

前回課題の解説

(第8回: 自作関数)

担当: 佐藤

2018年11月19日(月)

課題8-1

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

(この解答例ではa, b, cと同じにしているが)仮引数と実引数の名前をそろえる必要はない

```
void printAvg(int a, int b, int c) {  
    printf("%.1f¥n", (a + b + c) / 3.0);  
}
```

(double)(a + b + c) / 3;でもよい

```
int main() {  
    int a, b, c;
```

```
    printf("1つ目の整数を入力してください¥n");  
    scanf("%d", &a);  
    printf("2つ目の整数を入力してください¥n");  
    scanf("%d", &b);  
    printf("3つ目の整数を入力してください¥n");  
    scanf("%d", &c);  
    printAvg(a, b, c);
```

実引数の前にはデータ型を書かない

```
    return 0;
```

```
}
```

課題8-2

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

double rss(int h, int w) {
    return sqrt(h * h + w * w);
}

int main() {
    int h, w;

    printf("長方形の高さを入力してください\n");
    scanf("%d", &h);
    printf("長方形の幅を入力してください\n");
    scanf("%d", &w);
    printf("対角線の長さは%.1fです\n", rss(h, w));

    return 0;
}
```

sqrt()を使用するために必要(※)

※正確には、math.hをインクルードしなくてもsqrt()は使える。しかし、そうした場合、sqrt()の戻り値の型はintとみなされ、実行結果に小数点以下の値は反映されないであろう。詳しくは、[関数宣言](#)およびその一種である[関数プロトタイプ\(関数原型\)](#)について各自で調べてみよう!

課題8-3(1)

```
#include <stdio.h>

int gcd(int a, int b) {
    int c;

    while (b != 0) {
        c = a % b;
        a = b;
        b = c;
    }

    return a;
}

int main() {
    int a, b;

    printf("1つ目の整数を入力してください\n");
    scanf("%d", &a);
    printf("2つ目の整数を入力してください\n");
    scanf("%d", &b);
    printf("%dと%dの最大公約数は", a, b);
    printf("%dです\n", gcd(a, b));
    printf("%dと%dの最小公倍数は", a, b);
    printf("%dです\n", a * b / gcd(a, b));

    return 0;
}
```

別解

```
for (;;) {
    if (b == 0) {
        break;
    }
    c = a % b;
    a = b;
    b = c;
}
```

課題8-3(2)

● gcd()は2つの引数の間の大小関係に依存しない

▪ a == 24, b == 32 (a < b)

① b != 0

• c = 24;
• a = 32;
• b = 24;

② b != 0

• c = 8;
• a = 24;
• b = 8;

③ b != 0

• c = 0;
• a = 8;
• b = 0;

④ b == 0

• return a;

a < bの場合、初回の
ループでaとbの値
の入れ替えが行われ
る。その分、a > bの
場合に比べて1回だ
け余計にループ

▪ a == 32, b == 24 (a > b)

① b != 0

• c = 8;
• a = 24;
• b = 8;

② b != 0

• c = 0;
• a = 8;
• b = 0;

③ b == 0

• return a;

単にアルゴリズムの流れを把握するだけで
なく、アルゴリズムの性質について理解が
深まるため、知らないアルゴリズムに出
会ったら**具体的な値でトレース**してみよう!