

2022 年度  
プログラミング第 1 同演習

第 5 回

慶應義塾大学 理工学部 情報工学科

講義担当：河野健二

演習担当：杉浦裕太

## 【課題の提出方法】

提出時期：

- 次の授業日の午前 9:00
- 遅れた場合は減点されます
- 遅れる場合でも、さらにその次の授業日の 9:00 までには提出をしてください（提出フォームが自動的に閉じます）

提出先：

- CANVAS 経由

提出方法：

- 「xxxxxx.c」 ファイルを一つずつ CANVAS にアップロードして提出
- 課題を再提出したい場合は、変更した課題だけではなくすべての課題を再アップロードするようにしてください。（以前にアップロードしたものは消えてしまう可能性があります）

ファイル名：

- 課題のスライドのタイトルと同じ（採点時に重要なので必ず同じにしてください）。ただし課題を再アップロードする際はファイル名が変更されることもあり、これに関してはそのままで大丈夫です

終了対応：

- 課題が終わったら各自退出していただいてかまいません
- 授業終了の時間がきて課題が終わっていなくても帰宅していただいてかまいません
- 残りの課題は ITC パソコンか各自のパソコンで行ってください

ITC Linux パソコン以外で作業する際の注意点：

- 自分の環境で発生した問題は自己責任でお願いします
- 提出するプログラムコードは Linux 環境でコンパイルできることを確認すること
- 採点時に、Linux 環境でコンパイルできることを確認するため、その時点でコンパイルが失敗してしまった提出物の点数は自動的に 0 点になってしまいます

## 【必須課題】

### 05-a-01.c

#### 【課題】

x 座標と y 座標を `float` 型のメンバとして保持する構造体 `point` を定義して用い、二点の座標を入力するとその二点を通る一次関数の式を出力するプログラムを作成してください。

#### 【ヒント】

同一の点が入力された場合を考慮しなくて良い。切片がマイナスだったときに、+表記が残ることは気にしない。

#### 【実行例：入力は赤字】

```
input first point (x1 y1) : 3.2 2.4
input first point (x2 y2) : 1.2 3.6
linear function : y = -0.600000 x + 4.320000
---
input first point (x1 y1) : 2 4
input first point (x2 y2) : 5 1
linear function : y = -1.000000 x + 6.000000
```

#### 【解答例】

## 05-a-02.c

### 【課題】

年(int)、月(int)、日(int)をメンバに持つ構造体 `struct mydate` を用意し、入力した年、月、日を `yyyy/mm/dd` 形式で表示する関数 `void printdate(struct mydate)` を定義し使用せよ。

### 【ヒント】

桁数が少ない場合の表示には、`%02d`、`%04d` を利用する。

### 【実行例：入力は赤字】

```
1999 7 3
1999/07/03
---
2016 6 21
2016/06/21
```

### 【解答例】

## 05-a-03.c

### 【課題】

char, short int, int, long int, float, double 型の変数のバイト数と、前述の問題(a-01)で定義した構造体 struct point 型の変数のバイト数をそれぞれ表示するプログラムを作成してください。(コンピュータによっては実行例と値が違うこともある)

### 【実行例：入力赤字】

char	1(byte)
short int	2(byte)
int	4(byte)
long int	4(byte)
float	4(byte)
double	8(byte)
struct point	8(byte)

### 【ヒント】

sizeof を使う。

### 【解答例】

## 05-a-04.c

### 【課題】

以下のプログラムの穴を埋め、実行例のような出力を得るようにしてください。

※include は省略

```
struct point{
    int a;
    int b;
    int c;
};
int main()
{
    struct point* p;
    //p の領域を動的に確保する

    //数字 3 つを入力にとる。

    printf("%d + %d + %d  = %d\n", p->a, p->b, p->c, p->a + p->b + p->c);

    //確保した領域を解放

    return 0;
}
```

### 【実行例：入力赤字】

Input three nums : **1 2 3**

1 + 2 + 3 = 6

### 【解答例】

## 05-a-05.c

### 【課題】

2つの実数を入力とし、その和差積商を出力するプログラムを作成してください  
この時、和(float)・差(float)・積(float)・商(float)をメンバにもつ構造体 `calc_result` のポインタを戻り値とする関数 `struct calc_result* get_calc_result(float,float)` を作成し、それを用いること。

### 【ヒント】

0で除算するときは商を0にすること。  
`get_calc_result` 内では `malloc` を使用して動的メモリを確保したポインタを返すこと。

### 【実行例：入力赤字】

```
input two integers :23 34
23.000000 + 34.000000 = 57.000000
23.000000 - 34.000000 = -11.000000
23.000000 × 34.000000 = 782.000000
23.000000 ÷ 34.000000 = 0.676471
---
input two integers :100 0
100.000000 + 0.000000 = 100.000000
100.000000 - 0.000000 = 100.000000
100.000000 × 0.000000 = 0.000000
100.000000 ÷ 0.000000 = 0.000000
```

### 【解答例】

## 05-a-06.c

### 【課題】

学籍番号と数学・英語・理科の点数をそれぞれ `int` 型のメンバとして保持する構造体 `score` を定義し、学籍番号および合計点と平均点を計算し表示する関数 `void print_total_average(struct score *)` を作成して動作を確認するプログラムを作成してください。

