

## 計算機基礎実習II 期末試験問題 2019/01/17

### 試験の実施方法

kiso2 コマンドで「試験」表示されたファイルは、試験として実施されます。「試験」として出題されたファイルもkiso2 コマンドを用いて提出しますが、これらのファイルは試験実施中のみ提出することができます。

試験の実施方法は、以下の通りです。

1. 端末(ターミナル)上から、kiso2 コマンドに `-start-exam` オプションを付けて実行 (`kiso2 -start-exam`) することで試験が開始されます。表示フォントの関係で見にくいかもしれませんが、kiso2 の後にスペースを開けて `-start-exam` (先頭のハイフンは2つ「マイナス・マイナス」`start`「マイナス」`exam`) です。
2. 試験が開始されると、試験開始のメッセージと共にウェブブラウザで試験用Webページが自動的に表示されます。
3. 試験の実施中は、端末(ターミナル)上で経過時間と残り時間が不定期に更新され、kiso2 コマンドで「試験」表示されたファイルが提出できます。
4. kiso2 コマンドで「試験」ファイルを提出する場合は、kiso2 `-start-exam` で試験を開始した端末(ターミナル)とは別の端末(ターミナル)をもう一つ起動し、その端末でkiso2 コマンドによるファイルの提出を行ってください。
5. 試験実施中は、不定期にデスクトップ画面のスクリーンショットが取得されます。
6. 試験の残り時間が100秒を切ると1秒毎のカウントダウンが表示され、残り時間が0になると `kiso2 -start-exam` コマンドと共に試験が終了します。
7. `kiso2 -start-exam` コマンドが強制終了されると、そのままでは再度 `kiso2 -start-exam` を実行できなくなります。操作を誤って強制終了してしまった場合は、速やかに試験監督に申し出て下さい。再度 `kiso2 -start-exam` コマンドが実行できるようにします。ただし、再起動された `kiso2 -start-exam` コマンドは、再起動される度に残り試験時間が10分ずつ減るので注意して下さい。

### 解答方法と採点方法

問0と問9は必ず解答 提出してください。

問1から問8は選択問題として自由に解答してください(1つも解答しなくても、全て解答しても構いません)。ただし、試験時間は限られていますので時間配分には十分注意して解答してください。kiso2 コマンドで提出されたもののうちcc コマンドでコンパイルが可能なすべてのプログラムについて、それぞれ10点満点、または15点満点で採点されます。コンパイルができないプログラムソースは、提出しても採点の対象にはならないので注意してください。

プログラムの採点は、各問の実行例で示したものを含んだ複数の入力値に対する出力結果によって評価されます。関数定義を行う幾つかの問題では、提出されたプログラムソースから関数定義を抽出し、評価用の `main()` 関数によって採点が行われます。

提出プログラムが、採点用に用意された入出力結果を満たさなかった場合は、プログラムソースの内容で評価を行うことがあります。その場合、インデント(字下げ)や変数名などのプログラムソースの読みやすさの観点からも評価が行われます。

この試験全体は50点満点で評価されます。ただし、全ての問題の総点は50点を超えるよう

に配点されています。試験結果の合計点が50点を超えた場合、超えた点数は切り捨てられ50点として扱われます。

## 問0 exam15-0.txt (5点)

以下のリンクにあるアンケートフォームに回答して下さい。

[C言語についての理解度調査その3 kiso2-2018](#)

「C言語に関する単語や概念について、現在の理解度を自分がどのように感じているかお答え下さい。」

回答が終わったらフォーム画面下の「送信(Submit)」ボタンを忘れずに押して下さい。アンケートに回答後、「自分の学籍番号」が書かれたテキストファイル **exam15-0.txt** を作成し、kiso2 コマンドを用いて提出しなさい。

第2回、および第8回実施の中間試験とほぼ同内容です。これまでの回答との整合性に基づき採点されず。

## 問1 exam15-1.c (10点)

次のプログラムソースは、キーボードからASCII文字を1文字ずつ繰り返し入力すると、入力された文字が英小文字('a'-'z')であれば入力された文字をそのまま出力し、それ以外の場合は実行を終了するプログラムである。

[exam15-1.c](#)

```
#include <stdio.h>

int lower(char);

int main(void) {
    char c = 'a';

    while (lower(c)) {
        scanf(" %c", &c); // 改行文字を読み飛ばすために %c の前に空白がある
        printf(" %c\n", c); // ずらした文字であることを示すために先頭に空白を挿入
    }

    return 0;
}

int lower(char c) {
    if ('a' <= c && c <= 'z')
        return 1;
    else
        return 0;
}
```

このプログラムソースを利用し、まず、整数  $s$  の値をキーボードから入力すると、引き続いて繰り返し入力される英小文字を  $s$  だけ後にずらした英小文字を出力するプログラム **exam15-1.c** を作成し、`kiso2` コマンドを用いて提出しなさい。

