

Développement Logiciel Cryptographique

TP n° 1

Exercice 1

Écrire en C un programme qui affiche les tailles de chacun des types de base entiers et flottants.

Exercice 2

Augmenter votre programme pour qu'il affiche les tailles des types pointeurs sur chacun des types précédents.

Exercice 3

Le programme fourni en annexe montre comment afficher les différents octets de la représentation en virgule flottante d'une variable de type `float`.

Reprenez ce programme et donnez à `x` la valeur 0,40625. Exécutez et interprétez le résultat.

Recommencez en donnant à `x` des valeurs de votre choix.

Exercice 4

On suppose que $(a, b, c) = (3, 4, 1)$. Comment sont évaluées les expressions suivantes ?

- `(a == 3 || b > 5) && c != 0`
- `a == 3 || b > 5 && c != 0`
- `a == 2 || b != 4 || c < 7`

Exercice 5

Comment sont évaluées les expressions suivantes ?

- (a == 3) || b > 5
- (a = 3) || b != b
- 1 && 3
- !0 || b
- !0 && b
- if (a) {...}
- if (a = b) {...}

Exercice 6

Analysez le segment de programme suivant et prédisez ce qu'il va afficher.
Exécutez le programme et vérifiez votre prédiction.

```
b = 2;
c = 5;
d = 0;

if (b == c == d)
    printf("Vrai : (b,c,d) = (%u,%u,%u)\n", b, c ,d);
else
    printf("Faux : (b,c,d) = (%u,%u,%u)\n", b, c ,d);
```

Refaire l'exercice avec les variantes suivantes :

- if (b == c = d) {...}
- if (b = c == d) {...}
- if (b = c = d) {...}

```

#include <stdio.h>

int main()
{
    unsigned char *ptr;
    float x;

    x = -38.25;

    printf("\n");
    printf("Valeur de x : %f\n", x);

    ptr = (unsigned char *) &x;
    printf("\n");
    printf("Adresse de la variable x : %p\n", ptr);

    printf("\n");
    printf("Valeur de l'octet à l'adresse %p : %02X\n", ptr + 0, *(ptr + 0));
    printf("Valeur de l'octet à l'adresse %p : %02X\n", ptr + 1, *(ptr + 1));
    printf("Valeur de l'octet à l'adresse %p : %02X\n", ptr + 2, *(ptr + 2));
    printf("Valeur de l'octet à l'adresse %p : %02X\n", ptr + 3, *(ptr + 3));

    // *(ptr + 2) = (char) 0x99;
    // printf("\n");
    // printf("Valeur de x : %f\n", x);

    printf("\n");

    return 0;
}

```