TP4 – Les pointeurs

Année 2021/2022

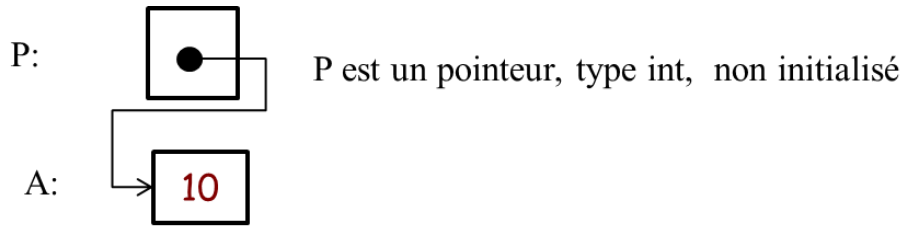
1. Définition

Un pointeur contient une adresse mémoire. Cette adresse pointe sur un objet qui dépend de la déclaration du pointeur. Les pointeurs sont utilisés plutôt, pour gérer correctement les tableaux. En passant un pointeur comme argument d'une fonction, il est alors possible à la fonction de modifié le contenu de la variable pointée par ce pointeur.

Deux opérateurs unaires sont liés aux pointeurs :

1. L'opérateur de « contenu » d'une adresse : *
   - Si « p » est un pointeur de type entier, « *p » vaut l'entier contenu à l'adresse « p ».
2. L'opérateur d'adresse d'une variable : &
   - Si « a » est une variable, « &a » vaut l'adresse à laquelle est stockée la variable « a ».

Exemple :



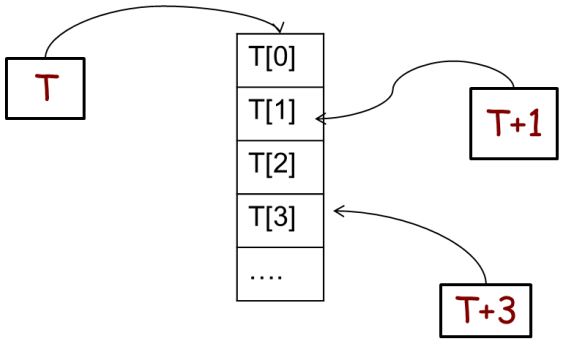
P = &A; affecte l'adresse de la variable A à la variable P.
On dit que P pointe vers A.

Exercice 1 :

En utilisant les pointeurs, définir une fonction « inverser » qui permet d'inverser le contenu de deux variables de type int.

2. Les tableaux

Un tableau est un pointeur sur une zone mémoire où sont stockées toutes les cases de façon consécutives : T pointe vers la première case du tableau (T[0]), (T+1) pointe vers la deuxième case du tableau (T[1]), ….



Exercice 2 :

Soit P un pointeur qui 'pointe' sur un tableau A:

int A[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};

int *P;

P = A;

Quelles valeurs ou adresses fournissent ces expressions:

*P+2	14 ✓
*(P+2)	34 ✓
&P+1	@ *P + 1 ✗
&A[4]-3	@A1 ✗ ✓
A+3	@A3 ✓
&A[7]-P	@ P7 -P0 = 7 ✓
P+(*P-10)	@A2 ✓
(P+(P+8)-A[7])	*(P+1) = 23 ✗

3. Allocation dynamique de la mémoire

Il est possible de faire pointer un pointeur vers une zone mémoire de taille définie dynamiquement au cours de l'exécution du programme. Deux fonctions seront à utiliser :

- La fonction « malloc » : permet d'allouer la mémoire nécessaire pour un tableau, en précisant sa taille ainsi que le type des éléments contenus dans le tableau.
- La fonction « sizeof » : permet d'obtenir la taille de la variable qu'on souhaite sauvegarder.

Exemple :

```
Int n= 15;
double *t;
t =(double *)malloc(n*sizeof(double));
```

Exercice 3 :

Ecrire le programme qui permet de définir un tableau d'entiers, avec un pointeur.

Toujours en utilisant les pointeurs, écrire :

- Une fonction qui permet de remplir le tableau
- Une fonction qui permet d'afficher le tableau.
- Une fonction qui permet d'inverser le tableau.

Exercice 4:

Écrire une fonction permettant de trier par ordre croissant les valeurs entières d'un tableau de taille quelconque (transmise en argument). Le tri pourra se faire par réarrangement des valeurs au sein du tableau lui-même

4. Les chaines de caractères

En langage C, on définit une chaine de caractère comme étant un tableau de caractère. La dernière case de ce tableau est toujours réservée au caractère « \0 ».

Pour définir une chaine de caractère, on peut procéder de deux manière :

- Une allocation statique : utilisée lorsqu'on connait exactement la taille de notre chaine de de caractère.

  Char ch[8] = « bonjour » ;

  Char ch[8] = {'b', 'o', 'n', 'j', 'o', 'u', 'r', '\0'}

- Une allocation dynamique : en utilisant un pointeur et la fonction malloc pour l'allocation de la mémoire.
  char *ch = « Bonjour » ;

Ou

```
char *ch;
ch = (char *)malloc(8 * sizeof(char));
```

Pour afficher une chaine de caractère, on utilise l'opérateur « %s ».
Printf(« %s\n », ch) ;
Pour saisir une chaine de caractère, on peut utiliser la fonction gets.
Gets(ch) ;
La bibliothèque « string.h » contient des fonctions qui permettent la manipulation des chaines de caractères

Exercice 5 :

Ecrire une fonction comparerChaine() qui permet d'effectuer une comparaison de deux chaînes. Elle prend en paramètre d'entrée deux chaînes de caractères et elle retourne :

- 0 en cas d'égalité.
- 1 si la première chaîne de caractères est supérieure alphabétiquement à la seconde.
- -1 si la première chaîne de caractères est inférieure alphabétiquement à la seconde

Exercice 6 :

Ecrire un programme qui supprime la première occurrence d'une chaine de caractère OBJ dans une chaine SUJ.

Exercice 7 :

Ecrire un programme qui remplace la première occurrence d'une chaine CH1 par la chaine CH2 dans une chaine de caractère SUJ. Utiliser une chaine de sauvegarder Fin pendant le remplacement.

Exercice 8 :

Ecrire un programme qui remplace toutes les occurrences d'une chaîne de caractères CH1 par la chaîne CH2 dans une chaîne de caractères SUJ. Utiliser une chaîne de sauvegarde FIN pendant le remplacement.