

2022 年度
プログラミング第 1 同演習

第 4 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】

04-a-01.c

【課題】

正の整数を引数 `n` とし、`n` 個のアスタリスクを出力する関数 `aster(int)` を作成してください。

【実行例：入力赤字】

```
input a positive integer : 4
****
---
input a positive integer : 6
*****
```

【解答例】

04-a-02.c

【課題】

int 型で定義したグローバル変数 `x` の値を、引数の値(キーボードからの入力値とする)に変更する関数 `assign(int)`を作成してください。

【ヒント】

main 関数には以下を記述し、assign 関数には処理を行ったことが分かる出力を含むこと。

```
int main(){
    int a;

    printf("input an integer :");
    scanf("%d", &a);
    printf("x :%d\n", x);
    assign(a);
    printf("x :%d\n", x);

    return 0;
}
```

【実行例：入力値は赤字】

```
input an integer : 10
x : 0
assigned.
x : 10
```

【解答例】

04-a-03.c

【課題】

引数として2つの実数と1つの四則演算子 (+, -, *, /) をとり、演算子に対応した計算結果を返す関数 `float calc(float, float, char)` を作成してください。

【ヒント】

0での除算に対しては0を返すようにすること。
また `main` 関数には以下を記述すること。

```
int main(void){  
    float a = 1234, b = 5678;  
    printf("%f + %f = %f\n", a, b, calc(a, b, '+'));  
    printf("%f - %f = %f\n", a, b, calc(a, b, '-'));  
    printf("%f * %f = %f\n", a, b, calc(a, b, '*'));  
    printf("%f / %f = %f\n", a, b, calc(a, b, '/'));  
    return 0;  
}
```

【実行例：入力は赤字】

```
1234.000000 + 5678.000000 = 6912.000000  
1234.000000 - 5678.000000 = -4444.000000  
1234.000000 * 5678.000000 = 7006652.000000  
1234.000000 / 5678.000000 = 0.217330
```

【解答例】

04-a-04.c

【課題】

キーボードから整数を 1 つ入力し、変数に読み込んだ整数の値と変数のアドレスを表示するプログラムを、ポインタを用いて作成してください。

【ヒント】

実行例では int 型の変数名は a としている。

アドレスの表示には%pを用いること。

```
printf("address of a is %p\n", &a);
```

```
printf("value of a is %d\n", a);
```

は不正解

【実行例：入力値は赤字】

input an integer : **3**

address of a is 0x0028fed8 （アドレスはパソコンそれぞれで異なるので注意）

value of a is 3

【解答例】

04-a-05.c

【課題】

int 型の変数に格納されている値を 100 に変更しようとする 2 つの関数
void hundred1(int) と void hundred2(int*)を作成し、それぞれ動作を確認するプログラムを作成してください。

【ヒント】

変数の初期値は何でもよい。実行例では変数名は a, b 初期値は 500 としている。実行例では変数 a、b をそれぞれ void hundred1(int)、void hundred2(int*)に渡している。

【実行例：入力赤字】

before : a = 500, b = 500
after : a = 500, b = 100

【解答例】

04-a-06.c

【課題】

ポインタを用いて、2つの整数の和を計算する関数 `sum(int *, int *)` を作成してください。

【ヒント】

main 関数は以下のように記述すること。

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

    printf("input 2 integers :");
    scanf("%d %d", &a, &b);
    printf("%d + %d = %d\n", a, b, sum(&a, &b));

    return 0;
}
```

【実行例：入力値は赤字】

```
input 2 integers : 3 5
3 + 5 = 8
```

【解答例】

04-a-07.c

【課題】

二つの float 型変数に格納されている値を入れ替える関数 `void swap(float*, float*)` を作成し、入力した二つの数を入れ替えて出力するプログラムを作成してください。

【実行例：入力赤字】

Input 1st number: **2.5**
Input 2nd number: **3.6**
Swapped 1st: 3.600000, 2nd: 2.500000

【解答例】

【標準課題】

04-b-01.c

【課題】

‘E’ が入力されるまで一桁の正の整数を入力し続け、最後に総和と平均を出力するプログラムを作成してください。‘E’、数字以外が入力されたらエラーを出力してください。最初の一回目は何かしらの正の整数が入力されることを想定して作って良い。

【ヒント】

入力プロンプトは表示しなくてよい。改行を無視する構造を入れること。数字も `char` 型の入力として扱い、数値の計算に工夫をすること。最初は何かしらの整数が入力されることを想定して良い。

【実行例：入力は赤字】

```
3
6
A
Illegal input: A
9
p
Illegal input: p
1
E
Sum is: 19
Average is: 4.750000
```

【解答例】

04-b-02.c

【課題】

`int i, int *ip, char c, char *cp` をそれぞれ宣言し、`ip` には `i` のアドレス、`cp` には `c` のアドレスを代入し、`ip`, `cp` をそれぞれインクリメントした際の値の変化を出力するプログラムを作成せよ。`ip`、`cp` それぞれの値の変化について考察せよ（考察に関して記述して提出する必要はない）

【実行例：入力は赤字】

