

2022 年度
プログラミング第 1 同演習

第 8 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】

08-a-01.c

【課題】

正の整数を入力 n とし、 $1 \sim n$ までの和を出力するプログラムを、`while` 文を使って作成してください。

【実行例：入力赤字】

```
input a positive integer : 5
total : 15
---
input a positive integer : 10
total : 55
```

【解答例】

08-a-02.c

【課題】

キーボードから西暦、月を整数で入力し、その月が何日まであるか出力するプログラムを作成してください。うるう年を考慮すること。**switch** 文を用いること。

【ヒント】

西暦は 1 以上の整数、月は 1~12 の整数のみの入力を想定してよい。
うるう年の定義は以下の通りである。

- (1) 西暦年号が 4 で割り切れる年をうるう年とする。
- (2) (1) の例外として、西暦年号が 100 で割り切れて 400 で割り切れない年は平年とする。

【実行例：入力は赤字】

```
input year, month : 2019 11
This month has thirty days
---
input year, month : 2019 8
This month has thirty-one days
---
input year, month : 2019 2
This month has twenty-eight days
---
input year month : 2020 2
This month has twenty-nine days
```

【解答例】

08-a-03.c

【課題】

“Please help me!” の依頼に対して y(yes)か n(no)を入力とし、y に対しては” Thank you! ”、n に対しては” Huh? ”、と出力し、y と入力するまで依頼を続けるプログラムを作成してください。while 文内で break を用いること。

【ヒント】

ホワイトスペースに注意すること。もしホワイトスペースがわからない場合は検索を試みてください。

【実行例：入力赤字】

```
Please help me!  
input y or n : n  
Huh?  
Please help me!  
input y or n : n  
Huh?  
Please help me!  
input y or n : y  
Thank you!
```

【解答例】

08-a-04.c

【課題】

最大で 5 つまでの正の整数を入力とし、0 が入力されるか 5 つの正の整数が入力されたら合計を出力するプログラムを作成してください。負の整数の入力に対してはエラーを出力した後、次の入力を受け付けること。while 文内で continue を用いること。

【実行例：入力値は赤字】

```
input a positive integer :3
input a positive integer :-4
Error : invalid input
input a positive integer :10
input a positive integer :4
input a positive integer :0
total : 17
```

【解答例】

【標準課題】

08-b-01.c

【課題】

正の整数を入力し、約数をすべて表示するプログラムを作成してください。

- ・ 0、負の整数ならエラーを出力
- ・ do-while 文を使用すること。

【実行例：入力赤字】

```
Input number:50
50 is divisible by 1 2 5 10 25 50
---
Input number:36
36 is divisible by 1 2 3 4 6 9 12 18 36
---
Input number:1
1 is divisible by 1
---
Input number:0
Incorrect Input
```

【解答例】

08-b-02.c

【課題】

入力した正の整数を素因数分解するプログラムを作成してください。while 文を用いること。

【実行例：入力は赤字】

```
1200
1200 is multiplication of 2^4 3^1 5^2
---
1
1 is multiplication of 1
---
11880
11880 is multiplication of 2^3 3^3 5^1 11^1
```

【解答例】

08-b-03.c

【課題】

入力した整数を以下のフォーマットで出力するプログラムを作成してください。switch文を用いること。

- ・入力する整数は±9,999の範囲、この範囲を超えたらエラーを出す。
- ・0 - zero, 1 - ichi, 2 - ni, …, 10 - ju, 100 - hyaku, 1000 - sen, “-” - minus を組み合わせ、数字に対応した日本語のローマ字を出力

【実行例：入力値は赤字】

```
2306
ni sen san hyaku roku
---
-3145
minus san sen hyaku yon ju go
---
0
zero
---
99999
Input range is -9999 to 9999
```

【解答例】

【ボーナス課題】

08-c-01.c

【課題】

EOF が与えられるまで又は入力が 10 回行われるまで、1 次関数 $ax + by = c$ を実行例のように与え続け、最後に与えた関数それぞれの交点座標をすべて出力するプログラムを作成してください。同じ座標の出力は 1 回にすること。簡易化のため、 $a=0$ または $b=0$ の入力があった場合はエラーメッセージを表示し、入力を続けてください。入力は整数を想定しても実数を想定してもどちらでも良い。

【制約】

$(X1, Y1), (X2, Y2)$ があるとき $|X1 - X2| \leq 0.000001$ かつ $|Y1 - Y2| \leq 0.000001$ ならば同じ座標と判断する。

同じ関数は入力されないものとする。

関数の入力は 2 つ以上は必ず行われると想定してよい。

【実行例：入力は赤字】

```
Input a,b,c
1 1 0
0 0 0
Illegal input
1 -1 0
2 4 3
Ctrl^d
(0.000000, -0.000000)
(-1.500000, 1.500000)
(0.500000, 0.500000)
---
Input a,b,c
12 3 41
2 5 1
4 1 21
4 4 4
Ctrl^d
(3.740741, -1.296296)
(4.222222, -3.222222)
(5.777778, -2.111111)
(1.333333, -0.333333)
(6.666667, -5.666667)
```

【解答例】

08-c-02.c

【課題】
EOF が入力されるか、入力が 5 回行われるまで 2 次元座標(x, y)を入力し続け、最後にその座標で作られる図形の種類をヒントの中の中から選び出力するプログラムを作成してください。なお、図形は入力した座標の順に辺を結ぶものとする。また、入力は必ずヒントの中の何かしらの図形を生成するもののみと想定してよい（星形のようなものは生成されない）

【制約】
本問題においては 2 つの数値 A, B があるとき、 $|A - B| < 0.00001$ なら同じ数値と判断する。
同じ座標は入力されないものとする。
円周率は 3. 1415926535 とする。

【ヒント】
図形の種類

ten (一点)
tyokusen (直線)
sankakukei (ただの三角形)
nitouhen_sankakukei (二等辺三角形)
sei_sankakukei (正三角形)
sikakukei (ただの四角形)
daikei (台形)
heikousihenkei (平行四辺形)
hisigaka (ひし形)
seihoukei (正方形)
gokakukei (五角形)
sei_gokakukei (正五角形)

【実行例：入力は赤字】
Input x,y:
-1 0
1 0
0 5
Ctrl^d
nitouhen_sankakukei

Input x,y:
-2 -2
2 -2
3 4
-1 4
Ctrl^d
heikousihenkei

【解答例】