

■ 文法的落ち穂拾い：関数における引数の型

前回のサンプルプログラムの中では、y 軸座標位置を計算する際に整数変数 x と実数変数 a, b との型が異なる演算が行われています。

また、関数の引数（カッコの中に与えるパラメタ、sin(x) であれば x の部分に相当）にも型があります。また、戻り値にも型があり、プログラマはこれらを意識して使用すべきです。つまり絶対値を求める関数のひとつ abs()は引数、戻り値ともに整数型です。このため、実数の絶対値を求めることはできません。実数向けには float 型に対応する fabsf()、double 型に対応する fabs()が用意されています。

右(上)のサンプルでは暗黙の型変換が働くことを前提に、特に float 型にキャストせず書いています。

```
int x, win;
float y, a, b;
.... (略) ....
for(x=-190; x<190; x++) {
    y=a * x + b;
    pset(win, x + 200.0, y + 200.0);
}
```

型の相違について、暗黙の型変換で済ませるか明記するかは、プログラマに任せられています。ただ、右(下)の例ほど厳密に書いてある場面には余り遭遇しません。

```
y=a * (float)x + b;
pset(win, (float)x + 200.0f, y + 200.0f);
```

特にこの教材では、座標位置などに float 型の引数を期待している EGGX の各種グラフィック関数に、double 型の実数をそのまま与えている点に注意してください。(sin() 関数は double 型向けで、float 型向けには sinf() がありますが、わざわざ sinf()関数を使うことはしていません。)

つまり、余り厳密に型を合わせて書かなくても良いが、問題が生じたときに型違いの可能性を疑える程度には意識しておくべきだ、と考えて下さい。

□ double 型変数への scanf

EGGX の関数などで説明したように、関数の引数が実数であった場合に、float と double の型合わせを厳密に守る必要はありません。しかし scanf() では事情が異なります。

scanf() 関数で実数型に値を入力する場合、float 型変数であれば %f、double 型であれば %lf を変換文字として指定します。(つまり明確に型を合わせなければなりません)

```
float f; double d;
scanf("%f %lf", &f, &d);
```

教科書 p.122 の表 7.3 及び脚註には printf と scanf で変換文字は同様に機能するように読めますが(p.121 最終パラグラフも)、そうではありません。

scanf では float は %f、double は %lf でなければなりません。

printf では float, double 共に %f, %lf が同様に機能します。(どちらでも構いません)

scanf でだけ厳密に合わせる必要があるのは、まだ学んでいない & つまり「ポインタ」の構造・機能に起因するものです。このクラスでは細かく説明しませんので、ポインタについて学ぶまでは「double の scanf には %lf」と丸覚えしてください。

□ 定数

`math.h` をインクルードすることによって、幾つかの数学関数で利用できる定数が定義されます。例えば円周率は下記のように定義されています。

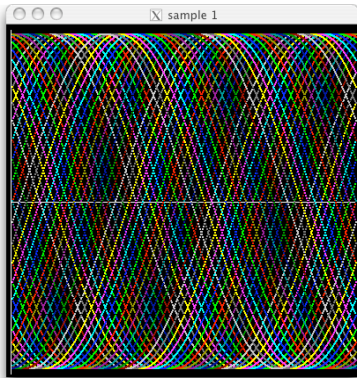
```
#define M_PI 3.14159265358979323846264338327950288
```

(興味のある受講生は `/usr/include/architecture/i386/math.h` ファイルを参照すると良いでしょう)

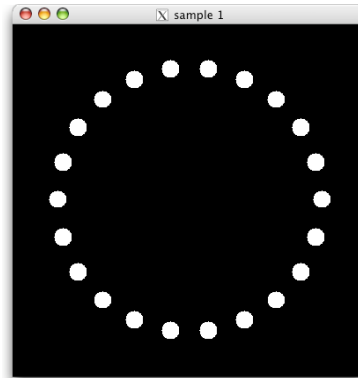
□ 課題 1.

三角関数など数学関数を利用して、何か美しい図形を描いて下さい。

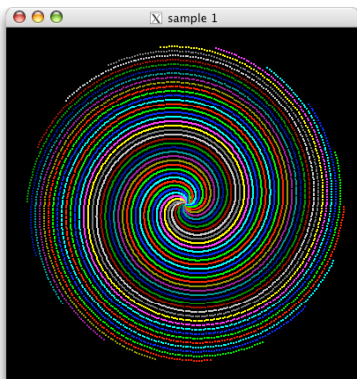
例えば下記のようなものなど。各自工夫して、きれいなものを作って下さい。(できれば二種類)



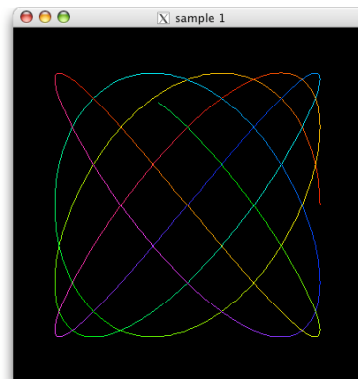
sin カーブを少しずつずらしながら色を変えて描いたもの



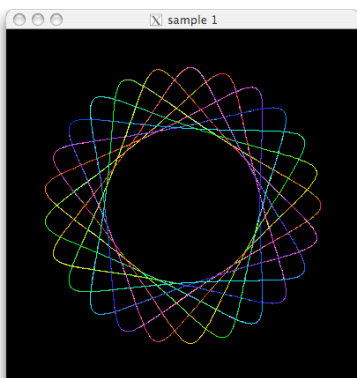
円周上に中心位置をずらしながら図形を描いたもの



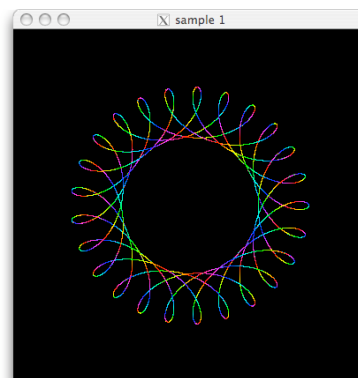
スパイラル



リサージュ曲線 (描いている途中)
徐々に色を変えてアニメーションとしてみた



スピログラフその1



スピログラフその2