
system("pause");

printf("\n\n partie 9\n");
printf("p=");scanf("%lf",&p);
printf("p=%e\n",p);

printf("q=");scanf("%f",&q);
printf("q=%e\n",q);

printf("r=");scanf("%lf",r);
printf("r=%e\n",r);
system("pause");

system("pause");
return 0;
}

```

### 3) CALCULS ET TESTS LOGIQUES

#### EXERCICE 2:

Le problème de la machine qui « rend la monnaie ». Ecrivez un programme qui, pour une somme donnée en euros, sans centimes, imprime le nombre (minimal) de billets nécessaires pour la composer.

Exemple :

$$1949 = 3 \times 500 + 2 \times 200 + 2 \times 20 + 1 \times 5 + 2 \times 2$$

(Indication :  $a$  et  $b$  étant des expressions entières, le quotient et le reste de la division entière de  $a$  par  $b$  s'obtiennent respectivement par les expressions  $a / b$  et  $a \% b$ ).

#### EXERCICE 3:

Ecrivez un programme qui lit au clavier un prix en euros *avec des centimes* puis l'affiche :

- a) arrondi à l'euro inférieur ;
- b) arrondi à l'euro le plus proche ;
- c) arrondi au quart d'euro le plus proche.

#### EXERCICE 4:

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

#### EXERCICE 5:

Ecrivez 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 formule suivante.

On pose:

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

puis

$$f = j + a_s + \frac{a_s}{4} - 2n_s + \frac{n_s}{4} + \frac{26m_1 - 2}{10}$$

(les barres de fraction indiquent des « quotients par défaut »). Dans ces conditions, le jour de la semaine est donné par le reste de la division de  $f$  par 7 (0 : dimanche, 1 : lundi, .etc.).

Faites attention, il peut arriver que la valeur de  $f$  soit négative.

N.B. La formule précédente ne tient pas compte du fait que la veille du vendredi 15 octobre 1582, jour de l'entrée en vigueur de la réforme de Grégoire XIII, était le jeudi 4 octobre 1582.

## 4) REPETITION, BOUCLES

#### EXERCICE 6:

Ecrivez un programme qui permet de rentrer des valeurs de résistances électriques réelles et qui calcul la résistance équivalente en **série** au fur et à mesure de la saisie. Le programme s'arrête lorsque que l'on entre une valeur négative (non prise en compte).

Résultat désiré:

```
valeur de R1:100
La resistance equivalente est: 100.000000
valeur de R2:200
La resistance equivalente est: 300.000000
valeur de R3:1000
La resistance equivalente est: 1300.000000
valeur de R4:500
La resistance equivalente est: 1800.000000
valeur de R5:-1
Appuyez sur une touche pour continuer...
```

#### EXERCICE 7:

Ecrivez un programme qui permet de rentrer des valeurs numériques positives et qui affiche le maximum au fur et à mesure de la saisie. Comme dans l'exercice précédent, le programme s'arrête lorsque que l'on entre une valeur négative (non prise en compte).

7A / Résultat désiré:

```
valeur? :10
```

```
Le maximum est 10.000000
valeur? :8
Le maximum est 10.000000
valeur? :152
Le maximum est 152.000000
valeur? :30
Le maximum est 152.000000
valeur? :-1
Le maximum est 152.000000
Appuyez sur une touche pour continuer...
```

**7B/ autre variante:** si le nombre que l'on vient d'entrer n'est pas le maximum lui même, le programme rajoute la remarque "toujours" dans sa réponse.

Exemple:

```
valeur? :10
Le maximum est 10.000000
valeur? :200
Le maximum est 200.000000
valeur? :100
Le maximum est toujours 200.000000
valeur? :3
Le maximum est toujours 200.000000
valeur? :500
Le maximum est 500.000000
valeur? :6
Le maximum est toujours 500.000000
valeur? :-1
Le maximum est toujours 500.000000
Appuyez sur une touche pour continuer...
```

## EXERCICE 8

Ecrivez un programme qui dessine les motifs constitués d'étoiles (caractère "\*\*") avec une seule boucle:

**Un rectangle:** le nombre de lignes et le nombre de colonnes sont demandés à l'utilisateur.

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

**Un triangle:** le nombre de lignes est demandé à l'utilisateur.

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

**Un sapin:** le nombre de lignes est demandé à l'utilisateur.

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

## EXERCICE 9

Ecrivez un programme qui détermine le nombre pi en calculant la racine de la fonction cosinus dans l'intervalle [0, 2] par la méthode de dichotomie.

## EXERCICE 10

Quel est le temps d'exécution précis nécessaire à la machine pour trouver la racine ci-dessus par la méthode de dichotomie. Ecrivez un programme pour le calculer.

### EXERCICE 11

Ecrire un programme qui affiche des nombres aléatoires :

- entiers compris entre 0 et 1000
- entiers compris entre 1000 et 2000
- réels compris entre 0 et 1
- réels compris entre 0.5 et 0.6

### EXERCICE 12

Ecrire les différentes parties d'un programme qui utilise deux tableaux, t et p, de N éléments réels (double) chacun et permet de:

1. saisir au clavier les N éléments du premier tableau t
2. afficher le premier tableau t (ses N éléments)
3. transférer les N éléments du premier tableau, t, dans le second, p
4. afficher le deuxième tableau, p
5. comparaison des deux tableaux t et p et afficher soit "Les tableaux ne sont pas identiques!" ou "Les tableaux sont identiques!".

### EXERCICE 13

Ecrire un programme qui permet de

1. entrer les coefficients d'un polynôme de degrés quelconque (demandé et  $<10$ ) dans un tableau
2. d'afficher ce polynôme sous une forme canonique
3. calculer le polynôme dérivé
4. d'afficher le polynôme dérivé

exemple:

