メディアプログラミング演習―第２回（第１テーマ２日目）―

演習２－０：復習(改修)

カラー画像を白黒写真（グレースケール画像）に変換プログラム

前回の演習１－２で完成したプログラムにおいて、グレースケール処理を関数化する

（この関数は、関数 draw の次に置く）

```
float  gray(int x, int y){
int pos = x + img_in.width*y;
color c = img_in.pixels[pos];
float r = red( c );  float g = green( c );    float b = blue( c );
return( 0.3 * r + 0.59 * g + 0.11 * b);
}
```

処理内容（「２重ループの中側」、以下同様）の部分を以下の通りとし、256 階調のグレースケールに変換する。

```
float gw=gray(x,y);
int pos = x + y*img_in.width;
img_out.pixels[pos] = color(gw,gw,gw);
```

演習２－１：グレースケール変換（確認）

前回配布した画像を用い，前述（「復習」）の方法で処理・表示しなさい。用いる画像は sample1-1,sample1-2 のいずれでも良いが、以下の各演習では同じ画像を用いる。

演習２－２：階調変換

(a)処理内容の部分を

```
float gw=gray(x,y);
 float gww=int(gw/64)*64;  <-(*)
 int pos = x + y*img_in.width;
 img_out.pixels[pos] = color(gww,gww,gww);
```

とし，実行しなさい．(*)行の働きを理解しなさい．

上記プログラム２行目で「64」では，256 階調が 4 階調となる．

(b) (*)行の定数２か所を「32」とし実行しなさい。どのような処理かを考えなさい。

演習２－３：画像反転処理

処理内容の部分を

```
float gw=gray(x,y);
 gw=255-gw;  <- (*)
 int pos = x + y*img_in.width;
 img_out.pixels[pos] = color(gw,gw,gw);
```

とし実行しなさい．(*)行の働きを理解しなさい．

演習２－４：ミラー変換

処理内容の部分を

```
int pos1 = x + y*img_in.width;
int pos2 = (img_in.width-x-1) + y*img_in.width; <-(*)
img_out.pixels[pos2] = img_in.pixels[pos1];
```

とし実行しなさい．(*)行の働きを理解しなさい．

演習２－５：モザイク処理

(a) 処理内容の部分(および下記下線部)を，

```
int d=8;
for ( int y = 0; y < img_in.height-d; y+=d)
{
  for ( int x = 0; x < img_in.width-d; x+=d)
  {
   int pos1 = (x+d/2) + (y+d/2)*img_in.width;
   color c = img_in.pixels[pos1];
   for(int xd=0;xd<d;xd++){
     for(int yd=0;yd<d;yd++){
      int pos2 = (x+xd) + (y+yd)*img_in.width;
      img_out.pixels[pos2] = c ;
     }
   }
  }
}
```

とし、実行しなさい。

(b) d=16 として、実行しなさい。