

Programmation en langage C

Séries d'exercices corrigés

```
void nom_fonction(type *p1);  
int main ()  
{  
    type v=valeur;  
    type *p = &v;  
    nom_fonction (p);  
    printf ("%lettre", v);  
    //ici v après traitement contient le résultat.  
}
```

- ▶ Cours bien illustrés
 - ▶ Exemples pratiques
- + CD-ROM contenant les sources
de toutes les applications du livre



Remerciements	03
Tables des matières	04
Introduction	09
Chapitre 1 : Initiation et généralités	11
1. Définitions de base	12
2. Algorithmique	14
3. Fonctionnement de la mémoire	19
♦ <i>Application</i>	21
Chapitre 2 : Les variables	23
1. Les variables	24
2. L'écriture	26
3. La lecture	27
4. Les opérateurs arithmétiques et logiques	28
♦ <i>Application</i>	32
Chapitre 3 : Les conditions	35
1. Définition et utilité	36
2. La structure If...else	36
3. La structure Switch	37
♦ <i>Application</i>	39
Chapitre 4 : Les boucles	42
1. Définition et utilité	42
2. La boucle For	42
3. La boucle While	43
♦ <i>Application</i>	44

Chapitre 5 : Les fonctions 45

1. Définition et utilité 46
2. Le type d'une fonction 46
3. Qu'est ce qu'un paramètre ? 47
4. Une fonction qui ne renvoi rien ?! 47
5. Prototype d'une fonction 48
6. Appel d'une fonction 48
7. Fonction récursive 49
- ♦ *Application* 50

Série d'exercice n°1 51

Chapitre 6 : Programmation en module 55

1. Définition et utilité 56
2. Un fichier header 56
3. Inclusion de fichier 56
4. Directives de préprocesseur 57
5. Quelques astuces 57
- ♦ *Application* 59

Chapitre 7 : Les pointeurs 61

1. Définition 62
2. Histoire d'adresses 62
3. Pointeurs et fonctions 63
4. Procédure en c ?! 64
- ♦ *Application* 65

Série d'exercice n°2 67

Chapitre 8 : Les tableaux 61

1. Définition et utilité 72
2. Lecture et écriture 72
3. Pointeurs et tableaux 73
4. Tableau multidimensionnel 73
- ♦ *Application* 74