

CHAPITRE 8

LES TYPES DE VARIABLES COMPLEXES

LES DECLARATIONS DE TYPE SYNONYMES: TYPEDEF

On a vu les types de variables utilisés par le langage C++: **char**, **int**, **float**, pointeur; le chapitre 9 traitera des fichiers (type **FILE**).

Le programmeur a la possibilité de créer ses propres types: **Il lui faut alors les déclarer** (après les déclarations des bibliothèques et les « define ») avec la syntaxe suivante:

Exemples:

```
typedef int entier;           // on définit un nouveau type "entier" synonyme de "int"

typedef int vecteur[3];      // on définit un nouveau type "vecteur" synonyme
                             // de "tableau de 3 entiers"

typedef float *fpointeur;   // on définit un nouveau type "fpointeur" synonyme */
                             // de "pointeur sur un réel"
```

La portée de la déclaration de type dépend de l'endroit où elle est déclarée: dans main(), le type n'est connu que de main(); en début de programme, le type est reconnu dans tout le programme.

```
#include <iostream.h>
```

```
typedef int entier;
typedef float point[2];
```

```
void main()
{
entier n = 6;
point xy;
xy[0] = 8.6;
xy[1] = -9.45;

etc ...

}
```

Exercice VIII 1: Afficher la taille mémoire d'un point à l'aide de l'opérateur sizeof.

Exercice VIII 2: Définir le type **typedef char ligne[80];**

- a- Déclarer dans le programme principal un pointeur de ligne; lui attribuer de la place en mémoire (pour 5 lignes). Saisir 5 lignes et les afficher.
- b- Ecrire une fonction de prototype **void saisie (ligne *tx)** qui effectue la saisie de 5 lignes puis une autre fonction de prototype **void affiche (ligne *tx)** qui les affiche. Les mettre en oeuvre dans le programme principal

LES STRUCTURES

Le langage C ++ autorise la déclaration de types particuliers: les structures. Une structure est constituée de plusieurs éléments de même type ou non (appelés **champs**) .

Exemple:

Déclaration:

```
typedef struct          // On définit un type struct
{
  char nom[10];
  char prenom[10];      // On a ici 4 champs
  int age;
  float note;
}
fiche;
```

Utilisation:

On déclare des variables par exemple: **fiche f1,f2;**

puis, par exemple: **strcpy(f1.nom,"DUPONT");**
 strcpy(f1.prenom,"JEAN");
 f1.age = 20;
 f1.note = 11.5;

L'affectation globale est possible avec les structures: on peut écrire: **f2 = f1;**

Exercice VIII 3:

- a- Déclarer la structure ci-dessus, saisir une fiche, afficher ses champs.
- b- Même exercice mais en créant une fonction de prototype **void saisie(fiche &fx)** et une fonction de prototype **void affiche(fiche fx)**

STRUCTURES ET TABLEAUX

On peut définir un tableau de structures (mais ceci est assez peu utilisé) :

Exemple: (à partir de la structure définie précédemment)

Déclaration: **fiche f[10]; /* on déclare un tableau de 10 fiches */**

Utilisation: **strcpy(f[i].nom,"DUPONT") /* pour un indice i quelconque */**
 strcpy(f[i].prenom,"JEAN");
 f[i].age = 20;
 f[i].note = 11.5;

Exercice VIII 4: Créer une structure **point{int num;float x;float y;}**

Saisir 4 points, les ranger dans un tableau puis les afficher.

STRUCTURES ET POINTEURS

On peut déclarer des pointeurs sur des structures. Cette syntaxe est très utilisée en langage C++, en raison des possibilités d'allocation dynamique de mémoire.

