

CI 2: Exercices complémentaires

l.cuvillon@unistra.fr

Télécom Physique Strasbourg — September 27, 2021

Consignes

Créer un fichier source par exercice.

Exercice C1 (*Utilisation de la bibliothèque mathématique*)

Écrire un programme qui :

- demande à l'utilisateur de saisir les longueurs des 2 côtés d'un triangle rectangle.
- affiche la longueur de l'hypoténuse.



Note: La fonction de la bibliothèque mathématique `sqrt()` retourne la racine carrée de son argument ou paramètre. Exemple: `a = sqrt(b);` ($a = \sqrt{b}$).

Il est nécessaire d'ajouter `#include <math.h>` en dessous de `#include <stdio.h>` pour avertir le compilateur que `sqrt()` est une fonction définie par la bibliothèque mathématique. De plus, il faut compiler avec l'option `-lm` dans la ligne de commande `gcc`.

Exercice C2 (Etoiles)

Utiliser 2 boucles **for()** imbriquées afin d'écrire un programme qui pour une valeur saisie *n*, affiche une étoile, passe à la ligne \n, affiche deux étoiles, passe à la ligne, affiche trois étoiles... jusqu'à *n* étoiles.

Exemple pour *n*=5:

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



Note: On rappelle qu'il est toujours préférable d'utiliser un nom de variable explicite tel *numLigne*, *etoile_index* plutôt que *i*, *j*, *a*, ... dans vos codes :

```
for (nb_ligne=0 ;    ;    )
{
    for (nb_etoile= ;    ... ;    )
    {
        ...
    }
}
```

Exercice C3 (Saisir un nombre avec **do...while()** ;)

1. Traduire en langage C, compléter avec les variables nécessaires, exécuter :

```
Faire
Saisir un nombre
Tant que (nombre != -1)
```

2. Modifier le programme pour afficher la moyenne d'une suite d'entiers positifs entrés au clavier. On arrêtera la saisie quand le nombre -1 est entré, comme dans l'exemple suivant :

```
Entrez un entier positif : 5
Entrez un entier positif : 2
Entrez un entier positif : 3
Entrez un entier positif : -1
La moyenne de ces 3 entiers vaut 3.333333
```

Exercice C4 (Deviner un nombre)

Le code suivant permet d'initialiser une variable avec une valeur aléatoire :

```
#include <stdlib.h>
#include <time.h>

...
int nombre_aleatoire;

srand(time(NULL));    /*initialisation du generateur pseudo-aleatoires
    avec une valeur changeante: l'heure courante*/
nombre_aleatoire= rand()%100;
    /*rand() donne une valeur aleatoire (%100 pour 0<= <100)*/
```

1. Écrire alors un petit jeu qui demande au joueur un nombre tant qu'il n'a pas deviné celui tiré par l'ordinateur. On indiquera à l'utilisateur si son nombre est plus petit ou plus grand que le nombre recherché. Voici un exemple de dialogue avec l'ordinateur :

```
Entrez votre nombre: 50
C'est plus !
Entrez votre nombre: 25
C'est moins !
...
Gagné !!!
```

2. Modifier votre programme pour limiter à 10 le nombre d'essais. On prendra soin d'indiquer au joueur le nombre d'essais qu'il lui reste.

Exercice C5 (Boucles avancées)

Écrire un programme composé de 2 boucles seulement (de préférence **while**) imbriquées, qui crée la sortie suivante sur l'écran pour n=4 :

```
0
0 1
0 1 2
0 1 2 3
0 1 2 3 4
0 1 2 3
0 1 2
0 1
0
```



Note: Cet exercice est proposé par Anne Canteaut.

N'utiliser que 2 boucles imbriquées n'a que valeur d'exercice de réflexion.

Le meilleur code, car plus simple et lisible, consiste à faire 4 boucles: 2 boucles imbriquées pour la partie croissante de l'affichage puis 2 boucles pour la partie décroissante.

Exercice C6 (*Échange: un petit peu de réflexion algorithmique*)

Écrire un programme qui :

- demande la saisie de 2 entiers a et b;
- échange les valeurs de a et b
(Utiliser une variable additionnelle pour stocker temporairement une valeur);
- affiche les nouvelles valeurs de a et b.