14.3. 多次元配列

添え字を複数個指定できる配列のことを「多次元配列」と呼びます。
添え字が 2 つなら二次元配列、3 つなら三次元配列となります。

二次元配列の使用例

章の始めのサンプルプログラムでは、英語のテストの成績を管理するための配列を定義しました。
二次元配列は英語だけではなく、数学のテストの成績も管理したいとなった場合に使用します。

二次元配列を使用して 30 人分のテスト(英語、数学)の成績を管理するプログラムを以下に示します。

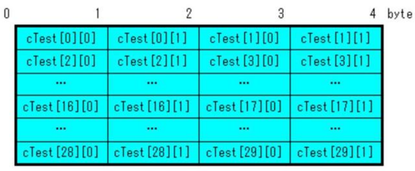
コード

```
#include <stdio.h>

int main()
{
   char cTest[30][2];
   int iIndex;

   /*
    * テストの点数を入力
    */
   for (iIndex = 0; 30 > iIndex; iIndex++) {
      /* 英語のテストの点数 */
      scanf("%d", &cTest[iIndex][0]);

      /* 数学のテストの点数 */
      scanf("%d", &cTest[iIndex][1]);
   }

   return 0;
}
```

上記コードでは以下のように二次元配列が宣言されています。

```
char cTest[30][2];
```

このような場合、配列はメモリ領域上に次のように配置されます。



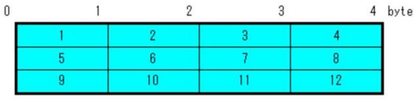
メモリ領域上には cTest[0][0], cTest[0][1] ・・・ cTest[16][0] ・・・ cTest[28][0] ・・・ cTest[29][1] の順に連続して 60 バイト確保されます。

二次元配列の初期化

配列の先頭から初期化する場合は以下のように記述します。

```
char cData[4][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
```

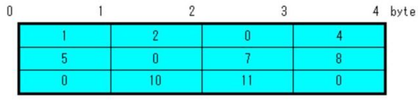
このような場合、配列の初期値はメモリ領域上に次のように配置されます。



次元単位で初期化する場合は以下のように記述します。

```
char cData[4][3] = {{1, 2,}, {4, 5,}, {7, 8,}, {10, 11,}};
```

このような場合、配列の初期値はメモリ領域上に次のように配置されます。

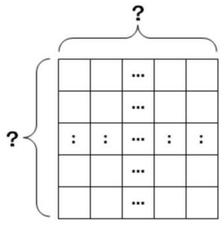


二次元配列の要素数

二次元配列の初期化では次のように要素数を省略することはできません。

```
char cData[][] = {1, 2, 3, 4};
```

このような場合、以下のように次元単位のサイズが不明確となるため、コンパイルすることができません。



次元単位のサイズを確定するため、多次元配列の場合、**省略可能な要素数は先頭の要素数のみ**です。
コンパイルするためには以下のように先頭以外の要素数を指定する必要があります。

```
char cData[][3] = {1, 2, 3, 4};
```

このような場合、以下のサイズとして二次元配列のメモリ領域を確保します。