補足

より良い例外処理

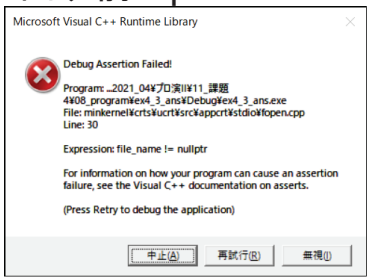
- ユーザが不適切な方法でプログラムを実行した場合でも安全に終了させるための工夫
  - argcをチェックし，こちらが意図した分だけコマンドライン引数が与えられているか
  - 入力/出力ファイルは正しくopenできているか
  - モードの書式は正しいか
- 不適切な場合，エラーメッセージを出力するとともにmainで1をreturnして終了するとよい

簡単なデバッグテクニック

- 怪しい処理行の前にブレークポイントを置き，ステップ実行（1行ずつ実行）
  - どの処理に分岐するか一目でわかる
  - 止まらない場合は，処理がそこまで到達していない
- （全環境共通）適当なprintf文を大量に設置する
  - コマンドプロンプトに表示される文字から，どこまで処理されたか一目でわかる
  - かなりいいかげんな方法であり，ステップ実行が難しい環境などで有用

よくある実行時エラー

- エラー文を読むことで解決することが多い
- よくあるケース
  - コマンドライン引数を与えずに実行
  - 動的確保/解放のコードに間違い
  - iioMallocImageBuffer関数を呼び出す前にpBufferに画素値を書き込み/読み込み
  - 画像処理関数において，配列の範囲外にアクセス



画像処理テクニックの例（1）

- 回転，反転
  - コピー先のインデクスを変える（回転時はサイズ注意）
- モノクロ化
  - 各画素でRGB値を同じ値にする
- 拡大/縮小
  - 拡大：ある画素の値を複数の画素にコピー
  - 縮小：特定の位置の画素はコピーし，他は切り捨て
- モザイク化
  - 出力画像のn×n画素を入力画像の同範囲の平均値で置き換え

画像処理テクニックの例（2）

- 特定の色の物体を検出
  - RGB値がある範囲内の画素値はそのまま，他は0で置き換える
- ノイズ付加
  - ある確率分布に従う乱数を生成し，画素値に付加
- ノイズ除去
  - 付加されたノイズによって対処方法が異なる
  - 平均値フィルタやメディアンフィルタなど
- ぼかし処理，輪郭強調，エッジ検出
  - 講義資料webページの補足説明を参照

バイナリエディタ（1/2）

- 画像ファイルやテキストファイルのバイナリ情報が閲覧できる
- これを利用すれば減色処理の正否がわかる
- Visual Studioの場合，ファイル→開く→ファイル．．．→対象のファイルを選択する→「開く」の下矢印→「ファイルを開くアプリケーションを選択」→「バイナリエディター」を選択

バイナリエディタ（2/2）

- バイナリデータと位置は16進数表記
- 右側にASCIIコードに従ったテキストも表示