このとき「 $s$  だけ後にずらす」とは、たとえば  $s=1$  のとき 'a' は 'b' へ、'b' は 'c' に変換され、また  $s=2$  のとき 'a' は 'c' へ、'b' は 'd' のように、アルファベット順で  $s$  文字だけ後の文字に変換することを指す。

ただし  $s$  だけ後にずらした文字が 'z' を超えている場合 'z' の直後の文字は 'a' と考え、英小文字の集合が循環しているような順序に基づいた英小文字を出力すること。さらに  $s$  に  $0 \sim 26$  以外の値が入力された場合、プログラムは何も出力せずに終了し、また、入力された文字が英小文字以外の場合は繰り返しを終了すると共に、実行も終了するものとする。

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-1
-1
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-1
27
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-1
0
a
a
b
b
z
z
A
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-1
1
a
b
b
c
z
a
A
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-1
2
a
c
b
d
z
b
A
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-1
25
a
z
b
a
```

```

z
y
0
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-1
26
a
a
b
b
z
z
#

```

## 問2 exam15-2.c (10点)

2つの正の整数  $N_k, N_{\ell}$  の値を順に入力すると  $\sum_{k=1}^{N_k} \sum_{\ell=1}^{N_{\ell}} \frac{k^{\frac{1}{2}}}{\ell^2}$  の値を出力するプログラム **exam15-2.c** を作成し、kiso2 コマンドを用いて提出しなさい。ただし  $N_k, N_{\ell}$  いずれかに 0 以下の値が入力された場合、プログラムは 0.0 を値として出力するものとする。

実行例：

```

t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-2
0 0
0.000000
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-2
1 0
0.000000
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-2
0 1
0.000000
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-2
1 1
1.000000
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-2
1 2
1.250000
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-2
2 1
2.414214
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-2
3 4
5.902668
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-2
9 10
29.919817

```

### 問3 exam15-3.c (10点)

$a \leq b$  を満たす2つの整数  $a, b$  の値を入力すると  $a \leq i \leq b$  の範囲で  $i$  行目に  $i^2$  個の '\*' (半角アスタリスク) を出力するプログラム **exam15-3.c** を作成し、`kiso2` コマンドを用いて提出しなさい。ただし、入力された値が  $a \leq b$  を満たさない場合、プログラムは何も出力せずに終了する。

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-3
0 -1
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-3
0 0

t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-3
2 2
****
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-3
0 1

*
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-3
0 5

*
****
*****
*****
*****
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-3
-2 2
****
*

*
****
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-3
-5 8
*****
*****
*****
****
*

*
****
*****
*****
*****
*****
*****
*****
```

\*\*\*\*\*

## 問4 exam15-4.c (10点)

要素の値が 0 から 20 の範囲にある大きさ 50 の整数型の配列変数 `data[]` が以下のように与えられている。0 から 20 の値が、配列 `data[]` の要素にそれぞれいくつ含まれているかを出力するプログラム

**exam15-4.c** を作成して `kiso2` コマンドを用いて提出しなさい。ただし、プログラムは `data[]` に出現する要素の最小値から最大値までの範囲のみを出力するものとする。

```
int data[] = {17, 6, 13, 4, 7, 4, 13, 9, 16, 10, 15, 11, 10, 6, 7, 10, 4,
14, 3, 13, 16, 9, 4, 14, 17,
           6, 14, 11, 8, 3, 8, 11, 4, 10, 10, 15, 4, 17, 13, 10, 11, 13,
9, 15, 4, 14, 6, 16, 7, 11};
```

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/exam15-4$ ./exam15-4
3: 2
4: 7
5: 0
6: 4
7: 3
8: 2
9: 3
10: 6
11: 5
12: 0
13: 5
14: 4
15: 3
16: 3
17: 3
```

このプログラムの採点では、ここで与えられたものとは異なる `data[]` 要素の値でチェックが行われます。

## 問5 exam15-5.c (15点)

関数 `checkd()` は、1つの `long` 型の引数を受け取り、`int` 型の返り値を返す関数である。`checkd()` は、引数として受け取った整数値を 10 進数表記したときの各桁の数値をすべて加算した値を返り値として返す。また、`checkd()` の返り値は、受け取った引数の値と同じ正負の符号をもつものとする。

このような関数 `checkd()` を定義し、次のような `main()` 関数で実行されるプログラム **exam15-5.c** を作成し、`kiso2` コマンドを用いて提出しなさい。ただし、以下のプログラムソースの `main()` 関数は変更してはならない。

exam15-5.c

