最終レポートについて

課題4の最終レポート

- ユーザが指定した方法で入力画像を処理し，結果を別ファイルとして出力するプログラムを作成
  - 減色処理と左右反転処理は必須
  - 他の画像処理はオプション

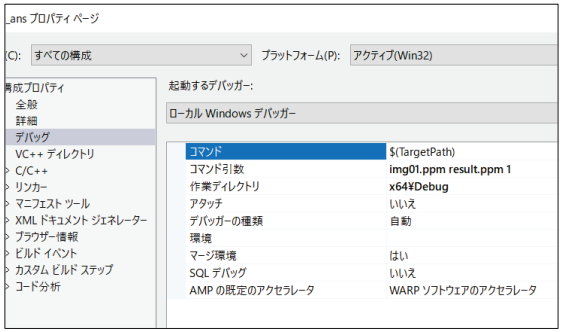
コマンドライン引数について

- ユーザのコマンドライン引数指定により，入力ファイル名，出力ファイル名，画像処理を選択できるようにすること
  - ex4.exe [入力ファイル名] [出力ファイル名] [モード]
- モードの仕様
  - 0：減色処理，1：左右反転，それ以外：コピー
  - オプション課題を実装したら2, 3, 4…で指定する
- 減色処理における下位nビットマスクは，n = (自身の学籍番号の下一桁 % 7) + 1 とする

コマンドライン引数の指定例

- コマンドラインの場合

- Visual Studioのコマンド引数の場合



実装の方針(1)

- main.cで，コマンドライン引数によってユーザが画像処理を選べるように改良する
  - main.cでのコマンドライン引数は”文字列”なので注意

```
int main(int argc, char* argv[])
{
  ...
  int mode;
  mode ← コマンドライン引数によるmode指定;

  ... /* 画像のロード */

  /* modeが0であれば減色処理 */
  /* modeが1であれば左右反転 */
  /* それ以外であればコピー */

  ... /* 画像のセーブ */
```

※文字列から数値への変換はatoi関数が便利．
[例]
```
int b;
b = atoi(″3″);
```
とすると，bには数値として3が代入される．

実装の方針(2)

- img_proc内，ipLRMirror関数を新たに作成
  - 左右を反転する方法は各自で考えること

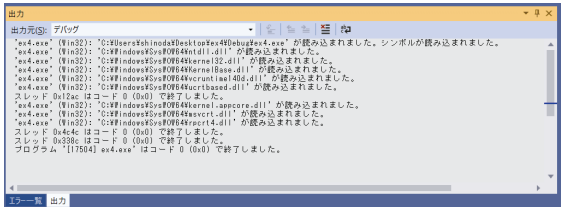
```
void ipLRMirror(IMAGE *p_in_image, IMAGE* p_out_image)
{
  int i, j;

  ...

}
```


レポートに載せる実行結果について

- 全てのモードの出力画像をのせ，結果が正しいか各画像ごとに説明し，必要に応じて考察をする
- デバッグ実行後の出力ウィンドウ(下記)を先頭行から全てのせ，メモリリークがないかを説明
  - モードに1を指定したときのみでかまわない
  - 確認ができればVisual Studio以外でもよい



最終レポートについての補足

- ソースコードと実行結果とも全員異なる内容になるのでレポートのコピペは不可能
  - 他人のソースコードやレポートをコピーした場合，オリジナルの作成者を含め単位は保証しない
  - これが原因で毎年数名が不可となっている
- 画像処理の実装種類数が多いほど評価UP
  - 例年は8割以上の学生がオプション課題を実装し，中には10種類以上の処理を実装する人も数名

レポートのテンプレートファイル

- レポート作成にあたっては，講義webページにあるレポートテンプレートファイルをダウンロードして利用すること
- 提出時はPDFに変換すること

終わった人は

- 作業が終わったら，レポートに備えてUSBメモリ等にプログラムをコピーする
- 最終週も出席を取るため，全員ネットワーク実験室に集まること