



**Yantra Technologies**

[www.yantra-technologies.com](http://www.yantra-technologies.com)

# Le Langage C

## Travaux Pratiques

[carole.grondein@yantra-technologies.com](mailto:carole.grondein@yantra-technologies.com)  
[david.palermo@yantra-technologies.com](mailto:david.palermo@yantra-technologies.com)

- Ecrire un programme C permettant de créer 2 variables
  - 2 caractères initialisés à 65 et 'A'
- Afficher ces variables sous forme d'entiers et de caractères

### Ecrire un programme C

- déclarant 2 variables entières
- affichant leur somme multipliée par 2.

Ecrire un programme qui divise et afficher 2 reel

Exemple:

\$> Division

Entrer 2 reel: 5.1 2.0

Résultat 2.55

\$>

Faire la même chose avec des entiers

## TP 5 : Saisie de nombres et de caractères

Saisir un caractère au clavier avec scanf. Afficher son code ASCII à l'écran.

```
$> codeAscsci
```

```
Voulez-vous entrer une lettre(l) ou un entier(e) ? l
```

```
Taper la lettre : A
```

```
Caractere = A code = 65
```

```
Voulez-vous continuer (O,o,Y,y) ?o
```

```
Voulez-vous entrer une lettre(l) ou un entier(e) ? e
```

```
Taper un entier : 65
```

```
Caractere = A code = 65
```

```
Voulez-vous continuer (O,o,Y,y) ?n
```

```
A bientôt
```

```
$>
```

- Ecrire un programme C permettant de poser la question ' voulez-vous jouer ? '.
- Il y aura saisie d'une réponse avec la fonction `getchar`. Si la réponse est ' o ' ou ' O ' vous afficherez ' c'est parti ! '. Si la réponse est ' n ' ou ' N ' vous afficherez ' tant pis '.

- Ecrire un programme permettant de saisir un entier au clavier et d'afficher son logarithme népérien

Écrire un programme qui saisit des entiers positifs. Le programme s'arrête dès qu'un entier négatif est saisi. Il affiche alors le nombre d'entiers positifs qui ont été saisis.



- Saisir un entier  $n$  puis calculer  $n!$ 
  - Utiliser une boucle while
  - Utiliser une boucle for

- Ecrire un programme C permettant de faire un menu simulant une calculatrice d'entiers
- i1 op i2    ex 25 \* 36
- 4 opérations standard : + - / \*
- Gestion des saisies
- Boucle globale

- Ecrire un programme permettant de simuler le jeu du 421
- Fonctions random : `int random(void)` dans `stdlib.h`

- Créer un programme C déclarant deux pointeurs sur des réels (Ptr1 & Ptr2)
- Utiliser malloc pour réserver l'espace mémoire
- Affecter -42 et 345 aux emplacements pointés par Ptr1 et Ptr2
- Afficher la valeur des pointeurs et des objets pointés

- Ecrire un programme permettant de saisir dix réels au clavier et de les ranger dans un tableau. Calculer et afficher la moyenne de ces réels

- Ecrire un programme permettant de créer un tableau d'entiers à deux dimensions et de permettre à l'utilisateur de saisir les éléments qu'il désire à l'aide des indices
- Ce programme doit aussi permettre la visualisation du contenu du tableau

- Le programme va contenir la déclaration suivante : `int tab[4]={1,5,8,10};`
- Modifier le programme de manière à afficher les 4 éléments du tableau avec leur adresse

- Saisir une chaîne de caractères, afficher les éléments de la chaîne et leur adresse (y compris le caractère ' \0 ')



- Créer une fonction `lance_de` permettant de renvoyer une valeur prise au hasard entre 1 et 6

- Changer la fonction `lance_de` de manière à ce que le nombre de faces du dé soit passé en argument.

$$\mathcal{F}_{n+2} = \mathcal{F}_{n+1} + \mathcal{F}_n$$

Ecrire une fonction fibonnaci récursive qui calcul la suite de Fibonnaci.

Tels que:

➤ Fibonacci 8

Fibonanci(8) est 21

Coder les fonctions suivante de x tels que

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + o(x^n) .$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots + (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!} + o(x^{2n}).$$

- Ecrire une librairie de fonctions permettant de gérer des personnes :
  - nom, prénom, âge,
  - saisie d'une personne, affichage d'une personne.
- Utiliser cette librairie dans un programme principal permettant de saisir et d'afficher un ensemble de personnes

- Reprendre la librairie personne en remplaçant les `*obj.champ` par `obj->champ`

- Ecrire une macro de debuggage qui affiche systématiquement le numéro de la ligne et le nom du fichier
- Ecrire une macro qui rend le maximum de deux valeurs

```
enum Sexe { INCONNUE=0,MASCULIN=1,FEMININ=2};  
  
struct Personne {  
    int numero;  
    char nom[10];  
    Sexe sexe;  
};
```

Écrire les fonctions nommées suivantes :

- **créer** permettant de créer un pointeur de la structure Personne
- **détruire** permettant de détruire un pointeur de la structure Personne
- **initialiser** permettant d'initialiser les champs de la structure Personne
- **afficher** permettant d'afficher les champs de la structure Personne



## TP 25 – Matrice (1)

```
enum Type { INT=1,DOUBLE=2};

union Element {
    double d;
    int i;
};

struct Matrice {
    int taille;
    union Element **tab;
    enum Type type;
};
```

Ecrire une fonction « créer » qui créer dynamiquement une matrices  $N * N$

Ecrire une fonction « detruire » qui désaloue mémoire d'une matrices  $N * N$

Ecrire une fonction « mult » qui fait la multiplication de deux matrices  $N * N$

Ecrire une fonction « add » qui fait l'addition de deux matrices  $N * N$

- Ecrire un programme créant un fichier ' toto.bin ' binaire contenant 2 entiers saisis au clavier.
- Ecrire un programme créant un fichier ' toto.txt ' texte contenant 2 entiers saisis au clavier
- Visualiser le contenu des fichiers avec la commande type ou un éditeur

- Ecrire deux programmes afin de lire les données écrites dans toto.bin et toto.txt