```
#include <stdio.h>

// chekud() のプロトタイプ宣言
```

```
int main(void) {
    long n;
    scanf("%ld", &n);

    printf("%d\n", checkd(n));
    return 0;
}

// checkd() の定義
```

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-5
0
0
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-5
-1
-1
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-5
1
1
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-5
123
6
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-5
-321
-6
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-5
123456789987654321
90
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-5
-12345678998765432
-89
```

このプログラムの採点では、提出されたファイルから関数 `checkd()` を抽出し、採点用のソースファイルの実行結果によって評価されます。

## 問6□exam15-6.c (15点)

関数 `yearday()` は、日付の月と日を表す2つの `int` 型の引数を受け取り、その年の1月1日からの延べ日数を `int` 型の返り値として返す関数である。

以下のプログラムソースでは、グローバル変数として各月の日数が格納されている `int` 型の配列変数 `mdays[]` が定義されており、キーボードから月と日を表す整数値を入力すると、関数 `yearday()` を用いて入力された日付の1月1日からの延べ日数を出力することが期待されている。

グローバル変数 `mdyas[]` を利用して延べ日数を計算する関数 `yearday()` を定義すると共に、このプログラム **exam15-6.c** を完成させ `kiso2` コマンドを用いて提出しなさい。ただし、月として入力された値 `m` が 1 ~ 12(月)ではない場合、または、日として入力された値 `d` がその月の日数の範囲外である場合は、再度、月と日の入力を行うものとする。

## exam15-6.c

```
#include <stdio.h>

// yearday() のプロトタイプ宣言

int mdays[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

int main(void) {
    int m, d;

    scanf("%d%d", &m, &d);

    printf("%d\n", yearday(m, d));

    return 0;
}

// yearday() の定義
```

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-6
1 32
1 0
13 1
12 31
365
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-6
1 1
1
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-6
2 1
32
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-6
7 7
188
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-6
9 13
256
```

この問題では *main()* 関数を変更する必要があります。

### 問7 **exam15-7.c (10点)**

関数 *addcoin()* は、1円、5円、10円、50円、100円、500円の6種類の日本円硬貨それぞれの枚数を6つの *int* 型の引数として受け取り、それらの総額を *int* 型の返り値を返す関数である。

このような関数 *addcoin()* を定義し、*addcoin()* を利用して6種類の硬貨の枚数を整数として入力すると、その総額を出力するプログラム **exam15-7.c** を作成し、kiso2 コマンドを用いて提出しなさい。



以下の実行例では、`main()` 関数によって硬貨枚数の入力と総額の出力が繰り返し行われているが、提出するプログラムでは必ずしもこれらを実現する必要はない。

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-7
1 1 1 1 1 1
666
1 0 0 0 0 0
1
0 1 0 0 0 0
5
0 0 0 0 0 1
500
1 2 3 4 5 6
3741
1 2 3 4 5 -6
-2259
^C
```

このプログラムの採点では、提出されたファイルから関数 `addcoin()` を抽出し、採点用のソースファイルの実行結果によって評価されます。

### 問8 `exam15-8.c` (10点)

関数 `tdiff()` は、3つの `double` 型引数を受け取ると、それらの値の最大値と最小値の差の絶対値を `double` 型の返り値として返す関数である。

このような関数 `tdiff()` を定義し、`tdiff()` を利用して入力された3つの実数の値の範囲の幅を出力するプログラム `exam15-8.c` を作成し、kiso2 コマンドを用いて提出しなさい。

以下の実行例では、`main()` 関数によって実数値の入力と出力が繰り返し行われているが、提出するプログラムでは必ずしもこれらを実現する必要はない。

実行例：

```
t180900@s01cd0542-160:~/kiso2-2018/exam15$ ./exam15-8
1 1 1.0
0.000000
0.0 1.1 2.2
2.200000
9.8 7.6 5.4
4.400000
1 7 1
6.000000
-2 -5.1 -2
3.100000
^C
```

このプログラムの採点では、提出されたファイルから関数 `tdiff()` を抽出し、採点用のソースファイルの実行結果によって評価されます。

**問9 exam15-9.txt (5点)**

この試験の間0から問8までで自身が得られる合計点(0~95点)を自己採点しなさい。自己採点した合計点のみを半角数字で入力したテキストファイル **exam15-9.txt** を作成し、kiso2 コマンドを用いて提出しなさい。

実際に得られた得点を  $x$ 、自己採点で予測した得点を  $y$  とすると、問8は  $\frac{10-|x-y|}{2}$  (小数点以下切り上げ)で計算されます。ただし、計算された点数が 0未満の場合、問9は 0点として採点されま

From:

<https://www-slab.math.ryukoku.ac.jp/> - **www-slab.math**

Permanent link:

<https://www-slab.math.ryukoku.ac.jp/lecture/kiso2/exam15-2018> Last update: **2019/01/18 16:09**