

## 課題 510

// (1) 最初の行に必ず書く呪文

```
int main () {  
    // (2) 整数変数の配列 mogura を宣言
```

```
    // (3) 配列に値を代入: いる,いない,いる,いない
```

```
    // (4) 出力: 変数の値をすべて出力してください
```

```
    return(0);  
}
```

## 課題 520

// (1) 最初の行に必ず書く呪文

```
int main () {  
    // (2) 整数変数の配列 mogura の宣言と代入  
    //      モグラは「いる」いない」いない」いる」  
  
    // (3) 出力: 変数の値をすべて出力してください  
  
    return(0);  
}
```

## 課題 530

// (1) 最初の行に必ず書く呪文

```
int main () {  
    // (2) 整数変数の配列 mogura の宣言と代入  
    //      モグラは「いる」いない」いない」いる」  
  
    // (3) 代入: jibun に 1 を代入  
  
    // (4) 条件文と出力: jibun の場所にモグラがいる?  
    //      いれば atari、いなければ hazure と出力  
  
    return(0);  
}
```

## 課題 540

```
// (1) 最初の行に必ず書く呪文
```

```
int main () {  
    // (2) 整数変数の配列 mogura の宣言と代入  
    //     モグラは「いる」いない」いない」いる」
```

```
    // (3) 入力: キーボードから jibun に値を読み込む
```

```
    // (4) 条件文と出力  
    //     jibun の場所にモグラがいるか?  
    //     いれば atari、いなければ hazure と出力
```

```
    return(0);  
}
```

## 課題 550

```
// (1) 最初の行に必ず書く呪文
```

```
int main () {  
    // (A) 一回だけ実行する部分  
  
    for ... 省略 ... { // 繰り返し文  
        // (B) くりかえし実行する部分  
  
    }  
  
    return(0);  
}
```

## 課題 560

```
// (1) 最初の行に必ず書く呪文
```

```
int main () {  
    // (2) 整数変数の配列 mogura の宣言と代入  
    //  
    //     つまり、以前ベタ書きした次の処理を  
    //     くりかえし文+scanf で書き直してください  
    //     位置 0 に「いる」つまり mogura[0]=1;  
    //     位置 1 に「いない」  
    //     位置 2 に「いない」  
    //     位置 3 に「いる」  
  
    // (3) 入力:          (540 と同じ, 省略)  
    // (4) 条件文と出力 (540 と同じ, 省略)  
  
    return(0);  
}
```

## 課題 610

```
// (1) 最初の行に必ず書く呪文
```

```
int main () {  
    // (2) 整数変数の配列 mogura の宣言
```

```
    // (3) モグラの位置を乱数で決める
```

```
    // (B) 繰り返し  
    // (4) 入力: キーボードから jibun に値を代入  
    // (5) 条件文と出力  
    //      jibun の場所にモグラがいるか?
```

```
    return(0);  
}
```

## 課題 620

```
// (1) 最初の行に必ず書く呪文
```

```
int main () {  
    // (2) 整数の変数 tensuu と配列 mogura を宣言  
    // (3) モグラの位置を乱数で決める  
    // (B) 繰り返し  
        // (4) 入力: キーボードから jibun に値を代入  
        // (5) 条件文と出力:  
            //      jibun の場所にモグラがいるか?  
            //      atari の時は tensuu に加算  
    // (6) tensuu を表示  
  
    return(0);  
}
```

## 課題 630

```
// (1) 最初の行に必ず書く呪文  
//     定数
```

... 以下、課題 620 と同様 ...

## 課題 690

```
// (1) ヘッダファイル  
//     定数
```

```
int main () {  
    // (A) 一度だけ処理する部分  
    //     (2) 宣言  
    //     (3) モグラの位置決め(乱数)
```

```
    // (B) 繰り返し  
        // (4) 入力  
        // (5) 条件文と出力
```

```
    // (6) tensuu を表示
```

```
    return(0);  
}
```

## 課題 690

```
// (1) ヘッダファイル  
//     定数
```

```
int main () {  
    // (A) 一度だけ処理する部分  
    //     (2) 宣言  
    //     (3) モグラの位置決め(乱数)
```

```
    // (B) 繰り返し  
        // (4) 入力  
        // (5) 条件文と出力
```

```
    // (6) tensuu を表示
```

```
    return(0);  
}
```

