Chapitre 4

# Le type entier

## Exercice

1. Faire un programme qui affiche la somme, la soustraction, la multiplication, le quotient et le reste de la division entière de deux entiers saisis au clavier. L’affichage sera de la forme :

10+20=30

10-20=-10

(solution : entier.c)

Chapitre 5

# Le type réel

## Exercices

1. Faire un programme qui affiche la partie entière, la partie décimale et l’arrondi entier d’un réel.

(solution : reel.c ).

2. Faire un programme qui calcule la surface et le volume d’une sphère dont le rayon est demandé à l’utilisateur.

Surface = 4 \* PI \* Rayon2.

Volume = (Rayon \* Surface)/3.

PI= 3.1415926535.

(solution : sphere.c ).

3. Faire un programme qui calcule la mensualité d’un crédit immobilier à taux fixe. Il demandera le montant du capital emprunté, le nombre d’années, et le taux annuel. La formule de calcul est :

 où

M = mensualité.

C = capital emprunté.

T = Taux mensuel = Taux annuel/12.

N = Nombre de mois = Nombre d’années \*12.

(solution : emprunt.c )

Chapitre 6

# Le type caractère

## Exercices

1. Faire un programme qui convertit une majuscule saisie par l’utilisateur, en une minuscule. Le résultat est affiché à l’écran.

(solution : caract.c ).

2. Faire un programme qui lit un nombre avec une virgule (et non un point décimal) et affiche sa racine carrée. Le programme lira des nombres positifs ayant toujours 2 chiffres avant la virgule, et deux chiffres après (12,54). La méthode à utiliser est la lecture de 5 caractères et leur traduction numérique avant d’effectuer la racine carrée.

(solution : reel\_caract.c ).

Chapitre 7

# Le type booléen

### Exercice

1. Faire un programme qui découpe une adresse IP (par exemple 134.157.15.38) en une partie réseau et une partie machine. On demandera l’adresse de la machine et le sous-masque réseau (par exemple 255.255.255.0)

(solution : sous\_masque.c ).

Chapitre 9

# Les instructions simples

## Exercice

1. Reprendre le programme de calcul de la mensualité d’un crédit, donné en exercice au chapitre 5 (emprunt.c). Ajoutez un contrôle de la saisie du capital, du taux et du nombre d’années.

Le programme rappellera les données saisies et les présentera ainsi que le résultat de la manière suivante : Largeur de 20 pour les textes cadrés à droite. Les numériques sont cadrés à droite sur le point décimal. Les entiers ont une largeur de 10, les réels possèdent deux décimales.

Capital : 500000 Euros

Durée (en mois) : 120 mois

Taux Annuel : 5.65 %

Mensualité : 5463.55 Euros

(solution : emprunt2.c )

Chapitre 10

# Les instructions composées

#### Exercices

1. Faire un programme qui lit un nombre, et qui indique s'il est entier ou réel. Dans le cas d'un réel, le programme affichera en plus, sa partie entière et sa partie décimale.

Modifiez ensuite ce programme pour qu'une erreur de saisie (frappe d’un caractère à la place d’une valeur numérique) produise un message d'erreur "Erreur de saisie" apparaisse.

(solution : ifelse1.c )

2. Faire un programme qui lit une lettre et qui indique quel est son numéro dans l'ordre alphabétique (N°1 pour 'A' ou 'a', N°2 pour 'B' ou 'b', …). Il affichera le message "ce n'est pas une lettre de l'alphabet" si ce caractère n'est ni une majuscule, ni une minuscule.

(solution : ifelse2.c )

#### Exercices

1. Faire un programme qui calcule la moyenne pondérée et la mention d'un étudiant. Le programme lit trois notes et trois coefficients (n1, n2, n3 et c1, c2, c3 sont des réels) et affiche:

Votre note finale est : 14.6

Votre mention est : Bien

Les mentions sont : Recalé (inférieur à 10), Passable (de 10 à 12), Assez Bien (de 12 à 14), Bien (de 14 16), Très Bien (de 16 à 20).

La formule pour la moyenne pondérée est : ((n1\*c1)+(n2\*c2)+(n3\*c3))/(c1+c2+c3)

Un étudiant aura la mention supérieure si sa note est proche à 0.5 près de la note permettant de l'obtenir (11.5 donnera la mention correspondant au 12)

(solution : moyenne\_mention.c )

2. Modifiez le programme menu2.c pour avoir comme choix possibles 1, 2, 3, et q (pour quitter), à la place de 10, 20, 30 et 0.

(solution : menu3.c )

#### Exercices

1. Faire un programme qui décompose un entier en base 16. Il affichera les unités hexadécimales, les dizaines, les centaines et ainsi de suite. Voici un exemple d'exécution du programme sur le nombre décimale 2500, qui s'écrie 9C4 en hexadécimal:

Entrez un nombre entier : 2500

Décomposition en base 16 :

4

C

9

2. Généraliser le programme précédent pour qu’il travaille sur n’importe quelle base comprise entre 2 et 36 (demandez la base de conversion au clavier).

