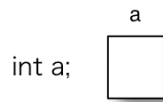


## ■ 配列

### □ 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（インデックス）と呼びます。インデックスは 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。

### □ 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：

- ・ 疑似乱数関数 rand( ) のために 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 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。

□ 練習問題 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、
- GWclear() によって画面を消去し、
- 少し下の場所に雪粒を描き直し、
- GWSleep() 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() {
    float x, y, w=10.0;
    int c=10, seed;
    ..... (中略) .....

    x=10.0; y=200.0;
    while(1) {
        GWclear(-1); /* 画面を消去 */
        GWscircle(x, y, x+w, y+w, c); /* 雪粒を描く */
        y-=w/2.0; /* 雪粒の半分の大きさだけ下に移動 */
        if(y<10.0) { /* 下まで届いたら... */
            x=(float)(rand()%200); /* 横位置はランダム */
            y=200.0; /* 縦位置は上端に戻す */
        }
        GWSleep(30);
    }

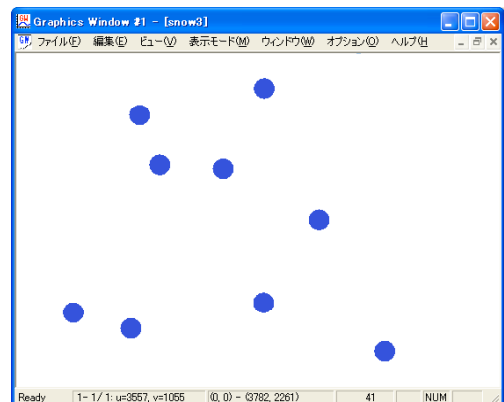
    /* このプログラムは終了しないので return 0; などは無し */
}
```

このプログラムを配列変数を使うように修正し、右図のように雪粒を 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、GWscircle() 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。

□ 練習問題 2.



過去に作成した跳ね返る球のアニメーションを描くプログラムについて、一つだけ四角が動いていたと思います。これを複数の四角が跳ね返るように修正してください。

衝突するたびに色を変えるなり、もっと多くの四角を出すなり、工夫してより面白いアニメーションを作ってください。