

## TP1 – Programmation en C

Temps prévu 3h

### Exercice 1 :

Soient les déclarations:

```
long  A = 15;
char  B = 'A';
short C = 10;
```

Quels sont le type et la valeur de chacune des expressions:

- |     |                          |        |                |                     |
|-----|--------------------------|--------|----------------|---------------------|
| (1) | $C + 3$                  | 13     | short (entier) |                     |
| (2) | $B + 1$                  | 66     | int (entier)   |                     |
| (3) | $C + B$                  | 75     | int (entier)   |                     |
| (4) | $3 * C + 2 * B$          | 160    | int (entier)   |                     |
| (5) | $2 * B + (A + 10) / C$   | 132.05 | double (réel)  | double (réel)<br>Le |
| (6) | $2 * B + (A + 10.0) / C$ | 132.05 | double (réel)  |                     |

### Exercice 2 :

Écrire un programme C qui calcule  $n!$  (factoriel) une fois avec un **while** puis une seconde fois avec un **for**.

### Exercice 3 :

Écrire un programme qui calcule les racines carrées de nombre fournis en donnée. Il s'arrêtera lorsqu'on lui fournira la valeur 0. Il refusera les valeurs négatives. Son exécution se présentera ainsi :

Donnez un nombre positif : 2

Sa racine carrée est : 1.414214...

Donnez un nombre positif : -1

Le nombre doit être positif

Donnez un nombre positif : 5

Sa racine carrée est : 2.2360...

Donnez un nombre positif : 0

### Exercice 4 :

Écrivez un programme qui lit la date du clavier et écrit les données ainsi que le nombre de données correctement reçues sur l'écran.

Exemple :

Introduisez la date (jour mois année) : 27 09 2020

Données reçues : 3

Jour : 27

Mois : 09

Année : 2020

**Exercice 5 :**

Essayez le programme suivant et modifiez-le de façon à ce qu'il affiche:

- $A^B$ ,
- la tangente de A en n'utilisant que les fonctions sin et cos,
- la valeur arrondie (en moins) de A/B, **3**
- la valeur arrondie (en moins) à trois positions derrière la virgule de A/B.

```
#include <stdio.h>
main()
{
    double A;
    double B;
    double RES;
    /* Saisie de A et B */
    printf("Introduire la valeur pour A : ");
    scanf("%lf", &A);
    printf("Introduire la valeur pour B : ");
    scanf("%lf", &B);
    /* Calcul */
    RES = A*A;
    /* Affichage du résultat */
    printf("Le carré de A est %f \n", RES);
    /* Calcul */
    RES = B*B;
    /* Affichage du résultat */
    printf("Le carré de B est %f \n", RES);
    return 0;
}
```

gD, E

pow  
C  
D  
↑  
ty

floor

C, D

9/10 = 3,000

( \* 100 ) / 100

#include &lt;math.h&gt;

**Type des données**

Les arguments et les résultats des fonctions arithmétiques sont du type **double**.

**Fonctions arithmétiques**

COMMANDE C	EXPLICATION	LANG. ALGORITHMIQUE
exp (X)	fonction exponentielle	$e^x$
log (X)	logarithme naturel	$\ln(X)$ , $X > 0$
log10 (X)	logarithme à base 10	$\log_{10}(X)$ , $X > 0$
pow (X, Y)	X exposant Y	$X^Y$
sqrt (X)	racine carrée de X	pour $X > 0$
fabs (X)	valeur absolue de X	$ X $
floor (X)	arrondir en moins	int (X)
ceil (X)	arrondir en plus	
fmod (X, Y)	reste rationnel de X/Y (même signe que X) pour X différent de 0	
sin (X) cos (X) tan (X)	sinus, cosinus, tangente de X	
asin (X) acos (X) atan (X)	arcsin(X), arccos(X), arctan(X)	
sinh (X) cosh (X) tanh (X)	sinus, cosinus, tangente hyperboliques de X	

## TP 2 – Programmation en C

Temps prévu 5 h

### Exercice 1 :

Ecrire un programme qui détermine la plus grande et la plus petite valeur dans un tableau d'entiers A. Afficher ensuite la valeur et la position du maximum et du minimum. Si le tableau contient plusieurs maxima ou minima, le programme retiendra la position du premier maximum ou minimum rencontré.

### Exercice 2 :

Un tableau A de dimension N+1 contient N valeurs entières triées par ordre croissant; la (N+1)ième valeur est indéfinie. Insérer une valeur VAL donnée au clavier dans le tableau A de manière à obtenir un tableau de N+1 valeurs triées.

### Exercice 3 :

On vous donne un tableau non trié T de n entiers. Ecrire un programme en C résolvant chacune des tâches suivantes ci-dessous avec les meilleurs algorithmes possibles auxquels vous pouvez penser.

