

2021

計算機プログラミング演習

第11回 ファイルの入出力
バイナリファイル

本日の演習プログラム

教科書 p.358 のプログラム(2)

```
/*    bfile_test1.c    */
#define    _CRT_SECURE_NO_WARNINGS    1
#include    <stdio.h>

int    main( void ) {

    FILE *file;
    file = fopen("test.dat", "wb" );
    if (file == NULL) {
        printf("ファイルをオープンできませんでした。¥n");
        return 1;
    }
    fclose( file ) ;
    return 0 ;
}
```

第11回の目標

- ファイルの入出力(バイナリファイル)
 - `fopen()` ファイルを「開く」
 - `fclose()` ファイルを「閉じる」
 - `fwrite()` データ全体を書き出す
 - `fread()` データ全体を読み込む
- 「データ全体」
 - 配列や構造体などのデータの塊をすべて

バイナリーファイルとは

- 変数の内容を(文字に変換せずに)直接ファイルに書き込む方法
- 整数(int) 4 byte, 倍精度実数(double) 8 byte
でファイルに格納(区切り記号、改行なし)
- 配列や構造体の保存が容易
- データの互換性に難

fwrite(), fread() 関数

- fwrite()

`fwrite(変数ポインタ, 変数サイズ, 記録する個数, ファイルポインタ)`

- fread()

`fread(変数ポインタ, 変数サイズ, 記録する個数, ファイルポインタ)`

バイナリーファイルの書き出し

教科書 p.358 のプログラム(1)

```
/*    bfile_test1.c    */

#define    _CRT_SECURE_NO_WARNINGS    1
#include    <stdio.h>

int    main( void ) {

    FILE *file;
    file = fopen("test.dat", "wb" );
    fclose( file ) ;
    return 0 ;

}
```

バイナリーファイルの書き出し

教科書 p.358 のプログラム(2)

```
/*    bfile_test1.c            */
#define  _CRT_SECURE_NO_WARNINGS  1
#include  <stdio.h>

int  main( void ) {

    FILE *file;
    file = fopen("test.dat", "wb" );
    if (file == NULL) {
        printf("ファイルをオープンできませんでした。¥n");
        return 1;
    }
    fclose( file ) ;
    return 0 ;
}
```

教科書 p.360 のプログラム

```
/*      bfile_test1.c      */
#define _CRT_SECURE_NO_WARNINGS 1
#include <stdio.h>

int main( void ) {
    int    buf[] = { 10, 100, 1000, 10000 } ;
    FILE   *file;
    file = fopen("test.dat", "wb" );
    if (file == NULL) {
        printf("ファイルをオープンできませんでした。¥n");
        return 1;
    }
    fwrite( buf, sizeof(buf), 1, file ) ;

    fclose( file ) ;
    return 0 ;
}
```

sizeof() 関数: 変数や変数型のサイズを答える

fwrite() でのデータの書き方

- 変数を一つずつ書き込む (非現実的?)

```
for ( i = 0 ; i < 4 ; i++ ) {  
    fwrite( &buf[i], sizeof(int), 1, fp );  
}
```

& を付けてポインタにする

- 配列を一度に書き込む

```
fwrite( buf, sizeof(int), 4, fp );
```

整数を 4 個出力 (ループ不要)

```
fwrite( buf, sizeof(buf), 1, fp );
```

配列(buf)を1個出力 (ループ不要)

