



2007年10月19日

## 第8回 C言語によるプログラミング (5)

# 情報処理論 (応用)

松山大学 経営学部

檀 裕也

<http://www.cc.matsuyama-u.ac.jp/~dan/education/application/>



# 前回の実習課題

- 1から100までの整数のうち、入力値の倍数のみ表示するプログラムを作成し、ソースコード `multiple.c` を添付してメールを送信せよ。
  - 宛先: `dan@cc.matsuyama-u.ac.jp`
  - 件名: プログラミング課題#06

```
C:¥>multiple.exe
整数 : 2
2 4 6 8 10 12 14 16 18 20 22 ...

C:¥>
```



# 解答例

multiple.c

```
#include <stdio.h>

int main()
{
    int i, x;

    printf( "整数：" );
    scanf( "%d", &x );

    for( i = x; i <= 100; i += x ){
        printf( "%d ", i );
    }

    printf( "¥n" );

    return 0;
}
```



# 別解

```
#include <stdio.h>
```

```
multiple.c
```

```
int main()
```

```
{
```

```
    int i, x;
```

```
    printf( "整数：" );
```

```
    scanf( "%d", &x );
```

```
    for( i = 1; x * i <= 100; i++ ){
```

```
        printf( "%d ", x * i );
```

```
    }
```

```
    printf( "¥n" );
```

```
    return 0;
```

```
}
```



# 補足

- 入力値に制限をつけることで、無限ループを回避する。

```
do{  
    printf( "整数：" );  
    scanf( "%d", &x );  
}while( x > 0 );
```

- 変換仕様に桁数を設定できる。

```
printf( "%3d ", i );
```



# 今回の授業内容

---

- C言語によるプログラミング (5)
  - for 文による繰り返しの処理



# 例題 1

**triangle.c**

```
#include <stdio.h>

int main()
{
    int i, j;

    for( i = 1; i <= 10; i++ ){
        for( j = 0; j < i; j++ ){
            printf( "*" );
        }
        printf( "\n" );
    }

    return 0;
}
```



# 実行例

- 三角形の図が表示される。

```
C:¥>triangle.exe
```

```
*
```

```
**
```

```
***
```

```
****
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
C:¥>
```





# 解説 ( 1 )

- for 文の定型構文
  - 次のように記述すると、ブロック内の処理を N 回繰り返すことになる

```
for( i = 0; i < N; i++ ){  
    printf( "*" );  
}
```



# 解説 ( 2 )

- for 文のネスト構造
  - for 文の中に for 文を使うことができる

```
for( i = 1; i <= 10; i++ ){  
    for( j = 0; j < i; j++ ){  
        printf( "*" );  
    }  
    printf( "%n" );  
}
```

```
C:¥>triangle.exe
```

```
*           i=1 の処理  
**          i=2 の処理  
***         i=3 の処理  
****        i=4 の処理  
*****       i=5 の処理  
*****       i=6 の処理  
*****       i=7 の処理  
*****       i=8 の処理  
*****       i=9 の処理  
*****       i=10の処理
```

```
C:¥>
```



# 例題 2

```
#include <stdio.h>
```

```
tree.c
```

```
int main()
```

```
{
```

```
    int i, j;
```

```
    for( i = 1; i <= 10; i++ ){
```

```
        if( i <= 7 ){
```

```
            for( j = 0; j < i; j++ ){
```

```
                printf( "*" );
```

```
            }
```

```
        }
```

```
    else{
```

```
        printf( "*" );
```

```
    }
```

```
    printf( "%n" );  
}
```

```
    return 0;
```

```
}
```



# 実行例

- 樹木の図が表示される。

```
C:¥>tree.exe
```

```
*
```

```
**
```

```
***
```

```
****
```

```
*****
```

```
*****
```

```
*****
```

```
*
```

```
*
```

```
*
```

```
C:¥>
```



# 解説

- 7行までと8行目以降で表示内容を変更する

```
if( i <= 7 ){
    for( j = 0; j < i; j++ ){
        printf( "*" );
    }
}
else{
    printf( "*" );
}
```

```
C:¥>tree.exe
*           i=1
**          i=2
***         i=3
****        i=4
*****      i=5
*****      i=6
*****      i=7
*           i=8
*           i=9
*           i=10

C:¥>
```



# 例題 3

```
#include <stdio.h>

int main()
{
    int i, j;

    for( i = 0; i < 8; i++ ){
        for( j = 0; j < 8; j++ ){
            if( ( i + j ) % 2 == 0 ){
                printf( "  " );
            }
            else{
                printf( "  " );
            }
        }
    }
}
```

```
printf( "%n" );
}

return 0;
}
```

chessboard.c



# 実行例

- 市松模様が表示される

```
C:¥>chessboard.exe
```

```
C:¥>
```



# 解説

j

i

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	8
2	2	3	4	5	6	7	8	9
3	3	4	5	6	7	8	9	10
4	4	5	6	7	8	9	10	11
5	5	6	7	8	9	10	11	12
6	6	7	8	9	10	11	12	13
7	7	8	9	10	11	12	13	14





# 実習課題

- 例題 2 を参考にして、右図の樹木を表示するプログラムを作成し、ソースコード `tree2.c` を添付してメールを送信せよ。
  - 宛先：  
dan@cc.matsuyama-u.ac.jp
  - 件名：プログラミング課題 #07

```
C:¥>tree2.exe

      **
     ****
    *****
   ********
  *********
 *****
*****
*****

      **
     **
    **

C:¥>
```



# 次回の予定

---

- 日時： 10月24日（水）  
5時限
- C言語によるプログラミング（6）  
– 繰り返しの処理