

2022 年度
プログラミング第 1 同演習

第 10 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】

10-a-01.c

【課題】

fgets を用いて標準入力から文字列を取得し、標準出力に表示せよ。
定数 MAX_LEN を 256 と宣言してかまわない。

【実行例：入力は赤字】

I'm student.

I'm student.

【解答例】

--

10-a-02.c

【課題】

`strncpy()`を用いて、標準入力から取得した文字列の最初の 3 文字を出力するプログラムを作成してください。

定数 `MAX_LEN` を 256 と宣言してかまわない。

【ヒント】

講義スライドの `strncpy()`の正しい使い方を参考にする事。

【実行例：入力赤字】

abcdefg

abc

【解答例】

10-a-03.c

【課題】

2つの文字列を入力し、二つの文字列が等しいか否かを出力するプログラムを作成してください。

定数 `MAX_LEN` を 256 と宣言してかまわない。

【実行例：入力は赤字】

```
input first string :abc
input second string :abc
same.
---
input first string :abc
input second string :abcd
different.
```

【解答例】

10-a-04.c

【課題】

文字列および鍵を入力し、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。文字列の読み込みには `fgets()` を用いること。

定数 `MAX_LEN` を `256` と宣言してかまわない。

【ヒント】

シーザー暗号とは、各文字を辞書順に鍵の個数だけシフトすることで作成できる暗号である。今回はアスキーコードに準じてシフトすればよい。

アスキーコードの範囲内についてエラー処理をしても良いが、範囲を超えた入力によってエラーが出てきても良い。

【実行例：入力は赤字】

```
input string :Hello World!  
input key : 3  
Khoor#Zruog$  
---  
input string :Pro1 is interesting.  
input key : 2  
Rtq3"ku"kpvgtguvkpi0
```

【解答例】

10-a-05.c

【課題】

3つの文字列を入力し、一番長い文字列を最終的に出力するプログラムを作成してください。

定数 MAX_LEN を 256 と宣言してかまわない。

【ヒント】

一番長い文字列が複数ある場合は、先に入力された方の文字列を出力すること。

【実行例：入力値は赤字】

```
input a string (1) :a
input a string (2) :abc
input a string (3) :de
max :abc
---
input a string (1) :abcde
input a string (2) :12345
input a string (3) :xyz
max :abcde
```

【解答例】

10-a-06.c

【課題】

strlen 関数と同じ機能を持つ関数 `int mystrlen(char *s)` を定義し、strlen 関数と比較せよ。

定数 `MAX_LEN` を `256` と宣言してかまわない。

【実行例：入力赤字】

I'm student.

Length(mystrlen): 13

Length(strlen) : 13

【解答例】

10-a-07.c

【課題】

関数「`double my_average(double x)`」は、引数として `double` 型変数を 1 つ受け取り、これまでに引数として与えられた数の平均値を返す。戻り値の型は `double` 型とし、ここでは以下のような利用方法を想定している。

```
int main()
{
    printf("%f\n", my_average(3.3));
    printf("%f\n", my_average(4.4));
    printf("%f\n", my_average(5.5));
    printf("%f\n", my_average(6.6));

    return 0;
}
```

`my_average` 関数の定義部（本体）を書き加えてプログラムを完成させなさい。ただし、本プログラムでは外部変数は使わずに、`my_average` 関数内で `static` 宣言をした静的なローカル変数を使いなさい。`my_average` 関数の関数定義の位置に依らず、関数のプロトタイプ宣言を行いなさい。

【実行例：入力値は赤字】

```
3.300000
3.850000
4.400000
4.950000
```

【解答例】

【標準課題】

10-b-01.c

【課題】

文字列処理関数の一つに、2つの文字列を連結する `strcat` 関数がある。これと同じ動作をする関数 `char *mystreat(char *s1, char *s2)` を作成し、動作を確認してください。
(`strcat()` の用法については各自調べること。)
定数 `MAX_LEN` を 256 と宣言してかまわない。

【ヒント】

実行例の場合、`main` 関数の記述は以下の通り。

```
int main()
{
    char str1[MAX_LEN] = "abc", str2[] = "def";

    mystreat(str1, str2);
    printf("%s", str1);

    return 0;
}
```

※`MAX_LEN` は適宜定義すること。

【実行例：入力赤字】

abcdef

【解答例】

10-b-02.c

【課題】

文字列 `s2` を逆順にした文字列 `s1` を作成する関数
`void mystrrrev(char *s1, char *s2)` を作成して使用せよ。
ただし、定数 `MAX_LEN 256` を宣言し、作成される文字列がこれを超えないようにすること。

【ヒント】

入力文字には改行が含まれていることを条件に処理をすること。

【実行例：入力は赤字】

I'm student.
[IN]I'm student.
[REV].tneduts m'I

【解答例】

10-b-03.c

【課題】

文字列 `s2` を `n` 回繰り返した文字列 `s1` を作成する関数
`void strrpt(char *s1, char *s2, int n)` を作成して使用せよ。
ただし、定数 `MAX_LEN 256` を宣言し、作成される文字列がこれを超えないようにすること。`strcat` は用いてはならない。

【実行例：入力は赤字】

Input string:**I'm student.**
Input repeat time:**3**
[ORG]:I'm student.
[RPT]:I'm student.I'm student.I'm student.

【解答例】

10-b-04.c

【課題】

1 文と 1 単語を入力し、文の中にその単語があるか判定するプログラムを作成してください。strstr()は用いてはならない。文、単語ともに 30 文字までとする。

【実行例：入力は赤字】

Input a sentence : asdfghjklzxcvbnm

Input a word : fgh

The word (fgh) exists

Input a sentence : asdfghjklzxcvbnm

Input a word : kjhgfds

The word (kjhgfds) doesn't exist

【解答例】

【ボーナス課題】

10-c-01.c

【課題】

EOF が入力されるまでに入力された単語を辞書順に出力するプログラムを作成してください。単語の文字数は最大 30 文字、入力数は最大 20 個までとする。

【ヒント】

strcmp()の返り値を利用する。

【実行例：入力は赤字】

```
dog
cat
star
go
exit
example
ascii
abc
aac
aaa
^d
aaa
aac
abc
ascii
cat
dog
example
exit
go
star
```

【解答例】

10-c-02.c

【課題】

文字列 `s` を引数にとり、これまでに与えられた `s` 内部に含まれる数値の和を返す関数 `double sumofstrnum(char*)` を作成しなさい。`main` 関数は以下のヒントのように書くこと。グローバル変数は使用してはならない。
`s` 内部に含まれる数値とは文字と文字に挟まれた数字部分である。
数値の種類は整数、小数のみと定める。

【ヒント】

`main` 関数は以下のように書くこと。

```
int main() {  
    printf("%lf\n", sumofstrnum("0011abc-20a2bc"));  
    printf("%lf\n", sumofstrnum("123"));  
    printf("%lf\n", sumofstrnum("aabbcca1b2b3"));  
    printf("%lf\n", sumofstrnum("aabb1.2bb23.560"));  
    printf("%lf\n", sumofstrnum("aabb-1.2bb-23.56p-"));  
    printf("%lf\n", sumofstrnum("aabb-23s34ccs2.0"));  
    return 0;  
}
```

【実行例：入力は赤字】

```
-7.000000  
116.000000  
122.000000  
146.760000  
122.000000  
135.000000
```

【解答例】