【例題】

WebClassで「v-flip.c」というソースプログラム（未完成）と baboon.bmp という画像ファイルを配布している。

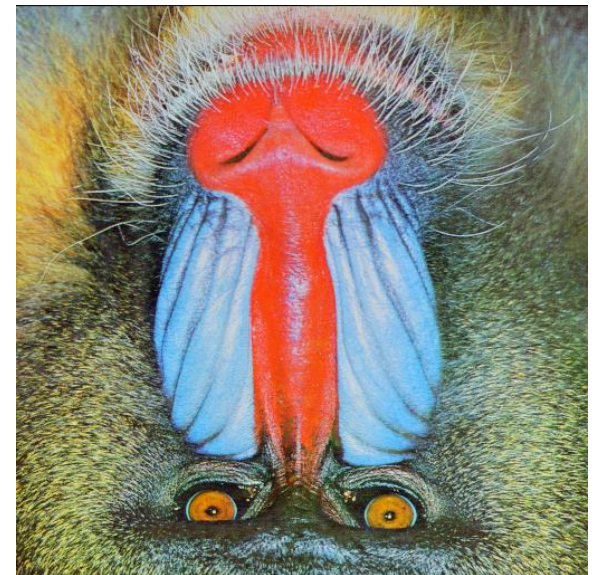
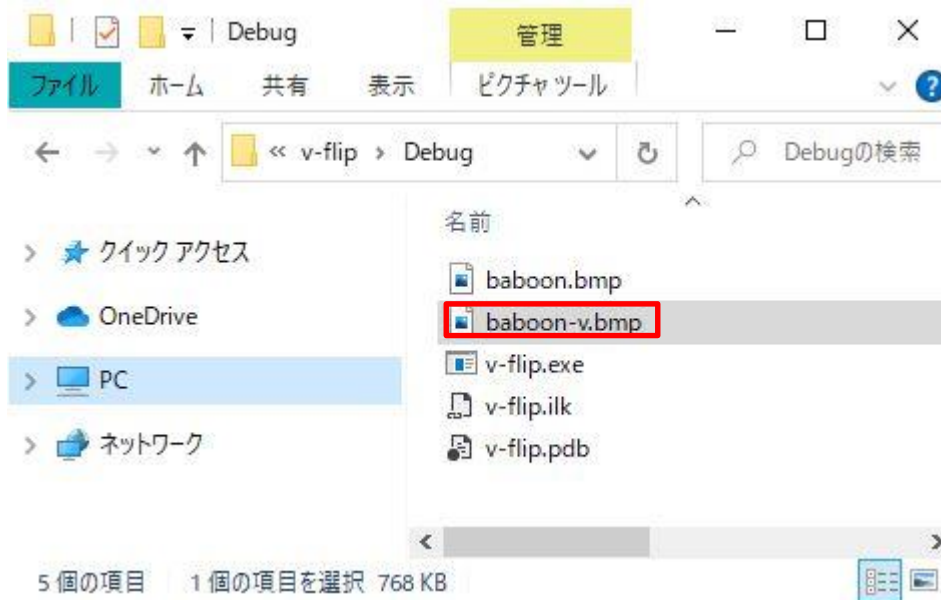
ソリューション名：	v-flip
プロジェクト名：	v-flip
ソースプログラム：	v-flip.c

