

NOTIONS DE BASE

Exercice 1

1. Dans le répertoire principal (*home directory*) créer trois répertoires : **Rep1**, **Rep2** et **Rep3**. Puis, dans **Rep1**, créer deux sous-répertoires **Ssrep1** et **Ssrep2**.
2. Dans **Ssrep1**, créer le fichier **prog1.c**.
3. Éditer le fichier **prog1.c** et écrire le programme qui affiche à l'écran la phrase :

```
Mon premier programme en langage C
```

Dans la ligne de commande commençant par `printf`, rajouter `\n` pour faire apparaître le message sur deux lignes, par exemple :

```
Mon premier programme
en langage C
```

4. Compiler **prog1.c** à l'aide de la commande `gcc`. On appellera l'exécutable **prog1.out**. Lancer l'exécutable.
5. Copier **prog1.c** dans le répertoire **Ssrep2**.
6. Détruire **Rep3**.
7. Renommer **Rep1** en **TD1**.
8. Déplacer le contenu de **Ssrep2** dans le répertoire **TD1**.
9. Détruire les deux sous-répertoires **Ssrep1** et **Ssrep2** ainsi que le répertoire **Rep2**.

Exercice 2

Écrire le programme suivant qui affiche le cosinus d'un nombre réel saisi à l'écran.

```
int main()
{
    double x;
    printf(''donner la valeur de x\n'');
    scanf('%lf', &x);
    printf(''cos(%lf) = %lf\n'', x, cos(x));
    return(0);
}
```

Compiler et exécuter le programme. Que doit-on rajouter pour le bon fonctionnement du programme.

Exercice 3

Dans un programme, trois entiers seront initialisés selon

```
int m=5 ; n=9 ; p=4
```

Soit un quatrième entier `q`. Prédire le résultat des opérations suivantes.

```
q=(n<p) ;
q=(p==m+n) ;
q=(p>n) ;
q=(m>n || m>p) ;
q=(p!=n) ;
```

Vérifier vos réponses en exécutant un programme qui affiche le résultat de chaque opération.

Exercice 4

On se propose d'écrire un programme qui donne les racines d'un polynôme du type $ax^2 + b = 0$ avec $a \neq 0$ et $b \neq 0$.

Un tel polynôme admet deux racines réelles si a et b sont signe contraire (c'est-à-dire si le produit ab est négatif), deux racines complexes sinon.

Effectuer le test qui convient à l'aide de l'instruction `if` et faire afficher à l'écran la valeur des deux racines, $\pm\sqrt{-b/a}$, ou la mention `Deux racines complexes` selon que ce test est vérifié ou non.

Exercice 5

On se propose d'écrire un programme qui donne les racines d'un polynôme du type $ax^2 + bx + c = 0$ avec $a \neq 0$.

Après le calcul du discriminant $\Delta = b^2 - 4ac$, trois cas de figure se présentent : le polynôme admet deux racines réelles distinctes ($\Delta > 0$), une racine double ($\Delta = 0$) ou deux racines complexes ($\Delta < 0$).

Dans un premier temps, on calculera le discriminant et à l'aide de l'instruction `if`, on affectera à un entier `d` la valeur `+1`, `0` ou `-1` selon le signe de Δ .

Dans un deuxième temps, à l'aide de l'instruction `switch`, compléter le programme pour qu'il affiche à l'écran la mention `Deux racines réelles`; `Une racine double` ou `Deux racines complexes` selon la valeur de l'entier `d`.

Exercice 6

Écrire un programme qui affiche le résultat de la somme de deux entiers.

1. Dans un premier temps, deux entiers `m` et `n` seront initialisés en début de programme. Le résultat de la somme `m+n` sera affiché à l'écran.
2. Dans un deuxième temps, en utilisant la commande `scanf`, les deux entiers seront saisis par l'utilisateur puis leur somme sera affichée.
3. Modifier le programme pour effectuer la somme de deux réels `a` et `b`.

Exercice 7

Dans un programme en C,

1. Déclarer deux variables de type `double` `x` et `y` et un pointeur `pt` de type `double`.
2. Afficher les adresses de `x`, `y` et `pt`.
3. Initialiser `x` et `y` à 2 et 3. Faire pointer `pt` sur `x`.
4. Afficher `pt` et la valeur stockée à l'adresse pointée (`&x`), c'est à dire `*pt`.
5. Affecter à `*pt` la valeur de `y` puis afficher les valeurs de `x` et `y`.
6. Afficher de nouveau les adresses de `x`, `y` et `pt`. Conclure sur les résultats obtenus.

Exercice 8

À l'aide d'une boucle `for`, écrire un programme en C permettant d'afficher à l'écran les dix premiers termes de la suite :

$$a_{n+1} = 1.1a_n$$

Le n^{ieme} terme de la suite est stocké à l'aide d'un `double` qui à chaque pas de la boucle est remplacé par le n^{ieme+1} terme. Prendre $a_0 = 2$ et pour l'affichage utiliser l'instruction :

```
printf("an = %lf\n", a);
```

Exercice 9

Écrire un programme en C permettant de calculer et d'afficher les sommes,

$$S_1 = \sum_{n=1}^N \frac{1}{n^2}$$

et

$$S_2 = 4 \sum_{n=0}^N \frac{(-1)^n}{2n+1}$$

Fixer la valeur de N à l'aide d'un `#define` en début de programme. Vers quel nombre tend la somme S_2 lorsque N est grand ?

Exercice 10

Écrire un programme en C permettant de calculer et d'afficher à l'aide d'une boucle `do...while` les N premiers termes de la suite,

$$a_{n+1} = \frac{x}{n} a_n$$

où x est un `double` saisi par l'utilisateur, N est défini à l'aide d'un `#define` en début de programme et $a_0 = 1$. À chaque pas de boucle incrémenter la somme,

$$S(n) = \sum_{n=0}^N a_n$$

du terme a_n . Vers quelle valeur connue tend S_n lorsque x vaut 1 ?