

2022 年度
プログラミング第 1 同演習

第 7 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】

07-a-01.c

【課題】

1 つの整数(i とする)を引数とし、1 から i までの整数を順番に出力する関数 `void print_sequence(int)`を再帰呼出しを使用して定義し、動作を確認するプログラムを作成してください。

【実行例：入力は赤字】

```
input an integer :5
1 2 3 4 5
---
input an integer :9
1 2 3 4 5 6 7 8 9
```

【解答例】

07-a-02.c

【課題】

2つの正整数を引数にとり、その2整数の最大公約数を入力する関数
`int greatest_common_divisor(int, int)`を再帰関数で定義し、2整数の最大公約数を入力するプログラムを作成してください。

【ヒント】

ユークリッドの互除法を使用する。

【実行例：入力値は赤字】

```
Input two integers :54 48
greatest common divisor is 6
---
Input two integers :21 49
greatest common divisor is 7
---
Input two integers :20 21
greatest common divisor is 1
```

【解答例】

07-a-03.c

【課題】

初項 a , 公差 d , 項数 n , の等差数列を出力する関数 `int tousa(int, int, int)` を再帰関数で定義し、各項を出力するプログラムを作成してください。

【ヒント】

出力は `tousa` 関数内と `main` 関数内、どちらでも構いません。

【実行例：入力は赤字】

Input a d n : **1 3 5**

[1] : 1

[2] : 4

[3] : 7

[4] : 10

[5] : 13

【解答例】

【標準課題】

07-b-01.c

【課題】

2つの整数 a , b を引数にとり、 a の b 乗を出力する関数 `float my_pow(int, int)` を再帰関数で定義し、結果を出力するプログラムを作成してください。数学関数ライブラリ `math.h` の使用は禁止します。

【実行例：入力は赤字】

```
Input two integers : 2 10
a^b = 1024.000000
---
Input two integers : 2 -3
a^b = 0.125000
---
Input two integers : -3 5
a^b = -243.000000
```

【解答例】

07-b-02.c

【課題】

2つの整数 m, n を入力し、 mPn および mCn を計算し出力するプログラムを作成してください。再帰呼び出しを用いること。

【ヒント】

$mPn = m! / (m - n)!$
 $mCn = mPn / n!$

【実行例：入力赤字】

```
input m and n :5 3
5 P 3 = 60
5 C 3 = 10
---
input m and n :8 4
8 P 4 = 1680
8 C 4 = 70
```

【解答例】

07-b-03.c

【課題】

入力として N を与えた後、 N 個の高さ付きの足場(正整数)を与える。1 個目の足場をスタート、 N 個目の足場をゴールとし、以下のルールで足場の上を進んでいく。

自分のいる足場を i 個目とすると $i+1$ 個目か $i+2$ 個目の足場に移動できる。
移動した際には、今いる足場の高さと移動先の足場の高さの差の絶対値のコストがかかる。
戻ることはできない。

この時、ゴールにたどり着く最小のコストを出力するプログラムを作成してください。

【制約】

$2 \leq N \leq 10$ とする。

$1 \leq$ 各足場の高さ ≤ 100

【ヒント】

動的計画法で解くのが基本ですが $N \leq 10$ 程度なら再帰でもできます。
動的計画法を知っているならばそれで解いていただいて構いません。(動的計画法の方が計算量は圧倒的に少ないです)

実行例 1 では

1 個目 \rightarrow 3 個目 \rightarrow 4 個目 \rightarrow 6 個目 \rightarrow 8 個目 \rightarrow 10 個目

で最小コスト 4 でできます。

【実行例：入力は赤字】

Input N : 10

2 9 3 4 1 3 2 3 1 2

Minimum cost is 4

Input N : 9

3 1 3 4 9 2 8 1 9

Minimum cost is 8

【解答例】

--

【ボーナス課題】

07-c-01.c

【課題】

入力として H, W を与えた後に H 行 W 列の迷路を与え、以下のルールで迷路を探索する。

スタート地点 : 5

通行可能 : 1

通行不可 : 0

ゴール地点 : 8

ある地点から移動できるのは上下左右 1 マスのみ

この時、スタート地点からゴール地点までたどり着くための最小移動回数を出力するプログラムを作成してください。ゴールにたどり着けない場合は-1 を出力してください。

【制約】

H ≤ 10, W ≤ 10 とする。

入力の各行の整数はスペース 1 つで区切られている。

スタート地点、ゴール地点はそれぞれ 1 つのみ。

【ヒント】

シンプルな経路探索です。

【実行例：入力は赤字】

Input H W : 4 5

1 1 5 1 1

1 0 0 0 1

1 0 8 1 1

1 1 1 0 0

Minimum move count is 6

Input H W : 4 3

1 1 5

0 0 0

8 1 1

1 1 1

Minimum move count is -1

Input H W : 10 10

8 1 1 1 1 1 1 1 1 1

0 0 0 0 0 0 0 0 0 1

0 1 1 1 1 1 1 1 0 1

0 1 0 0 0 0 0 1 0 1

0 1 0 1 1 1 0 1 0 1

0 1 0 1 0 5 0 1 0 1

0 1 0 1 0 0 0 1 0 1

0 1 0 1 1 1 1 1 0 1

0 1 0 0 0 0 0 0 0 1

0 1 1 1 1 1 1 1 1 1

Minimum move count is 54

【解答例】

--