

5 Exercices de programmation :

◇ Exercice 1 : Écrire un programme qui demande à l'utilisateur de saisir un caractère au clavier puis affiche son code ASCII.

◇ Exercice 2 : Écrire un programme qui lit un entier au clavier puis qui affiche son carré, sa racine carrée et son exponentielle. Les deux derniers résultats sont des réels, ils seront affichés avec une précision de 10^{-4} .

Vous utiliserez les fonctions de la librairie `math.h` de prototype :

- `double sqrt (double)`
- `double exp (double)`

Exemple d'exécution :

Entrer un entier n : 5

*5*5=25; racine(5)=2.2361; exp(5)=1.484132 E2*

Remarques:

- En C, on peut convertir une valeur d'un type à un autre en mettant le nouveau type entre parenthèses avant la valeur.

Par exemple, si x et y sont de type `float`, on peut écrire :

`x=(float) sqrt ((double) y);`

- Penser à modifier la ligne de compilation avec la balise `-lm` pour indiquer à la machine que la librairie `math` va être utilisée :

`gcc -Wall 'nom_fichier.c' -o 'nom_fichier_executable_sans_extension' -lm`

◇ Exercice 3 : Écrire un programme qui demande un entier compris entre 0 et 127 et qui affiche la nature du caractère correspondant :

- $\{0 \dots 31\} \cup \{127\}$: caractères de contrôle non affichables
- $\{65 \dots 90\}$: alphabet latin majuscule
- $\{97 \dots 122\}$: alphabet latin minuscule
- $\{48 \dots 57\}$: chiffres 0 \dots 9
- sinon : divers caractères affichables : ponctuation, $\{, \}$ \dots

Exemple d'exécution :

Entrer un entier compris entre 0 et 127 : 68

C'est le caractère 'D' de l'alphabet latin majuscule.

◇ Exercice 4 : Écrire un programme qui affiche le minimum de 3 variables (dont les valeurs sont fixées directement dans le programme (sans saisie au clavier)).

◇ Exercice 5 : Écrire un programme qui affiche la valeur de X^N où X est un réel et N un entier naturel saisis par l'utilisateur. *on écrira obligatoirement une boucle while, la boucle for sera vue au TP2*

◇ Exercice 6 : Écrire un programme qui étant donné un réel $x > 1$ affiche la plus grande puissance de 2 strictement inférieure à x . Par exemple, pour :

- $x = 1.3$ le résultat est 1 car $2^0 = 1 < 1,3 \leq 2^1$

· $8 < x \leq 16$ le résultat est 8 car $2^3 = 8 < x \leq 2^4$

◇ Exercice 7 : Écrire un programme qui étant donné un réel $x > 1$ affiche l'**exposant** de la plus grande puissance de 2 strictement inférieure à x . Par exemple, pour :

· $x = 1.3$ le résultat est 0 car $2^0 = 1 < 1,3 \leq 2^1$

· $8 < x \leq 16$ le résultat est 3 car $2^3 = 8 < x \leq 2^4$

◇ Exercice 8 : Placer un fichier texte déjà existant dans le répertoire courant. Écrire un programme qui compte le nombre de lettres minuscules 'e' contenues dans ce fichier.

◇ Exercice 9 : Le calendrier grégorien est le calendrier actuellement utilisé dans la majeure partie du monde. Conçu pour corriger la dérive séculaire du calendrier julien, sa dénomination porte le nom de son instigateur Grégoire XIII, pape de 1572 à 1585. Son point de départ, l'an 1, est une estimation de la naissance de Jésus.

Depuis l'instauration du calendrier grégorien, sont bissextiles, les années :

- divisibles par 4 mais non divisibles par 100
- ou divisibles par 400

Écrire un programme qui détermine si une année saisie au clavier est bissextile.

◇ Exercice 10 : Exécutez le programme suivant, que se passe t-il ?

```
#include <stdio.h>

int main()
{
  int a=10;
  while (a!=0)
  {
    a=a*a;
    printf("%d\n",a);
  }
  return (0);
}
```

Recherchez dans `/usr/include/machine` le fichier : **limits.h**.

Expliquez le comportement du programme précédent par la valeur de la constante

`__INT_MAX__`

◇ Exercice 11 : Ecrivez un programme de résolution d'une équation du second degré à coefficients réels saisis par l'utilisateur. Discuter en fonction des coefficients : $ax^2 + bx + c$ (envisager tous les cas, par exemple si $a = b = 0$, $c \neq 0$, il n'y a pas de solutions...)

1. version 1 : solutions réelles (vu en TD)
2. version 2 : solutions complexes
3. version 3 : ajouter une boucle demandant à l'utilisateur s'il veut résoudre une nouvelle équation ou s'arrêter.

