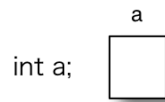


## ■ 配列

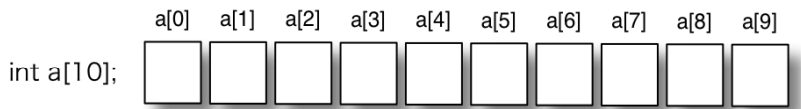
### □ 配列変数の宣言

関連性のある同じ型の複数の値を扱うために「配列変数」と呼ばれる仕組みがあります。プログラム中では変数名の後ろに [ ] をつけることで通常の変数と区別して表記します。

int 型の変数として a を宣言するには右図のように int a; と宣言します。これによって一つだけ、整数を代入できる領域が確保されます。



これに対して、int a[10]; と宣言することで、int 型の数値を 10 個扱うための配列変数が宣言できます。



このとき 10 個の整数を代入できる領域が確保され、それぞれの領域は [ ] をつけて番号で指定します。上の例では a[0], a[1], a[2], a[3], a[4], a[5], a[6], a[7], a[8], a[9] と 10 個用意されました。この一つ一つの領域のことを要素と呼び、要素の番号のことを添え字（あるいはインデックス）と呼びます。添え字は 0 から始まっているため、10 個宣言した配列変数の最後の要素が a[10] ではなく a[9] であることに注意してください。

(興味のある受講生は教科書 p.97, 6.1 「1次元配列」を参照)

### □ 簡単な利用例

乱数を使って 10 個の要素にそれぞれランダムな数値を代入し、後で表示するプログラムを右に示します。

ポイント：

- rand() は疑似乱数関数 (stdlib.h の include が必要)
- 最初に srand() 関数を一度だけ呼び出す必要あり。srand に与える引数はランダムな数値。Seed (種) と言う。
- rand() 関数は 0 から 2147483647 までの整数を返す。

```
int main() {
    int i, a[10], seed;

    printf("random seed = ");
    scanf("%d",&seed);
    srand(seed);

    /* 各要素にランダムな数を代入 */
    for(i=0; i<10; i++) {
        a[i]=rand();
    }

    /* 各要素をプリント */
    for(i=0; i<10; i++) {
        printf("a[%d] = %d\n", i, a[i]);
    }

    return 0;
}
```

i を 0 から 9 まで変化させながら 10 回ループさせ、a[i] = rand(); とすることで配列変数の各要素に数値が代入されています。実行して 10 行のランダムな数値が表示されることを確認してください。

## □ 課題 1.

下記のプログラム `snow.c` を取得して実行し、その結果を確認してください。特にアニメーションをどのようにして実現しているかに注目して下さい。

ポイント：

- `while()` によってループするたびに、
- `gclr()` によって画面を消去し、
- 少し下の場所に雪粒を描き直し、
- `msleep()` によって少しだけ待つ。(引数に指定する待ち時間の単位はミリ秒)

```
int main() {
    int win, seed;
    double x, y, w;
    ..... (中略) .....

    while(1) {
        gclr(win);                /* 画面を消去 */
        fillrect(win, x, y, w, w); /* 四角を描く */
        y-= w / 2.0;              /* 雪粒の大きさだけ下に移動 */
        if( y < 0.0 ) {
            x=(double)(rand() % 400); /* x 座標位置を再設定 */
            y=400.0;                /* y 座標位置は再び上端へ */
        }
        msleep(10);              /* 少し待つ */
    }

    /* このプログラムは終了しないので return 0; などは無し */
}
```

このプログラムを配列変数を使うように修正し、雪粒を 10 個に増やして下さい。

ヒント：

- `x, y` をそれぞれ配列変数として各雪粒の位置を 10 セット記憶できるようにします。
- 初期位置や下端に到達した判定、`fillrect()` による描画など、雪粒に関するすべての処理をループを使って各要素ごとに行います。

## □ 課題 2.

跳ね返る球のアニメーションを描くプログラム `pingpong.c` を取得して実行し、動作を確認してください。そこでは一つだけ球が動いていると思います。これを右図のように複数の球が跳ね返るように修正してください。

余力のある受講生は球の色を変えるなり、もっと多くの球を出すなり、工夫してより面白いアニメーションを作ってください。