v-flip.c ではbaboon.bmp ファイルを buf1 配列に読み込み、buf2 配列に上下反転した画像データを作成している。

プログラム中の buf2 配列を 「baboon-v.bmp」というバイナリーファイルに保存する部分を書き加えよ。

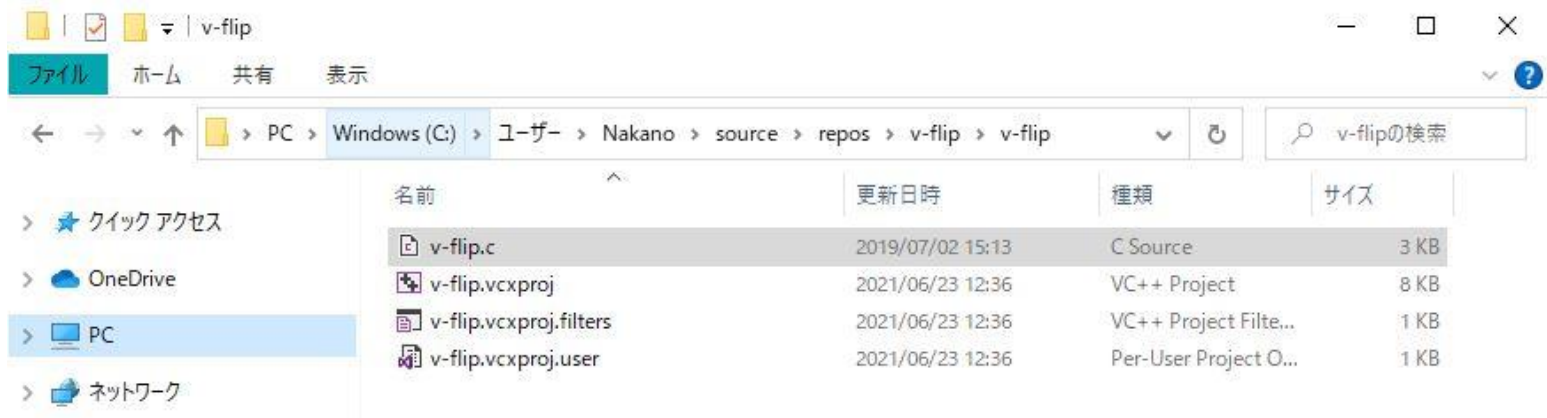
【例題】画像の出力と出力結果

プログラムが完成してビルド及び実行をすると、入力画像 (baboon.bmp) が上下反転した画像が出力される (baboon-v.bmp)



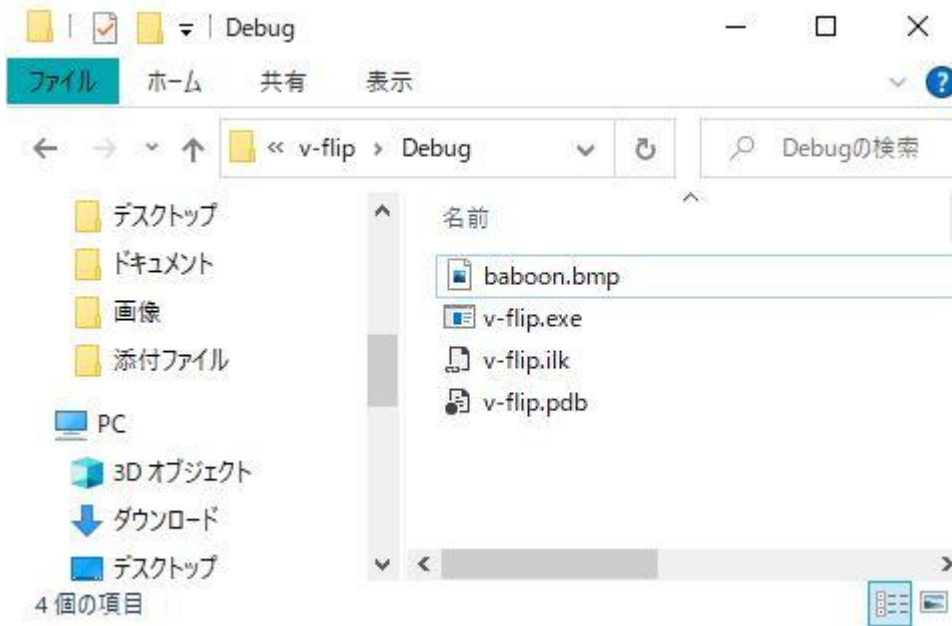
ソースプログラムの導入

- すでにあるソースプログラムを visual studio に読み込むには
 - 「新しいプロジェクト...」→「空のプロジェクト」(プロジェクト名を指定)
 - ファイルエクスプローラで Projects¥ソリューション名 ¥プロジェクト名 ディレクトリにソースプログラムをコピー
 - 画面右の「ソリューション エクスプローラー」内のソースファイルのフォルダを右クリック
 - 「追加(D)」→「既存の項目(G)」を選択
 - ソースプログラムを選択



入力画像の配置

入力画像(baboon.bmp)を配置する



ビルドする(ただし警告が表示される)