Un symbole spécial a été créé pour les pointeurs de structures, il s'agit du symbole ->

Exemple: (à partir de la structure définie précédemment)

Déclaration: **fiche *f; // on déclare un pointeur de fiche**

Utilisation: **f = new fiche; // réserve de la place pour une fiche**
 strcpy(f->nom,"DUPONT");
 strcpy(f->prenom,"JEAN");
 f->age = 20;
 f->note = 11.5;
 delete fiche ; // si on a terminé

Autre exemple :

```
f = new fiche[5*sizeof(fiche)]; // réserve de la place pour 5 fiches  
int i ;  
for(i=0 ;i<5 ;i++)  
{
```

```

        cin>>(f+i)->nom ;
        cin>>(f+i)->prenom ;
        cin>>(f+i)->age ;
        cin>>(f+i)->note ;
    }
    delete fiche ; // si on a terminé

```

Exercice VIII 5:

- a- Reprendre l'exercice VIII_4 en utilisant la notation « pointeur »
- b- Même exercice mais en créant une fonction de prototype **void saisie(point *px)** et une fonction de prototype **void affiche(point *px)**

CORRIGE DES EXERCICES

Exercice VIII 1:

```

#include <iostream.h>
#include <conio.h>

typedef float point[2];

void main()
{
    cout<<"TAILLE D'UN POINT: "<<sizeof(point)<<"\n";
    cout<<"\nPOUR SORTIR FRAPPER UNE TOUCHE ";
    getch();
}

```

Exercice VIII 2a:

```

#include <iostream.h>
#include <conio.h>

typedef char ligne[80];

void main()
{
    ligne *texte;
    int i;
    texte = new ligne[sizeof(ligne)*5]; // reserve de la place pour 5 lignes

    cout<<"\n          SAISIE DU TEXTE\n\n";
    for (i=0;i<5;i++)

```

```

    {
        cout<<"LIGNE Num "<<i<<"\n";
        cin>>texte[i]; // saisie de la ligne
    }

    cout<<"\n\n          AFFICHAGE DU TEXTE\n\n";
    for(i=0;i<5;i++)cout<<texte[i]<<"\n";
    delete texte;
    cout<<"\nPOUR SORTIR FRAPPER UNE TOUCHE ";
    getch();
}

```

Exercice VIII 2b :

```

#include <iostream.h>
#include <conio.h>

```

```

typedef char ligne[80];

```

```

void saisie (ligne *tx)
{
    int i;
    cout<<"\n          SAISIE DU TEXTE\n\n";
    for (i=0;i<5;i++)
    {
        cout<<"LIGNE Num "<<i<<"\n";
        cin>>tx[i]; // saisie de la ligne
    }
}

```

```

void affiche(ligne *tx)
{
    int i;
    cout<<"\n\n          AFFICHAGE DU TEXTE\n\n";
    for(i=0;i<5;i++)cout<<tx[i]<<"\n";
}

```

```

void main()
{
    ligne *texte;
    texte = new ligne[sizeof(ligne)*5]; // reserve de la place pour 5 lignes
    saisie(texte);
    affiche(texte);
    delete texte;
    cout<<"\nPOUR SORTIR FRAPPER UNE TOUCHE ";
    getch();
}

```

Exercice VIII 3a:

```
#include <iostream.h>
#include <conio.h>

typedef struct
{
    char nom[10];
    char prenom[10];
    int age;
    float note;
}
fiche;

void main()
{
    fiche f;
    cout<<"SAISIE D'UNE FICHE \n";
    cout<<"NOM: ";cin>>f.nom;
    cout<<"PRENOM: ";cin>>f.prenom;
    cout<<"AGE: ";cin>>f.age;
    cout<<"NOTE: ";cin>>f.note;
    cout<<"\n\nLECTURE DE LA FICHE:\n";
    cout<<"NOM: "<<f.nom<<" PRENOM: "<<f.prenom<<" AGE: "<<f.age<<" NOTE:
"<<f.note;
    cout<<"\n\nPOUR SORTIR FRAPPER UNE TOUCHE ";
    getch();
}
```

