

# 1 配列

## 学習のポイント

表データを扱うのに便利な二次元配列について学びます。

番号	名前/科目	国語	算数	英語
1	石原和男	80	65	70
2	河野健	70	55	75
3	鈴木二郎	75	80	70
4	藤山重男	80	60	60
5	渡辺太郎	90	80	75

上のような表は、生徒と科目という2つの要素から成っており、これを配列として表現するには添え字が2つから成る二次元配列を用いて次のように宣言します。

```
int ten[5][3];
```

Cの配列は基底が0から始まっているため番号1の生徒のデータは0行の要素に入ることにご注意してください。

二次元配列も次のように宣言時にデータの初期化を行うことができます。

```
int ten[5][3] = {{80, 65, 70},  
                {70, 55, 75},  
                {75, 80, 70},  
                {80, 60, 60},  
                {90, 80, 75}};
```

初期化データは行単位でで囲み、カンマ(,)で区切ります。初期化データがある場合は、配列の第1要素の指定を宣言時行わなくても、コンパイラが自動的にデータを数えて配列の要素数を設定してくれます。しかし、第2要素数以降は必ず指定しなければなりません。

「C言語」(河西朝雄著 ナツメ社)74頁

## 2 例題 14

3 教科 5 人の生徒の得点を `ten[][]` に初期化データとして与えておき、横 (生徒) の合計を求めて表示しなさい。

```
/*
  例題 14 C 言語 75 頁
  3 教科 5 人の生徒の得点を ten[][] に初期化データとして与えておき、横 (生
  徒) の合計を 求めて表示しなさい。
  reidai14.c
*/

#include <stdio.h>

int main()
{
  int ten[5][3] = {{80, 65, 70},
                  {70, 55, 75},
                  {75, 80, 70},
                  {80, 60, 60},
                  {90, 80, 75}};

  int j;
  int k;
  int s;

  printf("   国語   数学   英語   |   合計\n");
  printf("-----\n");
  for(j = 0; j < 5; j++){
    s = 0;
    for(k = 0; k < 3; k++){
      printf("%8d", ten[j][k]);
      s = s + ten[j][k];
    }
    printf("   |%8d\n", s);
  }

  return 0;
}
```

「C 言語」(河西朝雄著 ナツメ社)75 頁

上記のプログラムの修正を行ってください。

1. 生徒ごとの平均点を表示してください。
2. 科目ごとの平均点を表示してください。

### 3 連立方程式の数値解法

連立方程式の解法 ヤコビ法

```
/*
  連立方程式の解法
  ヤコビ (Jacobi) 法
  C 言語と基礎数値計算 59 頁 柴田優 工学図書株式会社
  jacobi.c
*/

#include <stdio.h>
#include <math.h>

#define N 3
#define MAX 100
#define EPS 1.0e-5

void JACOBI(int *no, double ma[], double mb[], double mx[]);

int main()
{
  int i, j;
  int num;

  double ma[N*N] = {3.0, -6.0, 9.0,
                    2.0, 5.0, -8.0,
                    1.0, -4.0, 7.0};

  double mb[N] = {6.0, 8.0, 2.0};
  double mx[N];
```

```

printf("\n-----配列 ma と-----");
printf("配列 mb の内容-----\n\n");

for(i = 0; i < N; i++){
    for(j = 0; j < N; j++){
        printf("%10.5lf", ma[N * i + j]);
    }
    printf("    %10.5lf\n", mb[i]);
}

JACOBI(&num, ma, mb, mx);
printf("\n-----ヤコビ法による");
printf("方程式の解 mx--\n\n");
printf("    繰り返し計算の回数 = %d\n", num);
for(i = 0; i < N; i++){
    printf("    mx[%d] = %8.5lf \n", i + 1, mx[i]);
}

return 0;
}

void JACOBI(int *no, double ma[], double mb[], double mx[])
{
    int    i, j, k, ii;
    double error, aa, xx[N];

    *no = 0;
    for(i = 0; i < N; i++){
        mx[i] = 0.0;
    }

    for(i = 0; i < MAX; i++){
        *no += 1;
        error = 0;
        for(ii = 0; ii < N; ii++){
            xx[ii] = mx[ii];
        }
        for(j = 0; j < N; j++){
            aa = mb[j];

```

```
for(k = 0; k < N; k++){
    if(k != j){
        aa -= ma[N * j + k] * xx[k];
    }
    mx[j] = aa / ma[N * j + j];
    error += fabs(mx[j] - xx[j]);
}
if(error < EPS){
    break;
}
}
}
```