1. Déterminez si T (**non trié**) contient une ou plusieurs paires d'entiers en double.  
Vous pouvez s'appuyer sur le calcul du nombre d'occurrence des éléments du tableau.
2. Étant donné un entier v, trouvez deux entiers a,b dans T tels que  $a + b = v$ .
3. Suivi de la question 2 : maintenant **si le tableau T donné est déjà trié** ?
4. Affichez les nombres entiers dans T compris entre une plage donnée [a . . . b] (inclus) dans l'ordre trié.

**Exemple :** soit T un tableau non trié contenant deux paires en double (3,3) et (2,2)

3	3	2	3	3	2	4	5	3	6	1	4	2	5	10	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	---



**Exercice 4 :**

Un carré magique d'ordre  $n$  est une matrice carrée  $n \times n$  telle que la somme des entiers de chaque ligne, chaque colonne et les deux diagonales est identique.

Par exemple :

Un carré magique d'ordre 3

8	1	6
3	5	7
4	9	2

Ecrire un programme qui teste si une matrice carrée représente un carré magique.

**Exercice 5 :**

Une matrice carrée est dite balancée si les sommes des éléments de ses quatre triangles isocèles sont égales et les sommes des éléments sur la première et la seconde diagonale sont nulles.

Ecrire un programme qui permet de vérifier si une matrice carrée est balancée.

1	2	-1	-2	3
1	2	3	-1	2
-4	0	-2	-1	0
5	1	4	-7	1
-1	-2	-1	1	6

**TP 3 - Programmation en C**

Temps prévu 6h

**Exercice 1 : Tableau de chaînes de caractères**

Écrire un programme C qui lit les noms des jours de la semaine et les mémorise dans un tableau de 10 chaînes de caractères nommé `Jour`.

Afficher ensuite, le nom du jour de la semaine correspondant au numéro de jour (à partir de 1 et jusqu'à 7).

Exemple :

"lundi" pour 1

"mardi" pour 2

...

"dimanche" pour 7

**Exercice 2 :**

Écrire un programme qui lit deux chaînes de caractères et copie la première à la suite de la seconde.

On utilisera la valeur de retour pour indiquer si la concaténation a été possible. Si la place est insuffisante, on copie le maximum de caractères possibles. Le tableau obtenu doit toujours être une chaîne de caractères.

**Exercice 3 :**

Écrire un programme qui vérifie si une chaîne de caractères est un palindrome (lisible dans les deux sens, "Esope reste ici et se repose").

**Exercice 4 :**

On appelle Anagramme deux mots écrits avec les mêmes lettres (tsarine et entrais).

Écrire un programme permettant de déterminer si deux chaînes de caractères sont des anagrammes.

Indication : on pourra utiliser un tableau de 256 (des entiers) permettant de mémoriser le nombre d'occurrences de chaque lettre.



**Exercice 5:**

Ecrire un programme qui recherche un mot dans une chaîne de caractères. Il affichera

la position du début du mot (et si le mot est absent, l'affichera le mot est inexistant).  
 Exemple : si la chaîne contient "bonjour et je", et si l'on recherche "je", le programme affichera :  
 Le mot commence à la position 3.

**Exercice 6:**

Écrire un programme qui compte le nombre de chacune des lettres de l'alphabet (majuscules, minuscules, chiffres, etc.).

Le programme devra recevoir un nombre quelconque de lignes. L'utilisateur tapera une "ligne vide" pour signaler qu'il a terminé la frappe de son texte (ce qui revient à dire qu'il frappera donc deux fois de suite la touche "Entrée", après la frappe de sa dernière ligne).

On supposera que les lignes frappées au clavier ne peuvent jamais dépasser 127 caractères.

En fonction du temps disponible :

Chercher le nombre des mots et des lignes dans le texte.

Ecrire un

**TP 3 – Programmation en C**

Temps prévu 6h

**Exercice 1 : Tableau de chaînes de caractères**

Ecrire un programme C qui lit les noms des jours de la semaine et les mémorise dans un tableau de chaînes de caractères nommé *Jour*.

Afficher, par la suite, le nom du jour de la semaine correspondant au numéro de jour lu entre 1 et 7.

Exemple :

"lundi" pour 1

"mardi" pour 2

...

...

"dimanche" pour 7

**Exercice 2 :**

Ecrire un programme qui lit deux chaînes de caractères et copie la première à la suite de la seconde.

On utilisera la valeur de retour pour indiquer si la concaténation a été possible. Si la place est insuffisante, on copie le maximum de caractères possibles. Le tableau obtenu doit toujours être une chaîne de caractères.

**Exercice 3 :**

Ecrire un programme qui vérifie si une chaîne de caractères est un palindrome (lisible dans les deux sens, "Esope reste ici et se repose").

**Exercice 4 :**

On appelle Anagramme deux mots écrits avec les mêmes lettres (tsarine et entrais).

Ecrire un programme permettant de déterminer si deux chaînes de caractères sont des anagrammes.

Indication : on pourra utiliser un tableau de 256 (des entiers) permettant de mémoriser le nombre d'occurrences de chaque lettre.



