

**Pointeurs**  
**Travaux Pratiques – Séance n° 10a**

---

## 1 Pour commencer

1) Considérez la liste d'instructions suivante :

```
int *p;  
int i;  
int k;  
i = 42;  
k = i;  
p = &i;
```

Après l'exécution de ces instructions, quelle instruction affectera 75 à i ?

1. k = 75;
2. \*k = 75;
3. p = 75;
4. \*p = 75;

2) Expliquez l'erreur suivante :

```
char c = 'A';  
double *p = &c;
```

3) Faites la trace de ce programme. Commencez par afficher l'adresse du tableau `blocks`.

```
int main() {  
    char blocks[3] = {'A', 'P', 'W'};  
    char *ptr = &blocks[0];  
    char temp;  
  
    temp = blocks[0];  
    temp = *(blocks + 2);  
    temp = *(ptr + 1);  
    temp = *ptr;  
  
    ptr = blocks + 1;  
    temp = *ptr;  
    temp = *(ptr + 1);  
  
    ptr = blocks;  
    temp = ++*ptr;  
    temp = ++*ptr;  
    temp = *ptr++;  
    temp = *ptr;
```

```
    return EXIT_SUCCESS;
}
```

## 2 Paramètres résultats

4) Écrivez la fonction `void swap(int* a, int* b)` qui échange les valeurs de `a` et `b`.

## 3 Chaînes

5) Récrivez les fonctions de manipulation de chaîne de caractères, `strcat`, `strchr` et `compare`, avec la notation de pointeur.

6) Écrivez la fonction `void metEnMajuscule(char *)` qui met en majuscule la chaîne de caractères donnée en paramètre.

## 4 Argv/argc

7) Écrivez un programme qui prend comme paramètres programmes une expression arithmétique, en notation infixe, formée de deux opérandes entiers et d'un opérateur (+, −, ×, et /), et qui affiche sur la sortie standard le résultat de son évaluation. Par exemple, si le programme exécutable s'appelle `calc`, on aura :

```
$ calc 30 + 5
30 + 5 = 35
```

8) Écrivez un programme qui affiche les  $n$  premières lignes qui lui sont données sur l'entrée standard. Le nombre de lignes désiré sera passé en paramètre au programme. Si le paramètre est omis, la valeur 5 sera utilisée par défaut.

9) Écrivez le programme `OptionParser` qui analyse les paramètres et les options d'une ligne de commandes et qui les affiche sur la sortie standard. Les options sont introduites par un tiret et **précèdent** les paramètres de la commande.

**Exemple :**

```
$ OptionParser -fc -abc -k foo -z
option f
option c
option a
option b
option c
option k
paramètre foo
paramètre -z
```