

TD1 : Introduction au C

Dans `td1.h` se trouve la description par des “macros” des instructions d’écriture, de lecture et d’attente dont voici la source.

```
/* -- source de td1.h -- */
#ifndef TD1H
#define TD1H
#include <stdio.h>
/* lit(variable) : Lit un entier au clavier (stdin) dans "variable"
 * ecrit(expression) : Ecrit un entier (calculé par expression)
 * sur l'écran (stdout)
 */

#define lit(a)    printf("Entrer une valeur entière: "); scanf("%d", &(a))
#define ecrit(a)  printf("%u\n", (unsigned int)(a))
#define attend() printf("\nAppuyer sur RETURN\n"); getchar()

#endif
```

Utilisez uniquement le type `int` pour vos variables.

EXERCICE 1:

1. Ecrivez un programme C qui demande un entier à l'utilisateur et affiche le carré de cet entier.
2. Ecrivez un programme C qui demande deux entiers à l'utilisateur et affiche le carré de la somme de ces deux entiers.

***** Correction Exercice 1 *****

```
void exercice1_1() {
    int valeur;
    int carre;
    lit(valeur);
    carre = valeur * valeur;
    ecrit(carre);
}

void exercice1_1_version_2() {
    int valeur;
    int carre;
    printf("Entrez une valeur: ");
    scanf("%d",&valeur);
    carre = valeur * valeur;
    ecrit(carre);
}
```

```

void exercice1_1_version_3() {
    int valeur;
    lit(valeur);
    valeur = valeur * valeur;
    ecrit(valeur);
}

void exercice1_1_version_4() {
    int valeur;
    lit(valeur);
    ecrit(valeur*valeur);
}

/* Pour la version 5, création de 2 fichiers sources:
   -fichier carre.h (fichier en-tete)
   -fichier carre.c

   Source de CARRE.H:
#ifdef CARRE_H
#define CARRE_H
int calculer_carre(int valeur); // prototype
#endif

   Source de CARRE.C:
#include "carre.h"

int calculer_carre(int valeur) {
    return (valeur*valeur);
}
*/

void exercice1_2() {
    int valeur_un;
    int valeur_deux;
    int resultat;
    lit(valeur_un);
    lit(valeur_deux);
    resultat = (valeur_un * valeur_un) + (valeur_deux * valeur_deux);
    ecrit(resultat);
}

Faire 4 versions différentes:
-une avec scanf et printf et non lit et ecrit
-une en ne mettant pas carre mais en faisant valeur = valeur*valeur
-une en faisant une fonction carre avant le main
-une en mettant le prototype de la fonction dans un .h

```

EXERCICE 2: Volume du cube

Ecrivez un programme qui lit la longueur du côté d'un cube puis qui affiche son volume.

***** Correction Exercice 2 *****

```
/*
  Fonction qui lit la longueur du côté d'un cube puis qui
  affiche son volume
*/
void exercice2() {
  int cote;
  int volume;
  lit(cote);
  volume = cote * cote * cote;
  ecrit(volume);
}
```

EXERCICE 3: Volume d'une sphère

Ecrivez un programme qui lit le rayon puis affiche le volume d'une sphère. On définira la constante π de manière symbolique par `#define PI 3.1416`. Le volume d'une sphère est $\frac{4}{3}\pi r^3$

***** Correction Exercice 3 *****

```
void exercice3() {
  int rayon;
  int volume;
  lit(rayon);
  volume = (4/3) * PI * (rayon*rayon*rayon);
  ecrit(volume);
}
```

EXERCICE 4: Affectation

1. Affecter successivement deux valeurs à une même variable tout en l'affichant pour visualiser le résultat. 2. Affecter (en une seule ligne) à 3 variables la valeur 1.

***** Correction Exercice 4 *****

```
void exercice4_1() {
  int var;
  var = 4;
  var = 15;
  ecrit(var);
}

void exercice4_2() {
```

```

int a, b, c;
a=b=c=1;
ecrit(a);
ecrit(b);
ecrit(c);
}

```

EXERCICE 5: Incrémentation

Trouvez 4 façons différentes d'incrémenter une variable (ajout de 1 à la valeur de cette variable) tout en affichant les résultats intermédiaires.

***** Correction Exercice 5 *****

```

void exercice5() {
    int r = 1;

    ++r;        //pré incrémentation
    ecrit(r);
    r++;        //post incrémentation
    ecrit(r);
    r = r + 1; //Ajout de 1 à la valeur de r
    ecrit(r);
    r+=1;      //Ecriture plus compact
    ecrit(r);
}

```

EXERCICE 6: pré et post incrémentation

Anticipez sur papier le résultat de l'exécution de ce programme.

```

#include "td1.h"

main ()
{
    int r;

    r = 1;
    ecrit(r);
    ecrit(r++);
    ecrit(r);
    ecrit(++r);
    ecrit(r);

    ecrit(1 + r++ + r);
    ecrit(1 + r + r++);
    ecrit(1 + ++r + r);

    attend();
}

```

***** Correction Exercice 6 *****

1
1
2
3
3
7
9
13

EXERCICE 7: Opérations booléennes

Essayez les différentes opérations booléennes en affichant leurs résultats. Par exemple :

```
ecrit(0 == 0);
```

N.B.: Si possible, trouvez d'autres exemples!

***** Correction Exercice 7 *****

```
ecrit(0 == 0); // 1
ecrit(0 > 5); // 0
ecrit(8 > 5); // 1
ecrit(0 && 5); // 0
ecrit(6 && 5); // 1
ecrit(6 || 1); // 1
ecrit(0 || 0); // 0
ecrit(0 || 6); // 1
```

EXERCICE 8: Opérateurs binaires

Prédire le résultat de l'exécution du programme suivant:

```
#include "td1.h"

main ()
{
    int a, b;

    a = 1 << 3;
    ecrit(a);
    ecrit(8 >> 2);
    b = a + (8 >> 2);
    ecrit(b);
    ecrit(a & b);
    ecrit(7 / 4);
    attend();
}
```

***** Correction Exercice 8 *****

```
a = 1 << 3;
ecrit(a);           // 8
ecrit(8 >> 2);      // 2
b = a + (8 >> 2);
ecrit(b);           // 10
ecrit(a & b);        // 8
ecrit(7 | 4);        // 7
ecrit(7 ^ 4);        // 3
attend();
```

EXERCICE 9: Priorité et associativité

Ecrivez un programme qui calcul (sans simplifier) les valeurs: $4 + \frac{4}{2}$, $\frac{4+4}{2}$, $24/6/2$, $\frac{24}{\frac{6}{2}}$,

***** Correction Exercice 9 *****

```
int a;

a = 4 + 4 / 2;
ecrit(a);
a = (4 + 4) / 2;
ecrit(a);
a = 24 / 6 / 2;
ecrit(a);
a = 24 / ( 6 / 2 );
ecrit(a);
```

EXERCICE 10: $a = bq + r$

La division entière $\frac{a}{b}$ donne $a = bq + r$ avec q le quotient et r le reste de la division.

Ecrivez le programme qui à partir de a et b calcul et affiche r et q. Enfin afficher un entier 1 ou 0 suivant que le calcul est exact ou non.

***** Correction Exercice 10 *****

```
int a,b;
int q,r;
lit(a);
lit(b);

q = a/b;
r = a%b;

ecrit(q);
ecrit(r);
```