

プログラミング演習I (第10回) 課題

• 基本①: basic_TwinPrime

- 双子素数とは、2つの素数の差が2である素数のペアのことである。
- まず、1つの整数の引数を取り、その値が素数かどうかを判定し、素数ならtrue、素数でないならfalseを返すisPrime関数を作成せよ。
- 次に、isPrime関数を利用し、1000までの双子素数を下記のようにすべて標準出力するとともに、その総数を出力するプログラムを作成せよ。
- 出力においてペアとなる素数は丸かっこ内にカンマ区切りで書くようにせよ
- (参考)
<https://ja.wikipedia.org/wiki/双子素数>

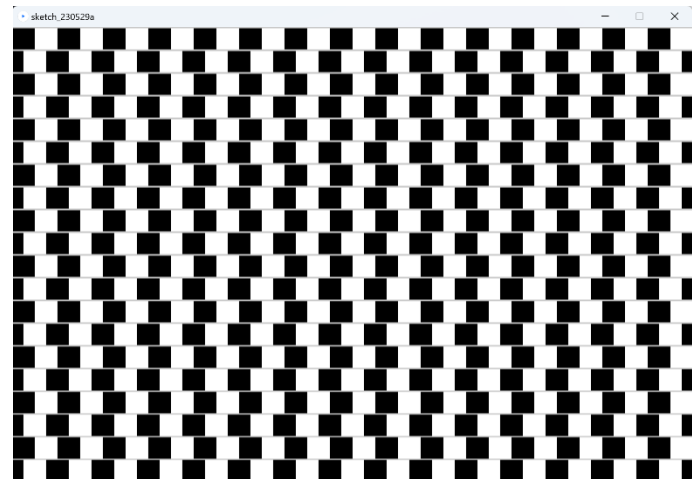
```
(3, 5)
(5, 7)
(11, 13)
(17, 19)
(29, 31)
(41, 43)
(59, 61)
(71, 73)
(101, 103)
(107, 109)
(137, 139)
(149, 151)
(179, 181)
(191, 193)
(197, 199)
(227, 229)
:
:
```

1000までの双子素数の数は**個です

プログラミング演習I (第10回) 課題

• 基本課題② basic_Cafewall

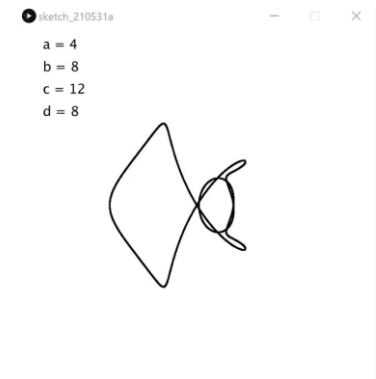
- 900x600のウィンドウを作成し、その上に縦横30ピクセルの黒と白の四角形を奇数行ではそのまま、偶数行では15ピクセル左にずらしながら表示するプログラムを作成せよ
- また、各行の間に灰色の太さ2の灰色の横線を入れるようにせよ。
- これにより下図と同じ「カフェウォール錯視」を表示するプログラムを作ってください。
- 錯視が起こってるかどうか自信がない場合は隣の人に確認してもらい提出せよ。
- 下端の灰色線は見えなくてもよいが、上端の灰色線は見えるようにせよ



プログラミング演習I (第10回) 課題

• 基本課題③ スケッチ名：**basic_Curve**

- 800x800のウィンドウを作成せよ
- ある a , b , c , d について、 t を0から0.001ずつ 2π まで増やしていきつつ x と y の値を求め、その座標に直径3ピクセルの円を描画する`drawCurve()`を作成せよ。
 - $x = 100(\cos(at) - (\cos(bt))^3) + 400$
 - $y = 100(\sin(ct) - (\sin(dt))^3) + 400$
- なお、`drawCurve`関数の引数は上記の式の a , b , c , d の4つの整数とするか、引数なしとせよ。
- また、画面をクリックするたびに a , b , c , d の値を1~20の間でランダムに決定し、`drawCurve`関数を用いて描画せよ
- a , b , c , d それぞれの値も表示せよ
- ヒント：`draw()`は空で`mousePressed()`で描画すると楽



プログラミング演習I (第10回) 課題

• 発展②スケッチ名 : advanced_CalcPI

- フーリエ級数展開を用いると, 下記の式を用いて円周率の近似値を求めることが可能である

$$\sum_{k=0}^N \frac{(-1)^k}{2k+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \frac{\pi}{4}$$

- このNをいくつまで指定するかというのを引数とし, 上記の式に4を掛けた値を返り値として, double型で返す関数 `calcPI` を作成せよ

- 関数の定義 : `double calcPI(int N);`

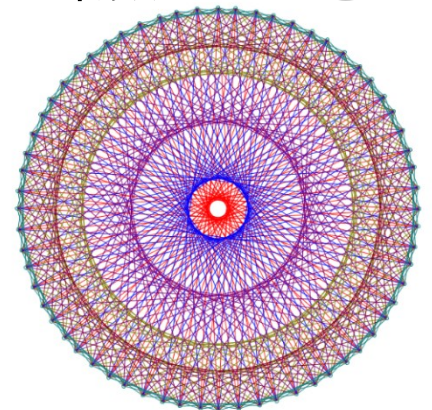
- また, `calcPI` の引数を10, 100, 1000, 10000, 100000, 1000000, 10000000としていった時の結果 (円周率の近似値) を標準出力せよ

```
N= 10 3.2323157489299774
N= 100 3.151493337005377
N= 1000 3.1
N= 10000 3.1
N= 100000 3.1
N= 1000000 3.1
N=10000000 3.1
```

プログラミング演習I (第10回) 課題

• 発展課題② スケッチ名 : `advanced_Mandala`

- 800x800のウィンドウ上の中心から300ピクセル離れたところに, 等間隔に64個の直径5ピクセルの円を描く
 - 半径300ピクセルの円周上に等間隔に小さな円を描く
- 64個のすべての点から, 素数個分だけ時計回り or 反時計回りに離れた点に線を描け (5, 13, 19, 31など)
 - 例えば素数として「5」を選択した場合は, 1個目の点から6個目の点, 2個目の点から7個目の点, 2個目の点から7個目の点, ..., 63個目の点から68個目の点 (実際は $68\%64=4$ 個目の点), 64個目の点から69個目の点 (実際は $69\%64=5$ 個目の点) をそれぞれつなぐような線を描く (64の剰余を使うのがポイント)
- 上記の処理について, 5つ以上の素数を使って線のパターンを描き下記のような糸掛け曼荼羅を作成せよ. なお設定する素数ごとに色を変更することで見え方を工夫せよ



ヒント

- 64個の点の座標を0から63までの変数を使って表現しよう！
- 360度の64等分なので、5.625度ずつ離れている計算になる
 - 例えば素数として31を選定した場合は、 i 番目と $i+31$ 番目が対象となる
 - つまり $5.625 * i$ と $5.625 * (i+31)$ になる
- 度を使う場合は、radiansを使うとよいよ！