Exercice VIII 3b:

```
#include <iostream.h>
#include <conio.h>

typedef struct
{
    char nom[10];
    char prenom[10];
    int age;
    float note;
}
fiche;

void saisie(fiche &fx)
{
    cout<<"SAISIE D'UNE FICHE \n";
```

```

cout<<"NOM: ";cin>>fx.nom;
cout<<"PRENOM: ";cin>>fx.prenom;
cout<<"AGE: ";cin>>fx.age;
cout<<"NOTE: ";cin>>fx.note;
}

void affiche(fiche &fx)
{
cout<<"\n\nLECTURE DE LA FICHE:\n";
cout<<"NOM: "<<fx.nom<<" PRENOM: "<<fx.prenom<<" AGE: "<<fx.age<<" NOTE:
"<<fx.note;
}

void main()
{
fiche f;
saisie(f);
affiche(f);

cout<<"\n\nPOUR SORTIR FRAPPER UNE TOUCHE ";
getch();
}

```

Exercice VIII 4:

```

#include <iostream.h>
#include <conio.h>

typedef struct {int num;float x;float y;} point;

void main()
{
point p[4];    // tableau de points
int i;

// saisie
cout<<"SAISIE DES POINTS\n\n";
for(i=0;i<4;i++)
{
cout<<"\nRELEVE N° "<<i<<" : \n";
p[i].num = i;
cout<<"X= ";cin>>p[i].x;
cout<<"Y= ";cin>>p[i].y;
}

// relecture
cout<<"\n\nRELECTURE\n\n";

```

```

for(i=0;i<4;i++)
{
    cout<<"\nRELEVE N° "<<p[i].num<<" :";
    cout<<"\nX= "<<p[i].x;
    cout<<"\nY= "<<p[i].y;
}
cout<<"\n\nPOUR SORTIR FRAPPER UNE TOUCHE ";
getch();
}

```

Exercice VIII 5a :

```

#include <iostream.h>
#include <conio.h>

```

```

typedef struct {int num;float x;float y;} point;

```

```

void main()
{
    point *p;    // pointeur de point
    int i;

```

```

    p = new point[4*sizeof(point)]; //reservation de place

```

```

// saisie
cout<<"SAISIE DES POINTS\n\n";
for(i=0;i<4;i++)
{
    cout<<"\nRELEVE N° "<<i<<" :\n";
    (p+i)->num = i;
    cout<<"X= ";cin>>(p+i)->x;
    cout<<"Y= ";cin>>(p+i)->y;
}

```

```

// relecture
cout<<"\n\nRELECTURE\n\n";
for(i=0;i<4;i++)
{
    cout<<"\nRELEVE N° "<<(p+i)->num<<" :";
    cout<<"\nX= "<<(p+i)->x;
    cout<<"\nY= "<<(p+i)->y;
}

```

```

delete p; // liberation de la place
cout<<"\n\nPOUR SORTIR FRAPPER UNE TOUCHE ";
getch();
}

```


Exercice VIII 5b :

```
#include <iostream.h>
#include <conio.h>

typedef struct {int num;float x;float y;} point;

// saisie
void saisie(point *px)
{
    int i;
    cout<<"SAISIE DES POINTS\n\n";
    for(i=0;i<4;i++)
    {
        cout<<"nRELEVE N° "<<i<<" :n";
        (px+i)->num = i;
        cout<<"X= ";cin>>(px+i)->x;
        cout<<"Y= ";cin>>(px+i)->y;
    }
}

// relecture
void affiche(point *px)
{
    int i;
    cout<<"\n\nRELECTURE\n\n";
    for(i=0;i<4;i++)
    {
        cout<<"nRELEVE N° "<<(px+i)->num<<" :";
        cout<<"nX= "<<(px+i)->x;
        cout<<"nY= "<<(px+i)->y;
    }
}

void main()
{
    point *p;    // pointeur de point

    p = new point[4*sizeof(point)]; //reservation de place

    saisie(p);
    affiche(p);

    delete p;    // liberation de la place
    cout<<"\n\nPOUR SORTIR FRAPPER UNE TOUCHE ";
    getch();
}
```

}