**Exercice 5:**

Ecrire un programme qui recherche un mot dans une chaîne de caractères. Il affichera

la position du début du mot (Si le mot est absent, il affiche 0). Le mot recherché est

le mot "le mot inexistant")

et si des chaînes contiennent "banjér" et "jeur".

Il affiche « le mot commence à la position »

"banjér" et "jeur".

affiche le mot commence à la p

Réponse : un programme qui compte la position de chaque caractère (majuscule, minuscule, ponctuation, chiffres, etc.)

Le programme devra accepter un nombre quelconque de lignes. L'utilisateur sera une

"ligne vide" pour signaler qu'il a terminé la frappe de son texte (ce qui revient à dire

qu'il frappera donc deux fois de suite la touche "Entrée", après la frappe de sa dernière

ligne).

On supposera que les lignes frappées au clavier ne peuvent jamais dépasser 127

caractères.

En fonction du temps disponible :

Chercher le nombre des mots et des lignes dans le texte.



## TP 4 – Programmation C

### Tableaux – Structures de données - Fonctions

Temps prévu (3h)

#### Exercice 1 :

Soit le modèle (type) de structure suivante :

```
struct point
{
    char c;
    int x,y;
}
```

1. Afficher le contenu d'une variable de type *struct point* sous la forme :  
**point B de coordonnées 10 12**
2. Écrire la déclaration d'un tableau (nommé *courbe*) de NP points (NP supposé défini par une instruction (*#define*)).
3. Afficher les valeurs des différents "points" du tableau *courbe*, après sont remplissage, sous la forme précédente de 1.
4. Créer une structure *pixel* dont les champs sont les suivants :
  - a. *p* de type *struct point*
  - b. *coul* de type *char*
5. Écrire la déclaration d'un tableau (nommé *image*) de N *pixel* (N supposé défini par une instruction (*#define*)). Remplir et afficher le tableau *image*.

#### Exercice 2 : facturation par code

Réaliser un programme établissant une facture pouvant porter sur plusieurs articles. Pour chaque article à facturer, l'utilisateur ne fournira que la quantité et un numéro de code à partir duquel le programme devra retrouver à la fois le libellé et le prix unitaire.

Le programme devra refuser les codes inexistants. A la fin, il affichera un récapitulatif tenant lieu de facture.

Les informations relatives aux différents articles seront définies dans la source même du programme. Elles seront toutefois placées à un niveau global, de manière à pouvoir, le cas échéant, faire l'objet d'un source séparé, appelable par *#include*.

On prévoira deux fonctions :

- Une pour rechercher les informations relatives à un article, à partir de son numéro de code,
- Une pour afficher la facture récapitulative.

**Exemple**

Combien d'articles à facturer ? 3

Code article ? 25

Quantité de produitX au prix unitaire de 370 ? 33

Code article ? 7 ?

\*\* article inexsistant – redonnez le code : 16

Quantité de produitY au prix unitaire de 199,50 ? 12

Code article ? 26

Quantité de produitZ au prix unitaire de 295,25 ? 6

**Facture**

Article	Nbre	PU	Montant
ProduitX	33	370	12210
ProduitY	12	199,50	2394
ProduitZ	6	295,25	1771,50



# TP N° 5

## Exercice 1 :

Ecrire un programme de cryptage (de chiffrement) "Rot 13". C'est un algorithme de chiffrement de texte. Il s'agit d'un décalage de 13 caractères de chaque lettre du texte à chiffrer. Entre autre c'est une rotation des lettres de 13 positions, c'est-à-dire qu'après la lettre Z on a la lettre A.

Exp. : correspondance entre les caractères en clair et chiffrés :

Caractère en clair	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Caractère chiffré	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M

On supposera que le texte ne contiendra pas des chiffres, des ponctuations, et des symboles.

On y prévoira :

- Une fonction pour la lecture du texte à chiffrer se fait par l'entrée standard
- Une fonction de chiffrement du texte.
- Une fonction de déchiffrement du texte
- Exercice 2

On veut écrire un programme qui applique le tri de sélection sur un tableau d'entiers à l'aide des fonctions.

- 1 - Ecrire la fonction **REPLIR** qui permet le remplissage d'un tableau d'entiers.
  - 3 - Ecrire la fonction **INSERER** qui permet d'insérer un entier dans un tableau déjà trié.
  - 2 - Ecrire la fonction **TRI\_INSERTION** qui permet de trier par ordre croissant les éléments d'un tableau à N entiers donné comme argument en utilisant la fonction **INSERER**.
- Ecrire la fonction **FUSION** qui construit un tableau trié par ordre croissant avec les éléments de deux tableaux A et B triés eux aussi par ordre croissant.
  - Ecrire la fonction **AFFICHER** qui permet l'affichage du tableau trié.
  - Ecrire un programme pour tester toutes les fonctions créées.

**NB:**

- Utiliser les pointeurs.
- Trier le tableau de gauche à droite en insérant à chaque fois l'élément  $I+1$  dans le tableau (déjà trié) des  $I$  premiers éléments.