```
degres du polynome (<10) :4
coefficient de degres 4:5
coefficient de degres 3:2
coefficient de degres 2:-1
coefficient de degres 1:3
coefficient de degres 0:4
polynome p:
( 5.00)*x^4 + ( 2.00)*x^3 + (-1.00)*x^2 + ( 3.00)*x^1 + ( 4.00)*x^0
polynome derive:
(20.00)*x^3 + ( 6.00)*x^2 + (-2.00)*x^1 + ( 3.00)*x^0
Appuyez sur une touche pour continuer...
```

### EXERCICE 14

Ecrire un programme qui affiche la table des codes ASCII compris entre 14 et 255 sous la forme:

```
14  ␣
15  ✖
16  ►
17  ◀
18  ↑
19  !!
20  ℔
21  $
...
250  .
251  ¹
252  ³
253  ²
254  ■
255
```

Même question mais sous la forme:

```

14  🎵
15  ✨ 16  ▶ 17  ◀ 18  † 19  !! 20  ℔ 21  $ 22  — 23  ‡ 24  ↑ 25  ↓ 26  → 27  ← 28  L 29  ↔
30  ▲ 31  ▼ 32  33  ! 34  " 35  # 36  $ 37  % 38  & 39  ' 40  ( 41  ) 42  * 43  + 44  ,
45  - 46  . 47  / 48  0 49  1 50  2 51  3 52  4 53  5 54  6 55  7 56  8 57  9 58  : ...etc...

```

Même question, mais sous la forme:

```

14  🎵 0e
15  ✨ 0f
16  ▶ 10
17  ◀ 11
18  † 12
19  !! 13
20  ℔ 14
21  $ 15
22  — 16
23  ‡ 17
...
...

252  ³  fc
253  ²  fd
254  ■  fe
255     ff

```

### EXERCICE 15

Ecrire un programme qui précise pour chaque caractères entrés au clavier si c'est une lettre en majuscule ou minuscule, un chiffre, un autre :

Entrez des caracteres suivi de Entree, (\* pour terminer).

```

A-> Lettre Majuscule
b-> Lettre Minuscule
m-> Lettre Minuscule
6-> Chiffre
*-> Autre
...

```

### EXERCICE 16

Ecrire un programme qui convertit les caractères entrés au clavier en majuscule (les caractères non alphabétiques ne sont pas transformés):

Entrez des caracteres suivi de Entree, (\* pour terminer).

```

A->A
b->B
m->M
*->*
...

```

### EXERCICE 17

Ecrire un programme qui affiche tous les chiffres trouvés dans une chaîne de caractères.

Entrez une phrase:

Nous partîmes 300, mais par un prompt renfort, nous nous vîmes 10 000, en arrivant au port.

30010000

Écrire un programme qui vérifie si une chaîne de caractère est convertible en entier (tous les caractères sont numériques).

conversion possible

## en 1255

```
en 125.360000
```

$3 = 00000000000000000000000000000000000011$

```
comparaison: -1
```

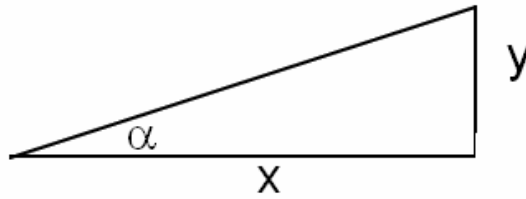
```
resultat: abcdef
```

Bonjour Monsieur Dupont

Écrire une fonction qui renvoie la taille d'une chaîne de caractère passée en paramètre. Utiliser cette fonction dans un programme principal.

### EXERCICE 27

Ecrire une fonction qui calcul l'angle  $\alpha$  (en radians) dans le triangle rectangle ci-dessous. Les valeurs  $x$  et  $y$  seront passées en paramètres. Nous utiliserons pour cela la fonction tangente de la bibliothèque mathématique du langage C (prototype : `double tan (double)`)



### EXERCICE 28

Ecrire une fonction qui permet de tester si un entier passé en argument est compris dans un intervalle passé en argument également. La fonction reverra 1 s'il est dans l'intervalle et 0 sinon.

### EXERCICE 29

- Ecrire une fonction qui retourne la valeur entière d'une chaîne de caractères représentant un nombre binaire.
- Idem pour une chaîne qui représente un nombre en hexadécimal.

### EXERCICE 30

Ecrire un programme qui demande un nom sans extension pour un fichier, rajoute l'extension ".txt", et crée ce fichier. Ensuite le programme écrira des nombres aléatoires compris entre -5 et 5 dans ce fichier.

### EXERCICE 31

Ecrire un programme qui duplique un fichier. Le nom de la copie sera formé à partir du nom du premier auquel on ajoutera 2 avant l'extension. "toto.c" -> "toto2.c"

### EXERCICE 32

- Ecrire un programme qui compte le nombre d'occurrence du caractère 'e' dans un fichier.
- Modifier le programme pour compter le nombre d'occurrence de chaque caractère et stocker les résultats dans un fichier.

### EXERCICE 33

Ecrire un programme qui récupère l'heure, la latitude et la longitude d'une trame GPS de type:

`$GPGGA,160610,4307.9898,N,00600.3392,E,1,07,01.4,0152,M,,M,000,0000*7B`