

プログラミング基礎

<http://bit.ly/prog2d>

多次元配列

前期 第10週

2017/6/26

【Point 1】 SUB×NUM、つまり2×5個のint型の記憶場所が用意された多次元配列testを宣言している (p. 202 図7-9)

```
1: #include <stdio.h>
2:
3: #define SUB 2
4: #define NUM 5
5:
6: int test[SUB][NUM];
7:
```



【Point 2】 多次元配列の要素を指定する場合は、添字をその次元分だけ並べて記述する

```
8: int main(void)
9: {
10:     int i;
11:
12:     test[0][0] = 80;
13:     test[0][1] = 60;
14:     test[0][2] = 22;
15:     test[0][3] = 50;
16:     test[0][4] = 75;
17:     test[1][0] = 90;
18:     test[1][1] = 55;
19:     test[1][2] = 68;
20:     test[1][3] = 72;
21:     test[1][4] = 58;
22:
```

test[0][0]	test[0][1]	test[0][2]	test[0][3]	test[0][4]
80	60	22	50	75
90	55	68	72	58
test[1][0]	test[1][1]	test[1][2]	test[1][3]	test[1][4]

【Point 3】 この繰り返し処理では、配列testの2つ目の添字を、変数iで変化させている

```
23: for(i=0; i<NUM; i++) {
24:     printf("%d番目の人の国語の点数は%dです。\\n",
           i+1, test[0][i]);
25:     printf("%d番目の人の算数の点数は%dです。\\n",
           i+1, test[1][i]);
26: }
27:
28: return 0;
29: }
```

test[0][i]
test[1][i] に対してi++をする

	test[0][0]	test[0][1]	test[0][2]	test[0][3]	test[0][4]
国語→	80	60	22	50	75
算数→	90	55	68	72	58
	test[1][0]	test[1][1]	test[1][2]	test[1][3]	test[1][4]

【練習10-1】

**サンプルプログラムをコンパイルして、
実行結果を確認しましょう。**

【課題10-1】

サンプルプログラム（p.200 Sample6.cでも同じ）
において、多次元配列testをグローバル変数として宣
言し、初期化の方法（要素1つずつの代入ではない）
を使ったプログラムに変更してください。
（初期化の記述はp.202を参照）

【課題10-2】

課題10-1で作成したプログラムに対して、多次元配列testに「英語」の科目を追加して下さい。

(配列の次元を増やすのではなく、**科目数SUBを増やす**ことに注意)

配列の初期化、点数の出力も英語に対応できるように変更してください。

国語→	test[0][0]	test[0][1]	test[0][2]	test[0][3]	test[0][4]
算数→	test[1][0]	test[1][1]	test[1][2]	test[1][3]	test[1][4]
英語→	test[2][0]	test[2][1]	test[2][2]	test[2][3]	test[2][4]

【課題10-2】

[実行結果（英語部分のみ、英語の得点を{54, 73, 65, 82, 98}とした場合）]

...

1番目の人の英語の点数は54です。

...

2番目の人の英語の点数は73です。

...

3番目の人の英語の点数は65です。

...

4番目の人の英語の点数は82です。

...

5番目の人の英語の点数は98です。

【課題10-3】

多次元配列testの指定した科目に対して最大値（つまり最高得点）を見つけ、その値を戻り値とする関数max_test()を作成してください。

[この関数のプロトタイプ宣言]

```
int max_test(int s);
```

```
/* 最大値を格納する変数を用意し、初期値をtest[s][0]とする */
```

```
/* サンプルプログラムの23行目以降を参考に繰り返し処理を作る */
```

```
/* 繰り返し処理の中で比較をして最大値を更新していく */
```

```
/* 繰り返し処理が終わったら最大値を戻す */
```

【課題10-3】

[配列testをグローバル変数として宣言する]

```
/* 課題10-2で作った配列を宣言しておく */
```

[mainでの処理]

```
printf("国語の最高得点は%dです。 \n", max_test(0));  
printf("算数の最高得点は%dです。 \n", max_test(1));  
printf("英語の最高得点は%dです。 \n", max_test(2));
```

[実行結果]

国語の最高得点は80です。

算数の最高得点は90です。

英語の最高得点は98です。

【課題10-4】

多次元配列testの指定した人に対して、最大値（つまり最高得点）の科目名を出力する関数max_sub()を作成してください。

[この関数のプロトタイプ宣言]

```
void max_sub(int n);
```

```
/* 引数nの人に対して最大値の科目名を見つけ出力する */
```

```
/* （値を戻すのではなく、関数内で結果を出力することに注意） */
```

【課題10-4】

[配列testをグローバル変数として宣言する]

```
/* 課題10-2で作った配列を宣言しておく */
```

[mainでの処理]

```
for(i=0; i<NUM; i++) {  
    max_sub(i);  
}
```

[実行結果]

- 1番目: 最高得点科目は算数で90点です。
- 2番目: 最高得点科目は英語で73点です。
- 3番目: 最高得点科目は算数で68点です。
- 4番目: 最高得点科目は英語で82点です。
- 5番目: 最高得点科目は英語で98点です。

まだ余裕のある人は…【課題10-5】

九九の計算表を生成するプログラムを作成してください。以下の補足を参考にしてください。

- ▶ main()内の処理だけで作ってよい
- ▶ 10×10の2次元配列を作り、0番目は使用しないように作ると、配列の添字がそのまま計算に使う
- ▶ for文の中に、さらにfor文（forのネスト構造）を使う必要がある（ネスト構造についてはp.164～を参照）
- ▶ 計算表の出力は全て2桁でそろえるようにprintf()の変換仕様を使うと、出力がそろってキレイになる（変換仕様についてはp.388～を参照）

【課題10-5】

[実行結果]

	1	2	3	4	5	6	7	8	9

1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

小テストについて

小テストの注意点

- 他人の力は借りずに、自分だけでプログラムを作成する。（つまり定期試験と同様）
- 小テスト中は、演習室外へのネットワークアクセスは遮断される。

小テストについて

小テスト中に参照できるもの

- 教科書, 配付資料
- 自分のホームディレクトリ（ホームフォルダ）以下に保存されているファイル
- 小テストでは紙媒体のものは参照可能
- 上記以外の情報を参照することは不正行為とする
例：USBで接続された機器に保存されているファイルの参照
ネットワークを介した情報の参照、など