

C 言語検定必勝プリント 明日のために 3 級編

No.14 配列で遊ぶ No.1

昨日のプログラムは完成しましたか？順番に少しずつやれば簡単なのですが一気に全体を仕上げようとするとうれないうちは大変だと思います。
今日は、昨日よりも C 言語に依存しない部分でなおかつよく検定に出題される項目のプログラミングを行ってみたい。とりあえず多次元配列を使ったプログラムを作ってみよう。

多次元配列・・・分らない。そうですね、前回の配列の説明では簡単にしか説明していなかったのが分りにくかったかもしれません。今回は文字列にターゲットを絞って多次元配列で遊んでみます。

難しい理屈は抜きにして、単純な文字列を使った次のソースを入力し実行してみてください。

```
#include <stdio.h>

void main(void){
    char    c[10];
    int     i;

    for( i=0 ; i<10 ;i++ )
        c[i]='A'+i;

    printf(" %s\n",c);
}
```

何をしようとしているか分かりますか？またその実行結果は分かりますか？じつはこれ、バグがあり毎回同じ結果になるとは限りません。たまたまうまく表示できたりしますが、文字列に独特な重大な欠陥があります。

まずループを抜けた時点での配列のイメージを下に書いて見ます。

c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]	c[8]	c[9]
'A'	'B'	'C'	'D'	'E'	'F'	'G'	'H'	'I'	'J'

どこがおかしいのでしょうか？0 から 9 全部で 10 個の箱に 1 文字ずつちゃんと入っています。

実のところ C 言語で使う文字列は、定義の際添え字の番号を決めて定義しているのにもかかわらず、どこまでが一塊かをきちんとしておかないと本当にどこまでが文字列なのかコンピュータ自身が分からなくなってしまうのです。

そこで文字列の最後を示すために上記のような場合文字列の最後には ‘\0’ を入れなければ駄目なのです。しかし全てに文字が入っているため、最後の printf でどこまで表示していいのか分からなくなってしまうのです。
どこまで表示するのでしょうか、答えは ‘\0’ が見つかるまでです。ちなみに最後の printf のひとつ手前に c[5]='\0'; と入れて実行してみてください。表示は ABCDE （5 個表示）となるはず。

面倒くさいですね。実はこんなところが C の苦手意識を倍化させる部分なのですが仕組みが分かれば結構簡単なのです。

またもうひとつのポイントとして配列の場合の特例ですが「c[5]」というふうに添え字がある場合はその添え字に対応する箱の中身を示しますが「c」という風に配列名だけだとその配列の先頭アドレス（配列が保存されている場所の先頭番地）の指定になることです。

普通の char c[5] こういった配列を 1 次元配列といいます。
配列のイメージは・・・

c[0]	c[1]	c[2]	c[3]	c[4]
------	------	------	------	------

こんな感じですね。
この配列を拡張して char c[2][5] とすると 2 次元配列といいます。ちなみにイメージを書いてみると

c[0][0]	c[0][1]	c[0][2]	c[0][3]	c[0][4]
c[1][0]	c[1][1]	c[1][2]	c[1][3]	c[1][4]

こんな感じになります。当然 3 次元配列、4 次元配列もつくれますよ。

※ちなみにこの考え方は非常に大切です。重要ポイント

では、配列をなぜ使うかを考えて見ます。ここでは3目並べのプログラムで考えて見ます。

まずステージを考えます、ステージ構成は以下のようになりますね。ここでは配列を使ってみます。

char st[3][3]⇒

	0 (A)	1 (B)	2 (C)
0	<div>[0][0] 0</div>	<div>[0][1] 0</div>	<div>[0][2] 0</div>
1	<div>[1][0] 0</div>	<div>[2][1] 0</div>	<div>[1][2] 0</div>
2	<div>[2][0] 0</div>	<div>[2][1] 0</div>	<div>[2][2] 0</div>

ここでは置いたかどうかだけ判断すればいいので char 型の変数を使い場所ごとの状態を表します。

これを変数で行うと大変です。試しにやってみましょう。

ここで、位置の入力をする際の座標ですが横を A～C、縦を 0～2 ですることとします。そして交互に打ち込むとします。

3つ並んだかどうかの判断はどうしましょうか？要するに縦横3つ並べばいいのですから、対戦相手が A さんと B さんだった場合、A さんが置いた所の値を 1 に、B さんが置いた所を -1 とすると、3つの合計が 3 になれば A さんの勝ち、-3 になれば B さんの勝ちと判断できそうです。

入力は、とりあえず 1 文字入力できればいいので getchar() を使います。その他の仕様を下に決めておきますね。

3 目並べ仕様書

1. 入力後、間違っていないか確認をし、入力がおかしければ再入力をする。
2. A さん B さん交互に打ち込む。
3. 先行は「o(オー)」、後攻は「x(エックス)」を表示する

とりあえずこれだけです、さあ組んでみましょう。といっても大変だと思うので初めにステージを画面に表示する部分まで書いておきます。後は自分で考えてみてください。

C 言語検定必勝プリント解答編 明日のために 3 級編

No.14 配列で遊ぼう

とりあえずステージ表示までを書いておきます。後は頑張ってね。

```
#include <stdio.h>

void main(void){
    int i,j;

    //ステージデータ定義及び初期化
    char st[3][3]={ 0,0,0,
                    0,0,0,
                    0,0,0 };
    printf("<<<3 目並べゲーム>>>\n");

    for(
```