(solution : base1.c )

3. Faire un programme qui demande un caractère Majuscule, le convertit en minuscule, et recommence TANT QUE le caractère ‘ Q ’ n’est pas tapé.

(solution : bouclesaisie.c )

4. Faire un programme qui affiche le résultat d’une expression arithmétique contenant des additions et des soustractions. Le programme lira chaque élément caractère par caractère, retrouvera la valeur numérique, et fera les opérations lorsqu'un signe '+' ou '- sera rencontré. La boucle générale sera une boucle de lecture d'un caractère TANT QUE la fin de ligne n'est pas atteinte.

Les expressions auront la forme : 102+28-329-10+124

(solution : calculateur.c )

#### Exercice

1. Faire un programme qui dessine le début d’un sapin (triangle) selon le nombre de lignes (N) saisi au clavier. Le triangle sera centré au milieu de l’écran. On tracera des espaces et des étoiles sur chaque ligne puis on passera à la ligne suivante.

N=3 N=4 N=5

\* \* \*

\*\*\* \*\*\* \*\*\*

\*\*\*\*\* \*\*\*\*\* \*\*\*\*\*

\*\*\*\*\*\*\* \*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

(solution : triangle.c )

Chapitre 11

# Les tableaux

## Exercices

1. Faire un programme qui mémorise 10 notes d’examen dans un tableau, et qui retire la plus faible. Le programme fera au préalable la saisie des 10 valeurs réelles.

L’algorithme consiste à localiser la note la plus faible, puis à décaler les éléments à droite de cette position, d’une case vers la gauche.

(solution : retrait\_faible\_note.c )

2. Faire un programme qui range dans un tableau N valeurs entières entrées au clavier (le nombre N est demandé à l’utilisateur), puis qui les trie par ordre croissant.

Si les valeurs saisies sont 10, -20, 48, 98, 100, 1, 0 (rangées dans les cases 0, 1, 2, 3, 4, 5 et 6), le tableau contiendra après le tri les valeurs -20, 0, 1, 10, 48, 98, 100 (dans les cases 0, 1….). Le principe de l’algorithme est de faire deux boucles imbriquées. La première parcourt le tableau de la première à la dernière case. Pour chaque valeur du compteur de cette boucle, la boucle interne la fait remonter tant qu’elle est inférieure à la valeur précédente. Cet algorithme est un *tri par bulle*.

(solution : tri\_tableau.c)

Chapitre 12

# Les chaînes de caractères

## Exercices

1. Modifier le programme base1.c (chapitre 10 sur les boucles while) pour obtenir la *conversion* d’un nombre décimal en n’importe quelle base de 2 à 36

Rappel : le programme base1.c décompose en unités, dizaines, centaines, un nombre donné. Pour 2500 décomposé en hexadécimal, il affiche 4 (unités), C (dizaines) et 9 (centaines). La modification consiste à afficher le résultat 9C4. On utilisera la gestion de chaînes.

(solution : base2.c )

2. Faire un programme qui convertit chaque lettre d’un nom en majuscules.

(solution : nom\_en\_majuscules.c )

3. Faire un programme qui trie selon l'ordre ASCII, les caractères d'une chaîne. On utilisera une seconde chaîne qui sera remplie au fur et à mesure. Chaque caractère de la chaîne de départ sera inséré dans la seconde chaîne "à la bonne place". Cet algorithme se nomme un *tri par insertion*.

(solution : tri\_chaine.c )

Chapitre 13

# Les structures

## Exercice

1. Faire un programme qui range une liste d'articles de sports dans un tableau. Un article devra être défini comme une structure comprenant 3 champs : un libellé (chaîne), un prix de vente (réel) et la quantité en stock (entier).

Le programme demandera ensuite un prix de référence, et affichera le nom de l'article dont le prix s'en approche le plus.

(solution : article.c )

Chapitre 14

# Les unions

## Exercice

1. Faire un programme qui découpe une adresse IP (par exemple 134.157.15.38) en une partie réseau et une partie machine. On demandera l’adresse IP de la machine et le sous-masque réseau (par exemple 255.255.255.0). Ce programme est une modification de l’exercice sous\_masque.c du chapitre 7.

L’union contiendra deux membres : un entier sur 32 bits, et un tableau de 4 caractères (4 octets). Les valeurs saisies sont rangées dans les octets. Le filtrage s’effectue sur l’entier.

Chapitre 15

# Les fichiers

### Exercices

1. Faire un programme qui convertit un fichier texte UNIX (le caractère fin de ligne est un "Line Feed") en un fichier texte Windows (le caractère fin de ligne est un "Carriage Return"). Ce programme donnera le nombre de lignes, de mots (séparés par des espaces) et de caractères.

