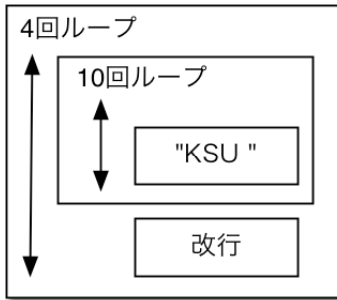


## ■ 二重ループ

for() 文を二重に重ねることもできます。右のプログラムを実行して結果を確認してください。

10 回 KSU を出力し、その後改行を出力、これを 4 回繰り返します。



```
#include <stdio.h>

/*
   for による二重ループ 473088 榎田裕一郎
*/

int main() {
    int i, j;

    for(i=0; i<4; i++) {
        for(j=0; j<10; j++) {
            printf("KSU ");
        };
        printf("\n");
    };
    return 0;
}
```

## ● 課題 1.

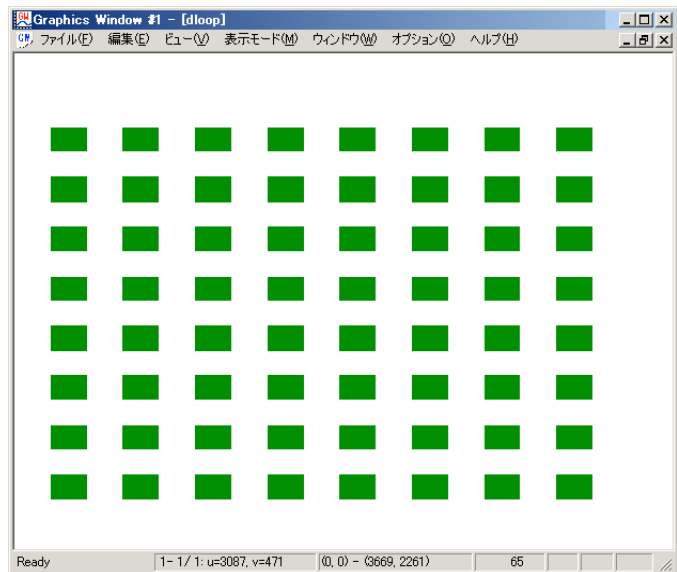
二重ループを使って、先週作成した横棒を引くプログラムを修正し、以下のような絵柄を描くプログラムを作成して下さい。

考え方のヒント：

- ・ サンプルでは長方形のサイズを 10 x 10 にしました。
- ・ それを 8 列 8 行に並べています。
- ・ 小さな四角を 1 つ描く記述があれば、
- ・ そこを 8 回ループさせれば一行ぶん(小さな四角を 8 つ描く)ができます。
- ・ そのループをさらに 8 回ループさせれば、8 行分(合計 64 個の小さな四角)が描けます。

自信のある人へ：

- ・ 小さな四角を描くたびに色を変えてみて下さい。



更に時間のある人へ：

今作ったものとは異なる座標位置の算出方法を試みて下さい。

- ・ x, y 座標そのものを for 文の変数に出来ないだろうか？
- ・ x, y 座標をループを回る毎に足すのではなく、毎回計算して出せないだろうか？

## ■ for 以外の方法によるループ

### □ while

右のプログラムを入力して実行してください。0 から 9 までの数字を出力するは  
ずです。

while 文はカッコの中の条件式が真である  
限り、それに続くブロックの処理を繰り返  
し続けます。

書式：

```
while( 繰り返し条件式 ) {  
    繰り返す処理  
};
```

### □ for と while の比較

for 文で同様のプログラムを作りましたが、while を用いてこ  
のように書くことも出来ます。for における初期処理、繰り返  
し条件、繰り返し毎処理が while 文でどのように配置されて  
いるかに注意して下さい。

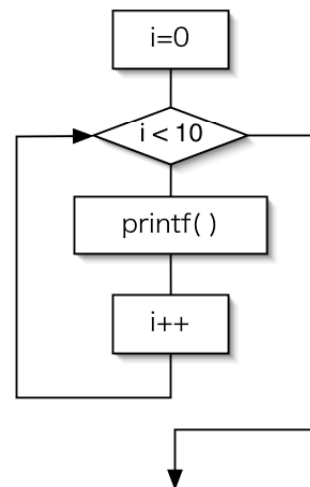
```
#include <stdio.h>  
  
/*  
    while によるループ 473088 榎田裕一郎  
*/  
  
int main()  
{  
    int i=0;  
  
    while (i < 10) {  
        printf("%3d¥n", i);  
        i++;  
    };  
    return 0;  
}
```

注意：int i=0; というの  
は宣言と同時に 0 を代入す  
るというもので、  
int i;  
i=0;  
と書いた場合と等価です。

The diagram compares the syntax of a while loop and a for loop. In the while loop, 'i=0;' is boxed and has an arrow pointing to the start of the loop. 'i<10' is boxed and has an arrow pointing to the loop condition. 'i++;' is boxed and has an arrow pointing to the loop body. In the for loop, 'i=0' is boxed and has an arrow pointing to the initialization part of the loop. 'i<10' is boxed and has an arrow pointing to the condition part. 'i++' is boxed and has an arrow pointing to the increment part. Red arrows also show the flow from the initialization to the condition, and from the condition to the body, and from the body back to the condition.

右の図は for による処理の流れを説明したときに用いたもので  
すが、今回の while による記述もまったくこの流れの通りに行われ  
ています。つまり例にあげた while と for によるループ処理は全  
く等価なものです。

あるループを for で書くか、while で書くかはプログラマの判断  
に任されています。より読みやすく、わかりやすい方法で記述す  
るように書き方を選ぶようにして下さい。(この例のような単純  
な処理ではどちらを選んでも大差ありませんが、より複雑な処理  
になるにつれ、こうした選択が重要になってきます。)



● 課題 2.

今週作成した二重ループを利用したグラフィクスプログラムか、先週作成した長方形を四つ描くプログラムか、どちらかを `while` を使ったループに書き換えて下さい。

補足資料：

□ `do~while` 文による表現

`do~while` 文によるループの表現方法もあります。

右のプログラムを実行して動作を確認してください。

`do` 文は、それに続くブロックの処理をまず行い、その後に `while ( )` に示された条件文による判定を行います。

書式：

```
do {  
    繰り返す処理  
} while( 繰り返し条件式 );
```

```
#include <stdio.h>  
  
/*  
    do while によるループ 473088 榎田裕一郎  
*/  
  
int main()  
{  
    int i=0;  
  
    do {  
        printf("%3d\n", i);  
        i++;  
    } while (i < 10);  
    return 0;  
}
```

□ `do~while` 文と `while` 文の比較

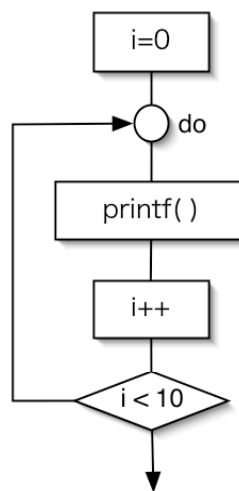
両者の処理の流れは右図のようになります。条件判定の位置が異なることに注目して下さい。

注意点：

`while ( )`文では、括弧内の条件判定が合わないと、繰り返し処理部分を一度も実行しないことがある。

`do~while ( )`文では、最低 1 回は繰り返し処理部分を実行する。

do ~ while によるループ



while によるループ

