

課題の提出について

- 課題はすべて**関数として**提出してください.
- まず、関数の形を作ります.

問題例：整数 n が与えられたとき、 n の2乗と n の3乗を足した値を返す関数 `kadai(n)` を作れ.

解答

```
function kadai(n){  
    var k = n * n * n + n * n  
    return k  
}
```

←関数を定義する

←値を返す

このようにすると、`kadai(23)`などと入力することにより、結果を計算することができる.

`puts(kadai(23))` ---> 答の12696が出力される.

`puts(kadai(33))` ---> 答の37026が出力される.

これを提出する！

putsなど、これ以外の物は含めない

課題の提出方法

アルゴリズム・データ構造1のホームページ → 課題提出ページ

The screenshot shows a web browser window with the URL `osami.s280.xrea.com/Algo_Data2015/submit.html`. The page has a sidebar on the left with a list of lessons from '第2回課題' to '第14回課題'. The main content area is titled '第2回課題' and contains a submission form. The form includes fields for '学籍番号' (001122334) and 'パスワード' (masked with dots). Below these are buttons for '提出' (Submit), '内容チェック' (Check Content), 'ダウンロード' (Download), and 'ようこそ' (Welcome). A text area contains the message '文法構造は大丈夫そうです。' (Syntax structure seems to be fine). Below this is a section for 'プログラム' (Program) with a timestamp 'Wed Apr 1 16:59:54 2015'. The program code is a JavaScript function for calculating Fibonacci numbers. Below the code is a section for '説明' (Explanation) with a text area containing a detailed description of the Fibonacci sequence calculation. At the bottom, there is a section for 'EAによるコメント' (Comments by EA) with an empty text area. Blue arrows point from the text on the right to various elements on the page: from the first bullet point to the lesson list, from the second to the password field, from the third to the '内容チェック' button, from the fourth to the '学籍番号' field, from the fifth to the '提出' button, and from the sixth to the 'ダウンロード' button.

学籍番号: 001122334 パスワード:

第2回課題 提出 内容チェック ダウンロード ようこそ 山本修身 さん

文法構造は大丈夫そうです。

プログラム (Wed Apr 1 16:59:54 2015)

```
function kadai1(){
  function fib(n){
    if (n <= 1) return 1;
    else return fib(n - 1) + fib(n - 2);
  };
  for (var i = 0; i < 30; i++){
    puts(i + " " + fib(i));
  }
}
```

説明

フィボナッチ数列の計算を効率的に行うプログラム。フィボナッチ数列は並んだ2つの数列の値の和によってその直後の数列の値が定義される。この数列の一般解を求めることはそれほど難しくなりが、再帰的なプログラムとして実現することはそれほど難しくない。

EAによるコメント

- まずは、課題番号をクリックする
- そのあと、プログラムとコメントを入力する。
- まず、ボタン「内容のチェック」を押して文法に問題がないことを確認。
- 学籍番号とパスワードを入力する。
- ボタン「提出」を押してアップロードする。
- ボタン「ダウンロード」を押して内容がアップロードされたことを確認。

複数のプログラム（関数）をまとめてプログラム欄に入力すること。
説明もまとめて説明欄に書くこと

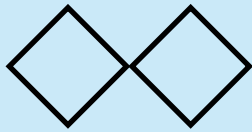
本日の課題 1

0からnまでの3つの整数の組 (i_1, i_2, i_3) のうち i_3 が i_1 と i_2 の中間値 $(i_1 + i_2) / 2$ に一致するような組の個数を返す関数 `kada11(n)` を作れ. 小数を結果とする演算の結果は誤差を含むことに注意してプログラムを書くこと.

今日の課題 2

1から n までの整数のうち $m1$ で割り切れて、 $m2$ では割り切れないような数の総数を返す関数 $kadai2(n, m1, m2)$ を書け。ただし、 $m1 \neq m2$ とする。

課題の提出は $kadai1(n)$, $kadai2(n, m1, m2)$ を並べて一つの枠に入れてアップロードすること。それ以外のputsなどの文は入れないこと。



解答例 1

問題の条件をそのままプログラムとして実装すれば良い.

```
function kadai1(n){  
    var i1, i2, i3;  
    var count = 0;  
    for (i1 = 0; i1 <= n; i1++){  
        for (i2 = 0; i2 <= n; i2++){  
            for (i3 = 0; i3 <= n; i3++){  
                if (i1 + i2 == 2 * i3)  
                    count += 1;  
            } /* for */  
        } /* for */  
    } /* for */  
    return count;  
} /* kadai1 */
```

```
for (i = 1; i < 20; i++){  
    puts(i + ": " + kadai1(i));  
} /* for */
```



```
1: 2  
2: 5  
3: 8  
4: 13  
5: 18  
6: 25  
7: 32  
8: 41  
9: 50  
10: 61  
11: 72  
12: 85  
13: 98  
14: 113  
15: 128  
16: 145  
17: 162  
18: 181  
19: 200
```

解答例 2

こちら問題に指定されている条件をそのまま実装すれば良い。

```
function kadai2(n, m1, m2){  
  var i;  
  var count = 0;  
  for (i = 1; i <= n; i++){  
    if (Math.floor(i / m1) * m1 == i &&  
        Math.floor(i / m2) * m2 != i)  
      count += 1;  
  }  
  return count;  
} /* kadai2 */
```

$$\left\lfloor \frac{1000}{5} \right\rfloor - \left\lfloor \frac{1000}{5 \times 7} \right\rfloor = 200 - 28 = 172$$

```
puts(kadai2(1000, 5, 7));  
puts(kadai2(1000, 3, 7));
```

→
172
286

$$\left\lfloor \frac{1000}{3} \right\rfloor - \left\lfloor \frac{1000}{3 \times 7} \right\rfloor = 333 - 47 = 286$$