

TP sur le chapitre 2 (conditions et boucles)

Exercice 1 : Sous Linux, lancez un terminal, puis réalisez les opérations suivantes :

- `mkdir progC` (pour créer le répertoire `progC`),
- `cd progC` (pour aller dans le répertoire `progC`),
- `emacs hello.c &` (pour démarrer l'éditeur de textes `emacs` sur le fichier `hello.c` ; n'oubliez pas le `&` qui vous permet de continuer à exécuter des commandes dans le terminal sans quitter l'éditeur de textes),
- écrivez dans `emacs` un programme qui affiche "hello world" à l'écran, et sauvegardez (sans quitter `emacs`),
- `gcc -o hello hello.c` (pour compiler le programme),
- `./hello` (pour exécuter le programme).

Exercice 2 : Sur votre machine, quelle est la taille des types `char`, `short`, `int` et `long` ? Il faudra écrire un programme `type` pour répondre à cette question.

Exercice 3 : Écrivez un programme `age` qui demande à l'utilisatrice son année de naissance, et qui affiche l'âge qu'elle aura au 31 décembre 2050.

Exercice 4 : En vous aidant des erreurs indiquées par le compilateur `gcc`, corrigez le programme suivant.

```
#include <stdio.h>
int main(void) {
    int h,m,s
    printf("Veuillez saisir h, m et s\n");
    scanf("%d", &h);
    scanf("%d", &m);
    scanf("%d", &s);
    s = 3600*h+60*m+s;
    printf("Resultat : %d\n");
    return;
}
```

Exercice 5 : Est-ce que `gcc` corrige toutes les erreurs ?

Exercice 6 : Écrivez un programme qui demande à l'utilisatrice de saisir un entier n , puis une suite de n entiers, et qui calcule la somme de ces n entiers.

Exercice 7 : Écrivez un programme qui demande à l'utilisatrice de saisir une suite d'entiers terminée par 0, et qui calcule la somme de ces entiers.

Exercice 8 : Écrivez un programme qui demande à l'utilisatrice de saisir une suite d'entiers croissante, et qui termine dès qu'un entier strictement inférieur au précédant est saisi. Le programme affiche alors le nombre total d'entiers saisis.

Exercice 9 : Une approximation de racine carrée de N peut s'obtenir par la suite récurrente $u_0=N$ et

$$u_{n+1} = (u_n + N/u_n)/2.$$

- Faites un programme qui calcule et affiche u_{10} .
- Faites un programme qui calcule et affiche u_x , avec x défini comme le plus petit entier tel que $|u_x^2 - N| < 0.1$.

Exercice 10 : Écrivez un programme qui affiche une ligne de n fois le caractère ' * '. Par exemple :

```
Combien vaut n ? 3
***
```

Exercice 11 : Écrivez un programme qui affiche un carré de $n \times n$ fois le caractère ' * '. Par exemple :

```
Combien vaut n ? 3
***
***
***
```

Exercice 12 : Écrivez un programme qui affiche un carré de $n \times n$ fois le caractère ' * ', décalé de deux espaces à partir du bord gauche de l'écran. Par exemple :

```
Combien vaut n ? 3
  ***
  ***
  ***
```

Exercice 13 : Écrivez un programme qui affiche un triangle rectangle dont l'angle droit est en bas à gauche. Par exemple :

```
Combien vaut n ? 3
*
**
***
```

Exercice 14 : Écrivez un programme qui affiche un triangle rectangle dont l'angle droit est en haut à gauche. Par exemple :

```
Combien vaut n ? 3
***
**
*
```

Exercice 15 : Écrivez un programme qui affiche un triangle rectangle dont l'angle droit est en bas à droite. Par exemple :

```
Combien vaut n ? 3
  *
  **
 ***
```

Exercice 16 : Écrivez un programme qui affiche un triangle rectangle dont l'angle droit est en haut à droite. Par exemple :

```
Combien vaut n ? 3
***
**
*
```

Exercice 17 : Écrivez un programme qui affiche une maison dont la taille dépend d'un paramètre n

(supposé impair). Par exemple :

Combien vaut n ? 3

*

Autre exemple :

Combien vaut n ? 5

*

Exercice 18 : Écrivez un programme dans lequel une utilisatrice doit trouver un nombre compris entre 1 et 100 choisi aléatoirement (c'est-à-dire, au hasard) par l'ordinateur. L'utilisatrice doit trouver la réponse en moins de 8 étapes. À chaque étape, l'utilisatrice propose un nombre à l'ordinateur, et l'ordinateur répond « plus petit » si le nombre choisi est plus petit que le nombre proposé, « plus grand » si le nombre choisi est plus grand que le nombre proposé, ou « gagné » si les deux nombres sont égaux.

Remarque : Pour que l'ordinateur puisse choisir un nombre aléatoire, il faut :

- Inclure les bibliothèques `stdlib.h` et `time.h`.
- Appeler `srand(time(NULL))` en début de programme (dans la fonction `main`).
- Utiliser `rand()%(MAX-MIN+1)+MIN` à chaque fois que l'on souhaite générer aléatoirement un nombre dans l'intervalle `[MAX;MIN]`.