

#01
CPU・メモリ・機械語・変数・スタック
実習・課題内容
2022 年度 / プログラミング及び実習 III

角川裕次

龍谷大学 先端理工学部

今回第 1 回目は実習・課題ともに manaba への提出は不要です

- manaba に提出練習用の提出先を用意します
- 自由に提出練習して OK; 採点しません

次ページ以降 (次回以降に向けた参考):

次回からどのような形式で提出物を用意すべきかを示します

今回はプログラミング環境設定も兼ねて実習を行って下さい

- 来週から提出が必須となります

課題・実習の提出先: manaba 「レポート」

manaba 「レポート」の今回の提出先へ提出のこと

注意

- ファイル名は指定の通りにすること
- ファイル形式は指定の通りにすること
- 守られていない場合は採点しない場合がある
(採点作業の軽減への協力を宜しく)

提出ファイル不足 (特に「説明と考察」): その回は0点とする場合あり

実習・課題: 提出物のファイル形式に関する一般的注意

C 言語ソースコード (プログラムコード)

- 一文字ずつ手打ちで入力: 体を動かして頭の中に叩き込む
- ファイル形式: C ソースファイル (.c; ファイルそのもの)
- (不可: スクリーンショット画像や Word に流し込んだものなど)

実行結果スクリーンショット画像

- ファイル形式: 画像ファイル形式 (.jpeg や .png など)
- スクリーンショット取得ツールを使用のこと.
- (不可: デジカメによるディスプレイ撮影画像)

説明と考察: プログラム及び実行結果に対する説明と考察

- ファイル形式: テキストファイル (.txt)
- 指定文字数以上の「説明と考察」を書くこと.
- 講義で学んだ C 言語の機能とからめて説明
- (不可: Word や PDF など)
- (不可: ソースコードの「朗読」. 「朗読」は説明や考察でない)

★重要★

「説明と考察」って？

(必読: 毎回の提出に関係します)

例: このプログラムの「説明と考察」は?

```
char *strncat(char *s1, const char *s2, size_t n)
{
    char *tmp = s1;
    while (*s1)    /* s1を末尾まで進める */
        s1++;
    while (n--)
        if (!(*s1++ = *s2++))
            break; /* 途中に'\0'があれば終了 */
    *s1 = '\0'; /* s1の最後に'\0'を入れる */
    return tmp;
}
```

文字列 s1 に文字列 s2 を追加する関数

ただし追加結果の文字列長は $n - 1$ 文字までに制約

これは「朗読」：これではだめです

最初にポインタ変数 `tmp` を宣言して、初期値を `s1` にします。
そして `while` 文で `*s1` の間、`s1` を 1 増やします。次に `n` が成立する間、`*s1++ = *s2++` を実行しますが、不成立なら `while` 文を終了します。
最後に `*s1` に `\0` を代入して、`tmp` の値を関数値として返します。

これを読んでプログラムコードを理解できますか？

視点：単に C 言語の要素を日本語単語に置き換えただけ

- プログラムを理解していると主張されても納得しません
- そんな説明ではプログラムを読まされる側は理解できない

補足：詩や小説の本物の朗読は意味を深く理解していないと出来ないです

「説明と考察」はこんな感じで書く

以下の流れでプログラムは動作します。

- 1 まず最初にポインタ変数 `s1` が指す文字列の末尾、つまり文字列終端 `\0` の位置を指すよう、`s1` の値を更新します。(その結果、`*s1 == \0` になります.)
- 2 次に、`s1` の指す位置から、ポインタ変数 `s2` が指す文字列の先頭 1 文字目から、順々に書き込んでゆきます。その際に、最大書き込み文字数 `n` を 1 ずつ減らしてゆきます。
- 3 `s2` が指す文字列の末尾に来れば書き込みは終わり、`while` 文を抜けます。末尾の判定は、書き込んだ文字が文字列終端 `\0` かどうかで判断します。末尾が来る前に `n` が 0 になると書き込み文字数に達したことになるので、`while` 文を抜けます。
- 4 `n == 0` で `while` 文を抜けた場合には文字列の終端に `\0` が置かれていないため、`\0` を書き込む必要があります。(if 文による `break` 文の実行で `while` 文を抜けた場合には、既に `\0` は書き込まれているので 2 回続けて書き込まれます.)

以上により、文字列 `s1` の末尾へ文字列 `s2` が追加されます。ただし変数 `n` により追加する文字数を勘定しているので、その数を超えた文字数を追加することはありません。
(487 文字)

Q: 「説明と考察」が書けません

A: 「理解できていないからです」

「自分の言葉で書けない」＝「自分の頭で考えてない/理解できていない」

- 説明をしっかりと書けるまで理解に努める

「自分の言葉で書ける」＝「自分の頭で考えている/理解できている」

- 分かりやすい丁寧な文章書きにチャレンジ
- 試行錯誤しながら何度も書き換えをしてみないと身につかない
- 自分の理解度のチェックにもなる
- 「自分で書いたプログラムからちゃんと理解している」は大抵幻想
- ★卒論/就職後/あらゆるところで役立つ重要スキル★

実習

実習 A

★提出不要 (提出は任意) ★

p.147, List 6-2 (3 つの整数の最大値)
の理解・入力・実行・動作確認

提出物: 以下のファイル名で提出のこと

- j01a.c (C 言語ソースコード)
- j01a (実行結果スクリーンショット; ファイル拡張子は適宜)
- j01a.txt (説明と考察 300 文字以上),

★提出不要 (提出は任意) ★

p.149, List 6-4 (4 つの整数の最大値)
の理解・入力・実行・動作確認

提出物: 以下のファイル名で提出のこと

- j01b.c (C 言語ソースコード)
- j01b (実行結果スクリーンショット; ファイル拡張子は適宜)
- j01b.txt (説明と考察 300 文字以上),

★提出不要 (提出は任意) ★

p.150, List 6-6 (べき乗を求める, 別実装)
の理解・入力・実行・動作確認

提出物: 以下のファイル名で提出のこと

- j01c.c (C 言語ソースコード)
- j01c (実行結果スクリーンショット; ファイル拡張子は適宜)
- j01c.txt (説明と考察 300 文字以上),

課題

★提出不要 (提出は任意) ★

p.151, 演習 6-5

1 から n までの全整数の和を求めて返却する関数を作成せよ.

- 動作確認をする main 関数も作成すること
- スクリーンショットのとり方を身につける

提出方法の練習をしましょう

- 講義終了時間まで各自取り組むこと
- プログラミング環境の整備/確認を行って下さい

課題 (説明のつづき)

★提出不要 (提出は任意) ★

提出物: 以下のファイル名で提出のこと

- k01a.c (C 言語ソースコード)
- k01a (実行結果スクリーンショット; ファイル拡張子は適宜)
- k01a.txt (説明と考察 300 文字以上),

プログラム全体像

```
#include <stdio.h>
int sumup(int n) {
    ここを作成してください
}
int main(void) {
    for (int i = 0; i <= 20; i++) {
        printf("SUM(0,...,%d)=%d\n", i, sumup(i));
    }
    return 0;
}
```

0 始まりに注意

おわり