### 【ヒント】

実行例では `main` 関数は以下のように記述している。※`STUDENT_NUM` は 3

```
int main()
{
    int i;
    struct score list[STUDENT_NUM] = {
        { 61500001, 90, 72, 55 },
        { 61500002, 82, 77, 85 },
        { 61500003, 65, 91, 73 },
    };

    for (i = 0; i < STUDENT_NUM; i++) {
        print_total_average(&(list[i]));
    }

    return 0;
}
```

### 【実行例：入力は赤字】

```
student[61500001] total : 217, average : 72.333336
student[61500002] total : 244, average : 81.333336
student[61500003] total : 229, average : 76.333336
```

### 【解答例】



## 【標準課題】

### 05-b-01.c

#### 【課題】

学籍番号(int)・身長(int)・体重(int)をメンバとして保持する構造体 `struct student_data` を定義し、学生の情報を入力するとともに、身長、体重の最高値・最低値を学籍番号とともに表示するプログラムを作成してください。

生徒の最大数は 10 とし、学籍番号として負の数が入力されたら入力を終了する

#### 【ヒント】

for 文を途中で抜ける場合は `break` 文を使用する。

#### 【実行例：入力は赤字】

Input student number, height, weight : **61401234 150 100**

Input student number, height, weight : **61405678 200 50**

Input student number, height, weight : **61408765 399 10**

Input student number, height, weight : **61401111 29 10000**

Input student number, height, weight : **-1 -1 -1**

Tallest student number is 61408765 (399cm)

Shortest student number is 61401111 (29cm)

Heaviest student number is 61401111 (10000kg)

Lightest student number is 61408765 (10kg)

#### 【解答例】

## 05-b-02.c

### 【課題】

前述の問題(a-02)で定義した構造体 `struct mydate`、関数 `printdate` に加え、日付を 1 つ進める関数 `struct date tomorrow(struct date)` を定義して使用せよ。  
うるう年、大小の月を考慮せよ。

### 【ヒント】

大小の月の判断は `switch` 文を使用すると楽に書ける（使用しなくても今まで習った書き方でできる）。

### 【実行例：入力は赤字】

```
2016 6 28
Today is 2016/06/28
Tomorrow is 2016/06/29
---
2020 2 28
Today is 2020/02/28
Tomorrow is 2020/02/29
---
2019 2 28
Today is 2019/02/28
Tomorrow is 2019/03/01
---
2016 5 31
Today is 2016/05/31
Tomorrow is 2016/06/01
```

### 【解答例】

## 05-b-03.c

### 【課題】

前述の問題(a-01)で定義した構造体 `struct point` と半径(float)をメンバとして持つ構造体 `struct circle` を定義し、2 つの円が交わる点の数を返す関数 `int circle_intersect(struct circle, struct circle)` を定義して使用せよ。

### 【ヒント】

2 つの円の中心間の距離を  $d$  とする。2 円が 2 点で交わる条件は  $R-r < d < R+r \Leftrightarrow (R-r)^2 < d^2 < (R+r)^2$  ただし  $R \geq r$

1 点、0 点で交わる条件はこのヒントでは省略している。

### 【実行例：入力は赤字】

```
Input 1st Circle(x, y, r):0 0 3
Input 2nd Circle(x, y, r):5 0 2
Circles have 1 intersect point
---
Input 1st Circle(x, y, r):0 1 4
Input 2nd Circle(x, y, r):4 0 2
Circles have 2 intersect point
---
Input 1st Circle(x, y, r):0 0 1
Input 2nd Circle(x, y, r):5 0 2
Circles have 0 intersect point
```

### 【解答例】

## 【ボーナス課題】

### 05-c-01.c

#### 【課題】

入力として  $N$  個の三次元座標  $(x, y, z)$  を与え、その中の任意の 3 点を選び三角形を作成する。この時、作成した三角形の面積が最大となる座標の組み合わせを、その面積とともに出力するプログラムを作成してください。なお、面積が同じ座標の組み合わせが複数存在する場合、そのすべての組み合わせを出力すること。

#### 【制約】

$3 \leq N \leq 20$  とし、同じ座標は入力されないものとする。  
また、面積において差の絶対値が  $0.00001$  以下なら同じ面積と判断する。

#### 【ヒント】

`<math.h>` の使用を推奨。コンパイル時にオプションを付ける。その他、使い方、コンパイルの仕方などは各自調べてみてください。

#### 【実行例：入力は赤字】

```
Input N: 4
Input (x, y, z): 0 0 0
Input (x, y, z): 1 1 1
Input (x, y, z): 1 1 -1
Input (x, y, z): 2 2 0
Max area: 1.414214
Points: (0.000000, 0.000000, 0.000000), (1.000000, 1.000000, 1.000000), (1.000000, 1.000000, -1.000000)
Points: (0.000000, 0.000000, 0.000000), (1.000000, 1.000000, 1.000000), (2.000000, 2.000000, 0.000000)
Points: (0.000000, 0.000000, 0.000000), (1.000000, 1.000000, -1.000000), (2.000000, 2.000000, 0.000000)
Points: (1.000000, 1.000000, 1.000000), (1.000000, 1.000000, -1.000000), (2.000000, 2.000000, 0.000000)
---
Input N: 5
Input (x, y, z): 1 2 3
Input (x, y, z): 4 5 6
Input (x, y, z): 2 5 1
Input (x, y, z): 8 2 1
Input (x, y, z): 4 6 2
Max area: 17.363755
Points: (1.000000, 2.000000, 3.000000), (4.000000, 5.000000, 6.000000), (8.000000, 2.000000, 1.000000)
```

#### 【解答例】