

C言語による初級プログラミング 第4回 繰り返しと配列

塩浦 昭義 (東北大学大学院 情報科学研究科)

平成 22 年 7 月 14 日/15 日

◎ 配列の使い方

同様な変数をたくさん使いたいときに「配列」を使います。

- プログラムの最初に `int num[5]` と書く
⇒ 整数型の 5 つの変数 `num[0]`, `num[1]`, `num[2]`, `num[3]`, `num[4]` が使えます
- `int a[10]` と書く ⇒ 整数型の 10 個の変数 `a[0]`, `a[1]`, ..., `a[9]` が使えます
(番号が 0 から始まることに注意!)

◇ 例 1: `score.c` — 10 人分の数学の試験結果 (0 点~ 100 点) を入力すると, 合格者 (60 点以上) の番号が出力される。

```
#include <stdio.h>

main()
{
    int math[10];
    int i;

    i = 0;
    while (i <= 9) {
        printf("Input score No. %d ", i);
        scanf("%d", & math[i]);
        i = i + 1;
    }
    printf("Math Exam---passed: ");
    i = 0;
    while (i <= 9) {
        if (math[i] >=60) {
            printf(" No. %d ", i);
        }
        i = i + 1;
    }
    printf("\n");
}
```

解説

⇐ `int math[10]` により, 10 個の変数 `math[0]`, `math[1]`, ..., `math[9]` を用意する。

⇐ キーボードから数字を入力し, 配列の一つ `math[i]` に覚えさせるときは `scanf("%d", & math[i])` とする。

⇐ `math[i]` の値を順番に調べ, 値が 60 以上ならば `i` の値を出力する。

⇐ 見栄えを良くするために, 最後は改行する。

演習問題 1:(レポートのプログラム名は `report4-1.c`) プログラム `score.c` を次のように変更せよ。

(i) 数学だけでなく, 英語の試験の得点を入力できるようにせよ。さらに, 数学と英語の得点の合計が 130 点以上だったら合格とし, 合格者の番号を出力するプログラムに修正せよ。

(ii) 試験の得点を入力する際, 入力値が 0 点~ 100 点の範囲外の場合, 再入力させるようにせよ。(ヒント: 入力値が 0 点~ 100 点の範囲外の場合, `i = i + 1` を実行しなければよい)

◇ 例 2: max.c — 配列に 5 個の整数を入力し, その中の最大値を配列の最後に移動させる.

```
#include <stdio.h>

main()
{
    int num[5];
    int i, tmp;

    i = 0;
    while (i <= 4) {
        scanf("%d", & num[i]);
        i = i + 1;
    }

    i = 0;
    while (i <= 3) {
        if (num[i] > num[i+1]) {
            tmp = num[i];
            num[i] = num[i+1];
            num[i+1] = tmp;
        }
        i = i + 1;
    }

    i = 0;
    printf("Result: \n");
    while (i <= 4) {
        printf("%d \n", num[i]);
        i = i + 1;
    }
}
```

解説

← int num[5] により, 5 個の変数 num[0], num[1], ..., num[4] を用意する.

← 5 個の値を入力する.

← 配列の前後の要素 num[i] と num[i+1] を比較し, num[i] の方が大きい場合, 2 つの値を入れ替える.

値を入れ替えるとき, 一度 num[i] の値を tmp に覚えさせて, 次に num[i] の値を num[i+1] に変更. その後, num[i+1] の値を tmp (もともとの num[i] の値) に変更.

← num[0], num[1], ..., num[4] の値を出力.

上記のプログラムでは, num[4] に最大値が入っている.

なお, num[0], num[1], ..., num[3] に入っている値の順番は, 各値が入力された順番とは無関係になっている.

演習問題 2:(レポートのプログラム名は report4-2.c) プログラム max.c を次のように変更せよ.

(i) num[4] に最大値を入れた後, 2 番目に大きい値を計算し, num[3] に入れるようにせよ. (ヒント: num[0], num[1], ..., num[3] 中の最大値が, 全体で 2 番目に大きい値となる)

(ii) [上級者向け] 入力された num[0], num[1], ..., num[3] の値を入れ替えて, 小さい方から大きい方へ並ぶようにプログラムを修正せよ. (ヒント: 3 番目に大きい値を num[2] に入れる ⇒ 4 番目に大きい値を num[1] に入れる ⇒ 5 番目に大きい値を num[0] に入れる)

◇ 例 3: number.c — 数当てゲーム. 0, 1, ..., 9 の数字の中から異なる 3 つを選び入力する. 答え (2, 9, 5) と一致していたら “All numbers are correct.” と答え, そうでない場合は一致している数字の数を答える. 例えば, “5, 9, 0” と入力したら, “2 numbers are correct.” と答え, “3, 2, 1” と入力したら, “1 numbers are correct.” と答える. 数字の順番は考慮しない.

```
include <stdio.h>

main()
{
    int answer[3], num[3];
    int i, j, same;

    answer[0] = 2;
    answer[1] = 9;
    answer[2] = 5;

    printf("Input three different numbers: \n");
    i = 0;
    while (i <= 2) {
        scanf("%d", & num[i]);
        i = i + 1;
    }

    i = 0;
    same = 0;
    while (i <= 2) {
        j = 0;
        while (j <= 2) {
            if (num[i] == answer[j]) {
                same = same + 1;
            }
            j = j + 1;
        }
        i = i + 1;
    }
    if (same == 3) {
        printf("All numbers are correct!\n");
    } else {
        printf("%d numbers are correct.\n", same);
    }
}
```

解説

← answer[0], answer[1], answer[2] に答え 2, 9, 5 を入れておく.

← 0, 1, ..., 9 の数字を 3 つ入力させ, num[0], num[1], num[2] に入れておく.

← answer[0], answer[1], answer[2] の 3 つの数字と, num[0], num[1], num[2] の 3 つの数字の中に, 共通するものがいくつ存在するか数え, 変数 same に入れる. このためには, num[i] == answer[j] を満たす i, j の数を数えれば良い.

最初は num[0] を answer[0], answer[1], answer[2] の 3 つの数字と比較し, 次に num[1] を answer[0], answer[1], answer[2] と比較し, 最後に num[2] を answer[0], answer[1], answer[2] と比較している.

← same が 3 であれば全部正解とのメッセージを出力し, それ以外の場合は same の値を出力する.

演習問題 3:(レポートのプログラム名は **report4-3.c**) プログラム number.c を次のように変更せよ。
(i) 現在のプログラムでは、入力された数字の順番は考慮していないが、数字の順番も考慮して正誤を判定するプログラムに修正せよ。例えば、“5, 2, 9” と入力したら、“0 numbers are correct.” と答え、“5, 9, 2” と入力したら、“1 numbers are correct.” と答え、“2, 9, 5” と入力したら、“All numbers are correct.” と答える。
(ii) 正解を入力するまで、3桁の数を繰り返し入力できるように修正せよ。

今日のレポート問題: 演習問題 1, 2, 3 を解きなさい。
それぞれ問題ごとに1つずつプログラムを提出すること。
締め切り : 7月 21 日 (経済学部), 7月 22 日 (法学部) 授業時間終了時まで