課題　ガウス素数

ガウス素数を求めて表示するプログラムを作成せよ．プログラムは以下の仕様を満たすようにすること．

1. `get_command_argument` などを用いてコマンドライン引数として正の整数 n を受け取る．
2. 実部および虚部が $-n \leq i \leq +n$, $-n \leq j \leq +n$ の範囲のガウス整数について，ガウス素数を判定する．
3. ガウス素数の位置には 1 を，そうでない位置には 0 を出力する．

実行例は例えば以下の通り：

```
$ ./a.out 5
 0 1 0 1 0 0 0 1 0 1 0
 1 0 0 0 1 0 1 0 0 0 1
 0 0 0 1 0 1 0 1 0 0 0
 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 1 0 1 1 0 1 0
 0 0 1 0 0 0 0 0 1 0 0
 0 1 0 1 1 0 1 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1
 0 0 0 1 0 1 0 1 0 0 0
 1 0 0 0 1 0 1 0 0 0 1
 0 1 0 1 0 0 0 1 0 1 0

$ ./a.out 10
 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0
 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1
 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0
 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1
 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
 0 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0
 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0
 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1
 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0
 1 0 0 0 1 0 1 0 1 1 0 1 1 0 1 0 1 0 0 0 1
 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0
 1 0 0 0 1 0 1 0 1 1 0 1 1 0 1 0 1 0 0 0 1
 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0
 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1
 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0
 0 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0
 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1
 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0
 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1
 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0
```

gnuplot を用いて以下のように表示すると美しい幾何学模様が得られる．

```
$ ./a.out 20 > data.data
$ gnuplot
gnuplot> set palette gray negative
gnuplot> plot 'data.dat' matrix with image
```

ガウス整数とガウス素数の定義

以下では実部が a，虚部が b の複素数を (a, b) と表すことにする．このときガウス整数とは，整数 i, j に対して，(i, j) の形で表される複素数のことである．
また，ガウス素数 (i, j) を，自明な約数 $(\pm 1, 0)$, $(0, \pm 1)$, $(\pm i, \pm j)$, $(\mp j, \pm i)$ 以外の約数を持たないガウス整数として定義する．

アルゴリズム

ガウス素数の判定には，以下のエラトステネスのふるいの複素数への拡張版を用いよう．ただし，$0 \leq i \leq n$, $0 \leq j \leq n$ の範囲のガウス整数を扱い，表示の際には i および j の符号反転に対する対称を用いてよい．

1. 2 次元の論理型配列 `is_prime` を用意する．この配列の要素 `is_prime(i,j)` は，ガウス整数 (i, j) がガウス素数であるかどうかを示すフラグである．
2. 初期には $0 \leq i \leq n$, $0 \leq j \leq n$ の範囲のガウス整数を全て素数であると仮定し，対応する位置に `.true.` を設定する．ただし，$(0, 0), (1, 0), (0, 1)$ はガウス素数ではないので，`is_prime(0,0)`, `is_prime(1,0)`, `is_prime(0,1)` を `.false.` に設定する．
3. 配列を順に走査する．走査中の位置 (i, j) に対応するガウス整数 (i, j) について，`is_prime(i,j)` が `.true.` であれば，それはガウス素数の候補である．ガウス素数の候補に対して，以下の処理を行う．
   - ▷ ガウス整数の乗算が $(i, j) \times (l, m) = (il - jm, im + jl)$ で定義されることを用いて，ガウス素数の候補 (i, j) に対して，その倍数を素数の候補から取り除けばよい．具体的には，$l = 1, 2, \ldots$ のそれぞれに対して，$(il, jl), (il \mp j, jl \pm i), (il \mp 2j, jl \pm 2i), \ldots$ に対応する位置の `is_prime` を `.false.` に設定する．また，対称性から `is_prime(i,j) = is_prime(j,i)` となることを用いてよい．
   - ▷ ただし，$l = 1, m = 0$ に対応する，`is_prime(i, j)` は `.true.` に設定する（元に戻す）．

これらが終了した後に最終的に，`is_prime` が `.true.` になっている位置に対応するガウス整数がガウス素数となる．