## 課題 690

```
// (1) ヘッダファイル  
//     定数
```

```
int main () {  
    // (A) 一度だけ処理する部分  
    //     (2) 宣言  
    //     (3) モグラの位置決め(乱数)
```

```
    // (B) 繰り返し  
        // (4) 入力  
        // (5) 条件文と出力
```

```
    // (6) tensuu を表示
```

```
    return(0);  
}
```

## 課題 690

```
// (1) ヘッダファイル  
//     定数
```

```
int main () {  
    // (A) 一度だけ処理する部分  
    //     (2) 宣言  
    //     (3) モグラの位置決め(乱数)
```

```
    // (B) 繰り返し  
        // (4) 入力  
        // (5) 条件文と出力
```

```
    // (6) tensuu を表示
```

```
    return(0);  
}
```

## 課題 690

```
// (1) ヘッダファイル  
//     定数
```

```
int main () {  
    // (A) 一度だけ処理する部分  
    //     (2) 宣言  
    //     (3) モグラの位置決め(乱数)
```

```
    // (B) 繰り返し  
        // (4) 入力  
        // (5) 条件文と出力
```

```
    // (6) tensuu を表示
```

```
    return(0);  
}
```

## 課題 695

```
// (1) ヘッダファイル  
//     定数
```

```
int main () {  
    // (A) 一度だけ処理する部分  
    //     (2) 宣言  
    //     (3) モグラの位置決め(乱数)  
  
    // (B) 繰り返し  
    //     (4) 入力  
    //     (5) 条件文と出力  
    //     (5A) 記録 (進行状況表示のため)  
  
    // (C) 進行状況を表示  
  
    return(0);  
}
```

## 課題 710

[実行例]

```
$ ./mogura  
1  
2  
atari  
2  
3  
hazure  
6  
8  
atari  
tensuu = 2
```

## 課題 720

[実行例 (9x9)]

6

7

atari!

```
. X . . . . . . . .  
. X . . . . . X . .  
. . X X X . . . . .  
. . . . . . . . . .  
. . . X . X . X . .  
. . . . . . . . . .  
. . . . . . . 0 . .  
. . . . . . . . . .  
. . . . . . . . . .
```

## 課題 730

[実行例 (9x9)]

7

8

atari!

```
. X . . . . . . . .  
. X . . . . . X . .  
. . X X X . . . . .  
. . . . . . . . . .  
. . . X . X . X . .  
. . . . . . . . . .  
. . . . . . . 0 . .  
. . . . . . . . . .  
. . . . . . . . . .
```

## 課題 740

[実行例 (9x9)]

7

8

atari!

```
  1 2 3 4 5 6 7 8 9
1 . X . . . . . . 1
2 . X . . . . . X 2
3 . . X X X . . . 3
4 . . . . . . . . 4
5 . . . X . X . X 5
6 . . . . . . . . 6
7 . . . . . . . 0 7
8 . . . . . . . . 8
9 . . . . . . . . 9
  1 2 3 4 5 6 7 8 9
```

- 課題 730 を改造して、モグラたたきの様子が見やすいように 1 ... 9 の枠をつけてください
  - これも配列の境界まわりのあつかいで頭をひねる問題:-)

## 課題 810

```
// (1) ヘッダファイル  
//     定数
```

```
int main () {  
    // (A) 一度だけ処理する部分  
    //     (2) 宣言  
    //     (3) モグラの位置決め(乱数)
```

```
    // (B) 繰り返し  
        // (4) 入力  
        // (5) 条件文と出力
```

```
    // (6) tensuu を表示
```

```
    return(0);  
}
```

## 課題 820

```
// (1) ヘッダファイル  
//     定数
```

```
int main () {  
    // (A) 一度だけ処理する部分  
    //     (2) 宣言  
    //     (3) モグラの位置決め(乱数)
```

```
    // (B) 繰り返し  
        // (4) 入力  
        // (5) 条件文と出力
```

```
    // (6) tensuu を表示
```

```
    return(0);  
}
```

## 課題 830

```
// (1) ヘッダファイル  
//     定数
```

```
int main () {  
    // (A) 一度だけ処理する部分  
    //     (2) 宣言  
    //     (3) モグラの位置決め(乱数)
```

```
    // (B) 繰り返し  
        // (4) 入力  
        // (5) 条件文と出力
```

```
    // (6) tensuu を表示
```

```
    return(0);  
}
```