Auteur : Mohamed Messabihi
Matière : Introduction à l'algorithmique
✉ Mohamed.Messabihi@gmail.com

Université Abou Bakr Belkaïd - Tlemcen
Faculté des sciences
1ère L.M.D. MI

4 TP : Les fonctions

4.1 Dessin

Le but de cet exercice est de comprendre comment définir et utiliser nos propres fonctions paramétrées. Tester chacune des fonctions ci-dessous dans la fonction **main()** puis mettez la dans un commentaire pour tester la suivante, et ainsi de suite.

1. Écrire une fonction Dessiner_ligne qui affiche le caractère '*' 5 fois puis finit par un retour à la ligne.
   ☞**Exemple.** L'appel de Dessiner_ligne() affiche : * * * * *

2. Modifier la fonction Dessiner_ligne pour qu'elle affiche le caractère '*' n fois. L'entier n étant un paramètre passé à la fonction.
   ☞**Exemple.** L'appel de Dessiner_ligne(8) affiche : * * * * * * * *

3. Modifier la fonction Dessiner_ligne pour qu'elle affiche n fois un caractère c. L'entier n et le caractère c sont passés comme paramètres à la fonction.
   ☞**Exemple.** L'appel de Dessiner_ligne(10, '#') affiche : # # # # # # # # # #

4. En utilisant la fonction Dessiner_ligne, écrire une fonction Dessiner_carre qui permet d'afficher un carré de côté n avec le caractère c. L'entier n et le caractère c sont passés en paramètres.
   ☞**Exemple.** L'appel de Dessiner_carre(5, 'o') affiche le carré ci-contre.

```
o o o o o
o o o o o
o o o o o
o o o o o
o o o o o
```

5. En utilisant la fonction Dessiner_ligne, écrire une fonction Dessiner_triangle qui permet d'afficher un triangle rectangle au lieu d'un carré.
   ☞**Exemple.** L'appel de Dessiner_triangle(4, 'X') affiche le carré ci-contre.

```
X
X X
X X X
X X X X
```

6. Modifier la fonction Dessiner_triangle en lui ajoutant un paramètre sens pour prendre en compte l'orientation de l'hypoténuse du triangle. Si ce paramètre vaut 0, l ?hypoténuse est tournée vers le haut, comme dans l'exemple précédent. Si le paramètre vaut 1, elle est tournée vers le bas écrire une fonction.
   ☞**Exemple.** L'appel de Dessiner_triangle(6, 'U', 1) affiche le carré ci-contre.

```
U U U U U U
U U U U U
U U U U
U U U
U U
U
```

7. En utilisant uniquement la fonction Dessiner_triangle, écrire une fonction Dessiner_grand_triangle, qui affiche un triangle comme indiqué sur l'exemple ci-contre.
   ☞**Exemple.** L'appel de Dessiner_grand_triangle(4, '@') affiche le carré ci-contre.

```
   @
  @ @
 @ @ @
@ @ @ @
 @ @ @
  @ @
   @
```

4.2 Triangle de Pascal

On rappelle que la fonction factorielle est définie sur les entiers positifs de la façon suivante :

$$\begin{cases} 0! = 1 \\ n! = n \times (n-1) \times (n-2) \times \ldots \times 2 \times 1 \quad si \quad n>1 \end{cases}$$

1. Écrire une fonction `fact` qui permet de calculer et renvoyer la factorielle d'un entier passé en paramètre.

2. Écrire une fonction qui utilise cette fonction `fact` pour calculer et afficher les coefficients binomiaux du triangle de Pascal avec la formule suivante : $C_p^q = \frac{p!}{q!.(p-q)!}$

Remarque : on commence par $p = 0$ et $q = 0$ (avec p : ligne, q : colonne, $p \leq q$).

Le tableau ci-contre donne par exemple les coefficients pour p et q allant de 0 à 5.

q / p	0	1	2	3	4	5
0	**1**	–	–	–	–	–
1	**1**	**1**	–	–	–	–
2	**1**	**2**	**1**	–	–	–
3	**1**	**3**	**3**	**1**	–	–
4	**1**	**4**	**6**	**4**	**1**	–
5	**1**	**5**	**10**	**10**	**5**	**1**

Exemple :
L'appel de `triangle_Pascal(5)` donnera l'affichage ci-dessous :

```
1
1  1
1  2  1
1  3  3  1
1  4  6  4  1
1  5  10 10 5  1
```

4.3 Nombre premier suivant

1. Écrire une fonction `est_divisible(n, m)` qui renvoie 1 si n est divisible par m et 0 sinon.

2. Écrire une fonction `somme_diviseurs` qui utilise la fonction précédente pour calculer et retourner la somme des diviseurs d'un nombre passé en paramètre.

3. Écrire une fonction `est_premier` qui prend un paramètre entier n et retournant 1 si le nombre n est premier et 0 dans le cas contraire. Elle devra utiliser l'une des fonctions précédentes.

4. Écrire une fonction `prochain_premier` qui prend un paramètre entier n puis utilise la fonction précédente pour trouver et renvoyer le plus petit nombre premier plus grand ou égal à n.

5. Écrire un programme qui demande un entier n à l'utilisateur et affiche le premier nombre premier plus grand ou égal à n.

4.4 Échange

1. Écrire une fonction qui échange les valeurs des deux paramètres a et b qu'elle prend en entrée.

2. La fonction `main()` donnée ci-contre utilise cette fonction. Quel sera son résultat ? Testez-la dans un programme que vous exécuterez pour vérifier votre réponse.

3. Afin d'expliquer ce qui s'est passé, copier les lignes 3 et 5 respectivement au début et à la fin de la fonction `echange()`. Expliquez.

```
1. int main(void){
2.     int a=1, b=2;
3.     printf("avant: a=%d, b=%d\n",a,b);
4.     echange(a, b);
5.     printf("après: a=%d, b=%d\n",a,b);
6. }
```