

[プログラミング入門I 6回目 2003・10・2] 2次元配列

2次元配列(多次元配列)

配列とは、データを並べて、何番目か(インデックス)で参照するためのデータ型です。これがないと、プログラム中で何番目かを計算して、要素を取り出したり、格納したりするプログラムが書くことができません。この配列があれば、いろいろなものをプログラム中で表すために使うことができます。

たとえば、数列を配列と見立てるのは非常に自然な考え方です。数列の a_0 を $a[0]$, a_1 を $a[1]$, a_2 を $a[2]$, ... というように表します。フィボナッチ数列を計算するプログラムを考えてみましょう。フィボナッチ数列とは、

$a_0=1, a_1=1$ で始まって、 $a_{n+1}=a_n+a_{n-1}$ で生成されるプログラムです。次のプログラムは、 a_{10} を計算するプログラムです。10まで計算するには、配列が11なくてはならないことに注意してください。

```
main(){
    int a[11];
    int i;
    a[0] = 1;
    a[1] = 1;
    for(i=2; i <=10; i++)
        a[i] = a[i-2]+a[i-1];
    printf("a10 is %d\n",a[10]);
}
```

また、配列で表現できるものに、集合や表があります。配列を集合と見立てて考えてみましょう。右のプログラムが、

```
#include <stdio.h>
main(){
    int A[10];
    int I,flag;
    for(I=0; I < 10; I < 10)
        scanf("%d",&A[I]);
    flag = 0;
    for(i=0; i < 10; i++)
        if(a[i] == 3) {
            flag = 1;
            break;
        }
    if(flag) printf("3 is found\n");
    else printf("3 is not found\n");
}
```

配列に整数を入力して、その中に3があるかどうかを判定するプログラムです。このプログラムはいわば、数字の集合を入力して、その中に3がはいっているかどうかを判定しているわけです。

では、表はどうでしょうか。表は、たて横にマス目があって、何行何列というアクセスをするためのデータの形(データ構造)です。これも、配列を使ってあらわすことができます。配列はいわばマス目が1列に並んで、何番目と指定してアクセスできるデータの形ですから、たとえば、横 N 行、縦 M 列の表を作りたいときには、配列として、 $N \times M$ のサイズの配列を作り、 i 行 j 列にアクセスしたい時には $M*i+j$ 番目の要素にアクセスすればいいことになります。たとえば、10行20列の表 T の場合、 i 行 j 列は以下のようにして書けます。

```
int T[10*20]; x = T[i*20+j]
```

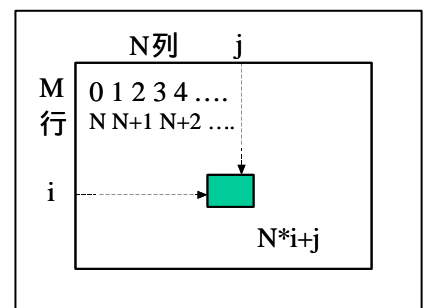
しかし、いちいちこのようにして要素の位置を計算して書くのは面倒なので、C 言語では2次元配列を次のように宣言することができます。

基本データ型 配列の名前[行のサイズ][列のサイズ];

配列への参照は、

配列名[行の位置][列の位置]

ここで、前の配列と同じように、何番目かを数えるのに、0から数えますに注意してください。上の例では、



```
int T[10][20];
```

と宣言した配列を参照するには以下のように書くことができます。

```
x=T[i][j];
```

これで、要素を参照するときにはいちいち列や行のサイズを気にする必要がなくなります。

上では行と列だけ、つまり 2 次元の配列を考えましたが、多次元にも拡張できます。

基本データ型 配列の名前 [1 次元目のサイズ] [2 次元目のサイズ] [3 次元目のサイズ] ... ;

参照も同様です。

文字列の配列

文字列とは、文字の列で、文字の 1 次元配列で表されることは前に述べました。ただし、文字列といった場合、単なる文字の列ではなくて、C 言語では文字 0 で終わる文字列であるということは注意してください。

文字列は文字の 1 次元配列ですから、文字列の配列は文字の 2 次元配列でされることになります。

```
char 文字列の配列の名前 [配列のサイズ] [文字列の長さ];
```

文字列の長さは、正確にはこの配列に格納する文字列の最大の長さです。たとえば、

```
char S[10][20];
```

は、最大 20 文字の文字列を 10 コ格納する配列です。これで、`S[i][j]`と書くと `i` は何番目の文字列か、`j` は、その文字列の文字の位置をすることになり、その文字が参照されます。文字列自体を参照するときには、`S[i]`と書きます。しかし、文字列同士の代入は変数のように書くことはできません。一文字一文字代入してもいいのですが、このようなときにはライブラリ関数 `strcpy` を使います。`strcpy`(文字列 1, 文字列 2)は、文字列 1 に文字列 2 をコピーします。たとえば、同じサイズの文字列 `w[20]`を宣言しておいて、ここに `i` 番目の文字列をコピーするには、次のようにします。

```
strcpy(w,s[i]);
```