

TD 1 : les pointeurs.

Version du 1^{er} mars 2011

Exercice 1.

Soient i une variable de type `int`, p et q des pointeurs sur `int`.

On suppose que :

- i se trouve à l'adresse 4830000,
- p à l'adresse 4830010, et
- q à l'adresse 4830020.

On suppose aussi que :

- l'adresse 4830000 contient 2,
- l'adresse 4830010 contient 4840010,
- l'adresse 4830020 contient 4840020,
- l'adresse 4840010 contient 12, et
- l'adresse 4840020 contient 22.

1. Lesquelles parmi les expressions suivantes affectent un pointeur à une variable de type pointeur, ou un entier à une variable de type entier, et pour ces expressions, quel est leur effet sur le contenu de la mémoire.

$$\begin{array}{cccc}
 1) p = i & | & 2) *p = \&i & | & 3) \&p = q & | & 4) p = \&q \\
 5) p = *\&q & | & 6) *p = q & | & 7) p = *q & | & 8) *p = *q
 \end{array}$$

Exercice 2.

Considérons les déclarations suivantes :

```
int tab[] = {5, 15, 34, 54, 14, 2, 52, 72};
int *p = &tab[1], *q = &tab[5];
```

1. Quelle est la valeur de $*(p+3)$?
2. Quelle est la valeur de $*(q-3)$?
3. Quelle est la valeur de $q-p$?
4. La condition $p < q$ est-elle vraie ou fausse ?
5. La condition $*p < *q$ est-elle vraie ou fausse ?

Exercice 3.

Considérons l'exécution des instructions suivantes :

```
#define N 10
int tab[N] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int *p = &tab[0], *q = &tab[N-1], temp;
while (p < q) {
    temp = *p;
    *(p++) = *q;
    *(q--) = temp;
}
```

1. Quel est le contenu du tableau.

Exercice 4.

Soit `tab` un tableau à une dimension et `p` une variable de type pointeur. Supposons que l'affectation `p = tab` vient d'être exécutée, et considérons les expressions suivantes :

- a) `p == tab[0]`
- b) `p == &tab[0]`
- c) `*p == tab[0]`
- d) `p[0] == tab[0]`

1. Lesquelles parmi les expressions ci-dessous sont illégales en raison d'incompatibilité de type ?
2. Parmi les expressions légales lesquelles sont vraies ?

Exercice 5.

Soit la déclaration suite :

```
#define N 10
int tab[N] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```

1. Utilisez un pointeur (sur un entier) pour écrire un programme C qui permet l'affichage, en ordre croissant, puis décroissant des éléments de `tab`.
2. Écrivez un programme C qui copie le contenu de `tab` dans un autre espace-mémoire alloué dynamiquement.