```
ip:  0x7fff8989dc6c
++ip: 0x7fff8989dc70
cp:  0x7fff8989dc6b
++cp: 0x7fff8989dc6c
```

【解答例】

04-b-03.c

【課題】

char 型ポインタを引数とし、格納されている文字が小文字なら大文字に変換する関数 void mytoupper(char *)を作成し、入力した小文字を大文字に変換するプログラムを作成してください。

【ヒント】

ascii コード n の小文字アルファベットを大文字に変換: $n = n - 'a' + 'A'$

【実行例：入力は赤字】

```
Input character:a
Output character:A
---
Input character:A
Output character:A
---
Input character:z
Output character:Z
---
Input character:4
Output character:4
```

【解答例】

【ボーナス課題】

04-c-01.c

【課題】

暗号の 1 つとしてシーザー暗号がある。これは文の各文字を一定数辞書順にシフトするものである。入力にシフト数とアルファベット小文字からなる文字列を取り、シーザー暗号を作成するプログラムを作成してください。シフト数は 26 の絶対値を超えないものとする。平文の文字数は、実行例の文字数程度を想定していれば良い。

【ヒント】

`scanf()` で複数の文字を読み取るときは `for` ループで `scanf()` を複数回呼び出す。その時は改行文字 `'\n'` を読み取ったときに `for` ループを終わるようにする。また、`scanf()` で数値を読み取った直後の改行文字の取り扱いにも注意。数値を読み取った直後に `scanf()` で 1 文字読み取ると改行文字 `'\n'` が読み取られます。

【実行例：入力は赤字】

```
Input shift num : 4
Input text : abcdefgxyz
Cipher text : efghijkbcd
---
Input shift num : -4
Input text : abcdefgxyz
Cipher text : wxyzabctuv
---
Input shift num : 0
Input text : abcdefgxyz
Cipher text : abcdefgxyz
```

【解答例】

04-c-02.c

【課題】

16 進数で入力された数値を 10 進数で表示するプログラムを作成してください。

【ヒント】

`scanf()`で複数の文字を読み取るときは `for` ループで `scanf()`を複数回呼び出す。
その時は改行文字 `'\n'`を読み取ったときに `for` ループを終わるようにする。
また、`scanf()`で数値を読み取った直後の改行文字の取り扱いにも注意。数値を読み取った直後に `scanf()`で 1 文字読み取ると改行文字 `'\n'`が読み取られます。

【実行例：入力は赤字】

```
Input hexadecimal number: 1234
Decimal number is 4660
---
Input hexadecimal number: cd65fe
Decimal number is 13460990
---
Input hexadecimal number: ffffffff
Decimal number is 268435455
```

【解答例】

04-c-03.c

【課題】

1~5 の乱数を発生させる関数 `int myrand()` および入力した整数を引数として乱数と一致すれば 1 点を与える `void game(int)` 関数を作成し、数当てゲームを行うプログラムを作成してください。入力は 10 回まで行えるものとし、点数が 3 点になったときゲームクリアとする。

【ヒント】

擬似乱数は漸化式 $X_{n+1} = (13 * X_n + 5) \bmod 24$, $X_1 = 8$ によって与えられるものとする。 $\bmod 24$ となっているため、さらに出力結果に一工夫を加える必要はある。これを用いて `myrand()` を作成すること。

【実行例：入力は赤字】

```
Please expect a number (1 to 5) :4
your expectation : 4, random number : 4
Correct! your score : 1
remaining : 9
...
Please expect a number (1 to 5) :4
your expectation : 4, random number : 3
Incorrect. your score : 2
remaining : 0
Game over.
```

【解答例】

04-c-04.c

【課題】

英字 1 文字の大文字・小文字を反転させる関数 `char alpharev(char)` を作成し、入力された文字列の大文字・小文字を反転させるプログラムを作成して下さい。
Ctrl-d でプログラムを停止できるようにすること。

【ヒント】

`scanf()` で複数の文字を読み取るときは `for` ループで `scanf()` を複数回呼び出す。
その時は改行文字 `'\n'` を読み取ったときに `for` ループを終わるようにする。
また、`scanf()` で数値を読み取った直後の改行文字の取り扱いにも注意。数値を読み取った直後に `scanf()` で 1 文字読み取ると改行文字 `'\n'` が読み取られます。

【実行例：入力は赤字】

```
% ./a.out
Hello #1234 WORLD;
hELLO #1234 world;
ProGramMing Tanoshi!!!
pROgRAMmING tANOSHI!!!
^d
```

【解答例】

【チャレンジ課題】

04-d-01.c

【課題】

有名なパズルゲームの1つにハノイの塔がある。プログラムの入力として、ハノイの塔の段数 N 、実行手数 T を与える。このパズルゲームを最短手数で解くとき、 T 手後の状態を実行例のように出力する関数 `void HanoiVisualize(int, int)` を作成してください。ハノイの塔の石は小さいものから 1,2,3,4,・・・と表現する。

【制約】

$1 \leq N \leq 20$ とし、 $T \leq 2^N - 1$ とする。

再帰関数の使用は禁止する。

【実行例：入力は赤字】

Input N T : 4 11

```
1
2 3 4
...
```

Input N T : 4 0

```
1
2
3
4
...
```

Input N T : 4 15

```
1
2
3
4
...
```

Input N T : 10 532

```
3
6 1
7 2
8 5
4 9 10
```

【解答例】