

情報理工学演習Ⅲ 演習課題（第1回）

テーマ : 最大公約数

実施日 : 2020年4月16日（木）

担当 : 有村 博紀, 堀山 貴史, 金森 憲太朗 (TF), 加井 丈志 (TA)

授業連絡先 : enshuiii-csit@ist.hokudai.ac.jp (演習課題の提出・質問など)

担当連絡先 : kanamori@ist.hokudai.ac.jp (金森), kai@ist.hokudai.ac.jp (加井)

以下の各問に関して、それぞれプログラムを実装せよ。実装したコードは必ず、以下に示す提出様式を守って、**本授業のメールアドレス (enshuiii-csit@ist.hokudai.ac.jp) 宛にメールで提出すること。**

- メールの件名 : 「[csit ex3]xxxxxxxx」, ここで xxxxxxxx は自分の学生番号。
- 添付ファイル名 : 第 n 回演習の間 m に対するコードのファイル名は, 「n_m.c」とする。

- メール宛先 : enshuiii-csit@ist.hokudai.ac.jp
- メール件名 : [csit ex3]02180000
- 添付ファイル : 1_1.c, 1_2.c, 1_3.c, 1_4.c

例：学生番号02180000の学生の場合

メール本文には、学生番号、所属コース、名前、実施回（第1回演習）、解いた問題番号を記載すること。課題提出によって出席をとるため、解いている途中の状態でも構わないので、演習時間内に一度は提出すること。

実装にはC言語を用い、可能ならば gcc (ver.4.8.5 以降) でコンパイルすること。

```
% gcc 1_1.c // コンパイル
% ./a.out // 実行
```

コンパイル・実行例

演習時間内に解き終わらなかった場合は、次回の演習時間までに提出すること。期間中であれば、何度でも再提出してよい。提出状況等は本授業のホームページまたはELMSグループ (moodle) 上に掲示予定なので、適宜確認すること。

問題番号に * が付されたものは、発展課題である。解答は必須ではないが、解ければ成績評価に加点されるので、是非挑戦してほしい。

【問1】 以下の Sample 1 は、標準入力から2つの非負整数を受け取り、それらの最大公約数を求めるプログラムである。これを 1_1.c として実装せよ。

```
1: #include <stdio.h>
2:
3: int main(){
4:     int a, b, ans = 0, i;
5:     scanf("%d %d", &a, &b);
6:     if (a < b) {
7:         int tmp = a;
8:         a = b;
9:         b = tmp;
10:    }
```

Sample 1

```

11:  for (i = 1; i <= b; i++) {
12:      if (a % i == 0 && b % i == 0) {ans = i;}
13:  }
14:  printf("%d\n", ans);
15:  return 0;
16: }

```

実装したプログラム 1_1.c に、標準入力から 24 と 36 を入力して実行し、結果を確認せよ。

（注）Sample 1 は愚直なアルゴリズムの実装であるので、値の大きな非負整数を入力すると、実行には時間がかかる。なお、計算機室の環境であれば、実行時間は `time` コマンドで以下のように計測できる。

```
% echo 1431655764 2147483646 | time ./a.out
```

実行例

もちろん、`time ./a.out` として実行し、あとから2つの数値を入力してもプログラムは実行可能である。しかしこの場合は、`time` の表示する実行時間に入力待ち中の時間も含まれてしまうため、上記実行例ではパイプライン処理を用いて実行している。

【問2】 2つの非負整数 n と m を受け取り、これらの最大公約数を $O(\min(n, m))$ 時間より高速に計算し、結果を出力するプログラムを 1_2.c として実装せよ（ヒント：授業で説明されたアルゴリズムを思い出そう）。ただし、`%` 演算は定数時間で実行可能とする。

入力は以下の形式で標準入力から与えられるものとする。出力の末尾には改行を入れること。

n m

入力形式

実装したプログラム 1_2.c に、標準入力から 24 と 36 を入力して実行し、結果を確認せよ。

【問3*】 平面上の2つの格子点 $P = (x_P, y_P)$ と $Q = (x_Q, y_Q)$ を受け取り、線分 PQ 上に存在する格子点の個数を出力するプログラムを 1_3.c として実装せよ。

入力は以下の形式で標準入力から与えられるものとする。出力の末尾には改行を入れること。

x_P y_P x_Q y_Q

入力形式

実装したプログラム 1_3.c に $P = (1, 11)$ と $Q = (5, 3)$ を入力して実行し、結果を確認せよ。

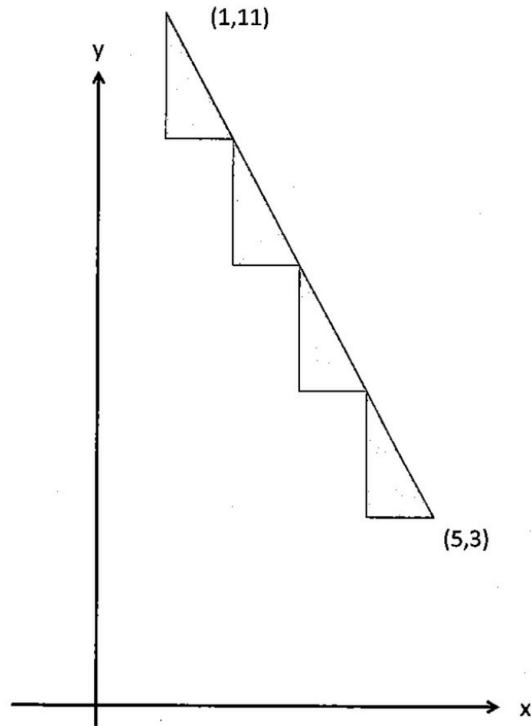


図 1 $P = (1, 11)$ と $Q = (5, 3)$ のとき、格子点は5つあり、それぞれの座標は x 座標が小さいものから順に $(1, 11)$, $(2, 9)$, $(3, 7)$, $(4, 5)$, $(5, 3)$ である。

【問4*】 平面上の2つの格子点 $P = (x_P, y_P)$ と $Q = (x_Q, y_Q)$ を受け取り、線分 PQ 上に存在する格子点のすべての座標を出力するプログラムを 1_4.c として実装せよ。
 入力は以下の形式で標準入力から与えられるものとする。出力は x 座標が小さい格子点から順に行い、各行に格子点の x 座標と y 座標を半角スペース区切りで表示すること。また、出力の末尾には改行を入れること。

$x_P \ y_P \ x_Q \ y_Q$	入力形式
-------------------------	------

$x_1 \ y_1$ $x_2 \ y_2$ $x_3 \ y_3$... // 但し, $x_1 < x_2 < x_3 < \dots$	出力形式
--	------

実装したプログラム 1_4.c に、標準入力から $P = (1, 11)$ と $Q = (5, 3)$ を入力して実行し、結果を確認せよ。