練習問題の答え

変数の型と初期値

```
unsigned  char A =  3; (16進  0x03, 2進          )
unsigned  char B = 33; (16進  0x21, 2進          )
char           C = -2; (16進  0xFE, 2進          )
char           D =  0;
```

式	結果
D = A & B	A: B: D:
D = A | B	A: B: D:

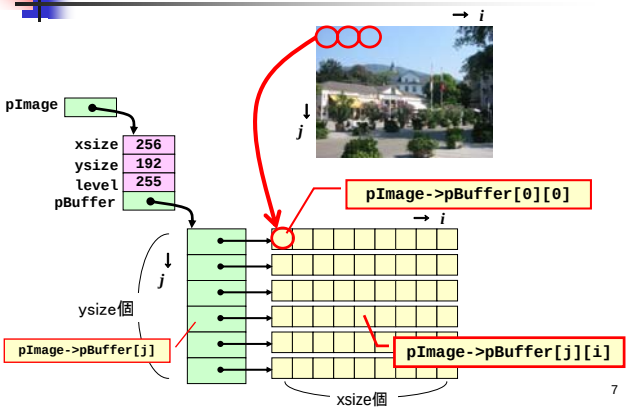
練習問題の答え

式	結果
D = A ^ B	A: B: D:
D = B << 2	B: D:
D = A >> 3	A: D:

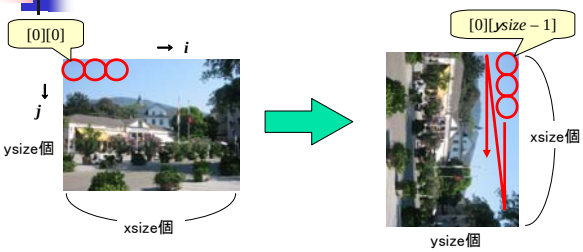
練習問題の答え

式	結果
D = C >> 3	C: D:
D = ~A	A: D:
A >>= 2	A: ←実行前 A: ←実行後

画像バッファの様子



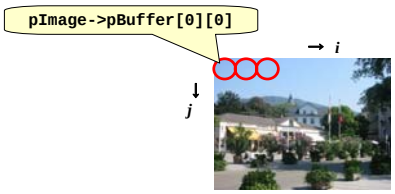
画像の90度回転　ipRotateImage



関数内で作業用の画像バッファメモリを確保し，
順に画素を埋めていけばよい．

iioMallocImageBuffer
iioFreeImageBuffer

画素のビットマスク ipBitMask



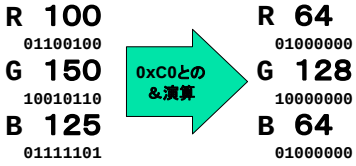
1画素ずつ処理していく
1画素はR,G,Bから成る構造体で表現している
→　R,G,Bのそれぞれに対して処理を行う

画素のビットマスク ipBitMask

pImage->pBuffer[j][i].r &= 0xC0;
pImage->pBuffer[j][i].g &= 0xC0;
pImage->pBuffer[j][i].b &= 0xC0;

2進表現で
1100 0000

下位6ビットが0なので
ビットごとのAND演算
をすると，下位6ビット
が0にマスクされる．



画素のビットマスク ipBitMask

pImage->pBuffer[j][i].r &= 0xC0;
pImage->pBuffer[j][i].g &= 0xC0;
pImage->pBuffer[j][i].b &= 0xC0;

サンプルプログラム3：
各画素の下位”6”ビットを0でマスク→0xC0との＆演算

演習課題：
各画素の下位”4”ビットを0でマスク →何を使えばよいか？