



Travaux Dirigés d'algorithmique n°5

Cours d'informatique de deuxième année

—Licence 2ème Année, Premier Semestre MIAS—

coercition, ensembles

► Exercice 1. Coercition et pointeur

Le programme suivant affiche les valeurs 2, 3, 5. Expliquez pourquoi.

```
#include <stdio.h>

int main(void)
{
    char a[8]={1,2,3,4,5,6,7,8};
    char b;
    void *pt;

    pt=a;
    b= ((char *)pt)[1];
    printf("%d\n",b);
    b= ((short *)pt)[1];
    printf("%d\n",b);
    b= ((int *)pt)[1];
    printf("%d\n",b);
}
```

► Exercice 2. Comparaison

Ecrire une fonction effectuant la comparaison de deux variables de type quelconque. Les variables seront transmises par adresse et comparées octet par octet.

► Exercice 3. Ensembles

Un ensemble est une collection d'éléments distincts. On souhaite représenter en machine le type abstrait ensemble (plus précisément, ensemble d'éléments d'un certain type). On donne ci-dessous quelques opérations de base qui permettront de manipuler des objets de ce type.

- initialiser un ensemble à l'ensemble vide,
- tester si un ensemble est vide,
- calculer le nombre d'éléments d'un ensemble
- tester l'appartenance d'un élément à un ensemble,
- ajouter, supprimer un élément à un ensemble,
- saisir, afficher un ensemble,
- calculer l'union, l'intersection de deux ensembles,
- etc

Pour simplifier, nous supposons que les ensembles sont **des ensembles d'entiers**.
On représente d'abord un ensemble par une structure contenant :

- un tableau (les éléments de l'ensemble),
- un entier (le cardinal de l'ensemble).

Définir en C un type **Ensemble** pour cette représentation, nous verrons plus tard comment réserver la place mémoire en fonction de la taille de l'ensemble. Implémenter les opérations mentionnées ci-dessus en supposant que :

- le tableau n'est pas trié,
- le tableau est trié (donc chaque opération devra conserver cette propriété).

Evaluer la complexité de ces opérations dans chacun des cas.