(solution : word\_count.c ).

2. Faire un programme qui charge le fichier d'élèves produit par le Listing 15.1 dans un tableau d'élèves. Il poursuit ensuite la saisie de nouveaux élèves dont les informations sont entrées au clavier. Ces nouvelles données doivent compléter le tableau. Une fois la saisie terminée, le programme devra écrire (en "écrasement") le fichier de départ avec la totalité des données (anciennes et nouvelles).

(solution : ajout\_eleves.c ).

### Exercice

3. Faire un programme qui convertit le fichier texte produit par le Listing 15.1 en un fichier binaire. Le programme chargera le fichier dans un tableau qui sera écrit en une seule fois.

(solution : conv\_fichier\_eleves.c ).

Chapitre 16

# Les procédures et les fonctions

## Exercices

1. Modifier le Listing 16.11 afin d'ajouter deux nouveaux choix au menu.

Le premier choix appellera une procédure de chargement d'un fichier texte dans le tableau des élèves. Le chargement rangera le premier élément du fichier dans la première case de libre du tableau. Le fichier contiendra sur chaque ligne un nom, prénom et age séparés par des espaces.

Le second choix appellera une procédure de sauvegarde du tableau des élèves, qui écrira le fichier d'origine en "écrasement".

Modifier la procédure de recherche d'un élève pour qu'elle retourne le *numéro* de la case du tableau où il se trouve.

(solution : gestion\_liste\_eleves.c)

2. Faire un programme qui utilise deux fonctions de traitement de chaînes de caractères, concatenation() et monindex(). Il demande deux chaînes et les concatène en utilisant la première fonction. Il demande deux nouvelles chaînes et recherche l’occurrence de la deuxième dans la première grâce à la seconde fonction.

concatenation(ch1,ch2) est la réécriture de strcat(). Elle possède deux arguments de type chaînes de caractères, et retourne une chaîne qui est le résultat de leur concaténation. Contrairement à strcat() qu’il ne faut pas utiliser, les deux arguments ne sont pas modifiés.

monindex(ch,sous\_ch) est la réécriture de index(). Elle possède deux arguments de type chaînes de caractères et retourne un entier. Il indique le numéro du caractère dans la chaîne (premier argument) où est trouvée la sous-chaîne (second argument). Elle retourne -1 si sous\_ch n’est pas trouvé dans ch.

(solution : traitement\_chaine.c)

Chapitre 18

# Les pointeurs

## Exercices

1. Modifier le programme inversion\_chaine.c (chapitre 12, Listing 12.3) afin d’utiliser des pointeurs de caractères à la place du tableau de caractère. Le programme saisit une chaîne et l’inverse.

(solution : inversion\_chaine2.c )

2. Faire un détecteur de *palindrome*. Un palindrome est un mot (ou une phrase), qui se lit aussi bien à l’envers qu’à l’endroit, par exemple "radar". On se basera sur inversion\_chaine2.c. Le programme indiquera si la phrase saisie en un palindrome.

(solution : palindrome.c )

3. Faire un programme d’édition de caractères. Il demandera la phrase à éditer. Elle sera mémorisée dans une chaîne de caractères allouée dynamiquement. Le programme demandera ensuite l’action à effectuer : *(d)estruction* d’un caractère, (*i)nsertion* d’un caractère, ou *(q)uitter*. Tant que l’action est différente de *(q)uitter*, on boucle sur une nouvelle action.

Destruction : On demandera la position du caractère à détruire. Le résultat sera affiché.

Insertion : On demandera la position où il faut insérer, le caractère à insérer Le résultat sera affiché..

(solution : editeur\_phrase.c )

Chapitre 19

# Liste de structures et tableaux dynamiques

1. Modifier le programme liste\_structures.c (Listing 19.1) pour avoir une procédure de saisie et une procédure d’affichage. Le programme principal proposera un menu pour choisir la saisie ou l’affichage.

Ajouter la possibilité de modifier les informations d’un élève déjà saisi. La procédure de modification appellera une fonction de recherche qui retournera un pointeur sur la structure trouvée. Le pointeur aura la valeur NULL si aucun élève n’est trouvé. Le nom de l’élève à rechercher est fourni en argument à cette fonction.

(solution : liste\_structures2.c )

2. Faire un programme de tri d'une liste de structures d'articles. Les champs de la structure article sont, le libellé (chaîne), le prix (réel), la quantité (entier). Ils sont lus à partir d’un fichier, puis triés selon le prix (du plus petit au plus grand).

(solution : liste\_articles.c )

Chapitre 20

# Interface avec le système UNIX

#### Exercice

1. Réécrire la commande UNIX "cp". Le programme copie\_fichier.c copiera le premier fichier dans le second. Les noms des deux fichiers sont fournis sur la ligne de commande.

(solution : copie\_fichier.c )