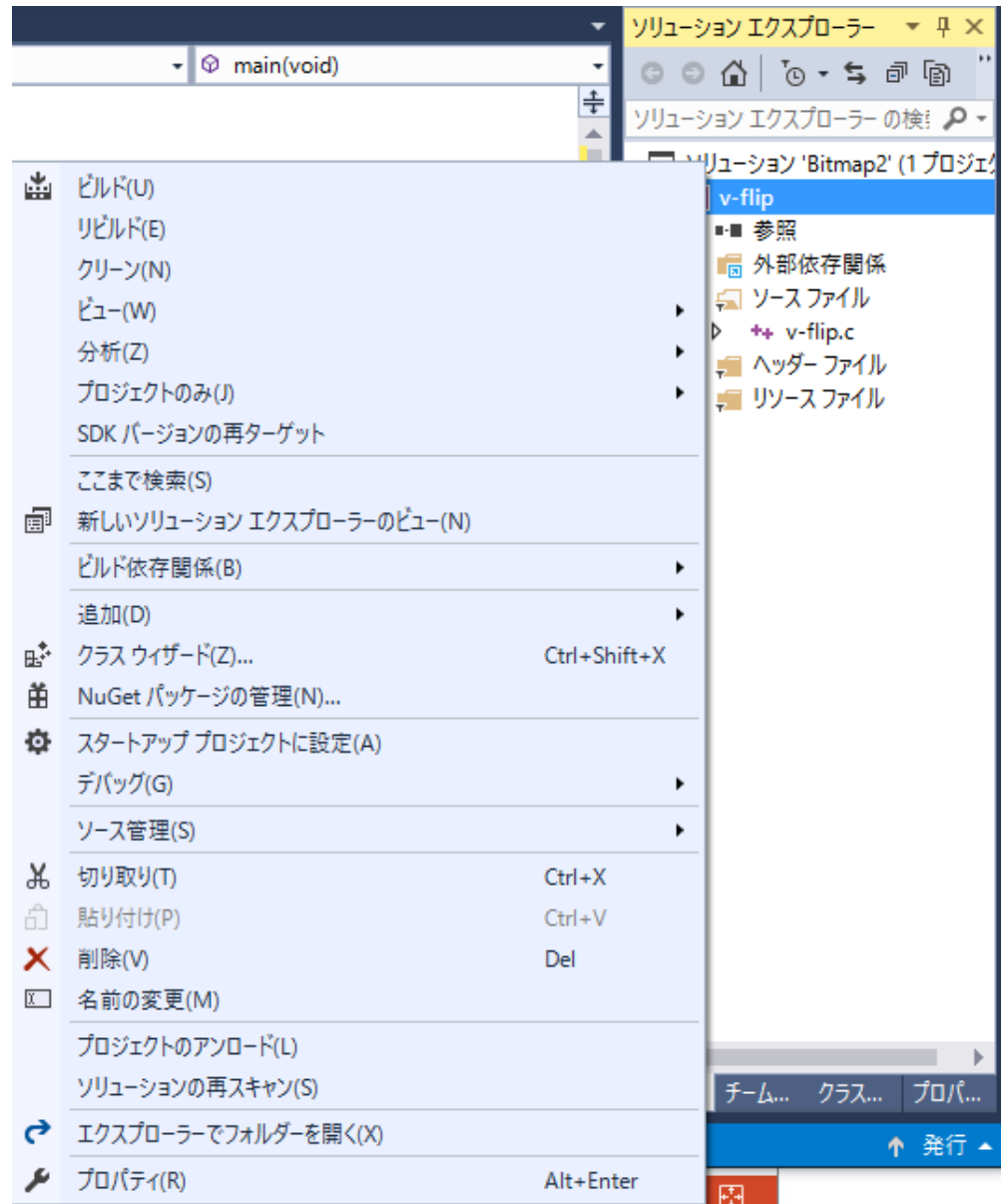
エラー一覧						
ソリューション全体		0 エラー	1 警告	0 メッセージ	ビルド + IntelliSense	エラー一覧を検索
コード	説明	プロジェクト	ファイル	行	抑制状態	
C6262	関数はスタックの '1573012' バイトを使用します: /analyze:stacksize '16384' を超えています。データの v-flip 一部をヒープに移動することを考慮してください。		v-flip.c	55		

スタックサイズの拡張(1)

使用する配列が (規定の) 1MB を超えるので

「ソリューションエクスプローラー」からプロジェクト名を右クリック

「プロパティ(R)」を選択



スタックサイズの拡張(2)

[構成プロパティ]

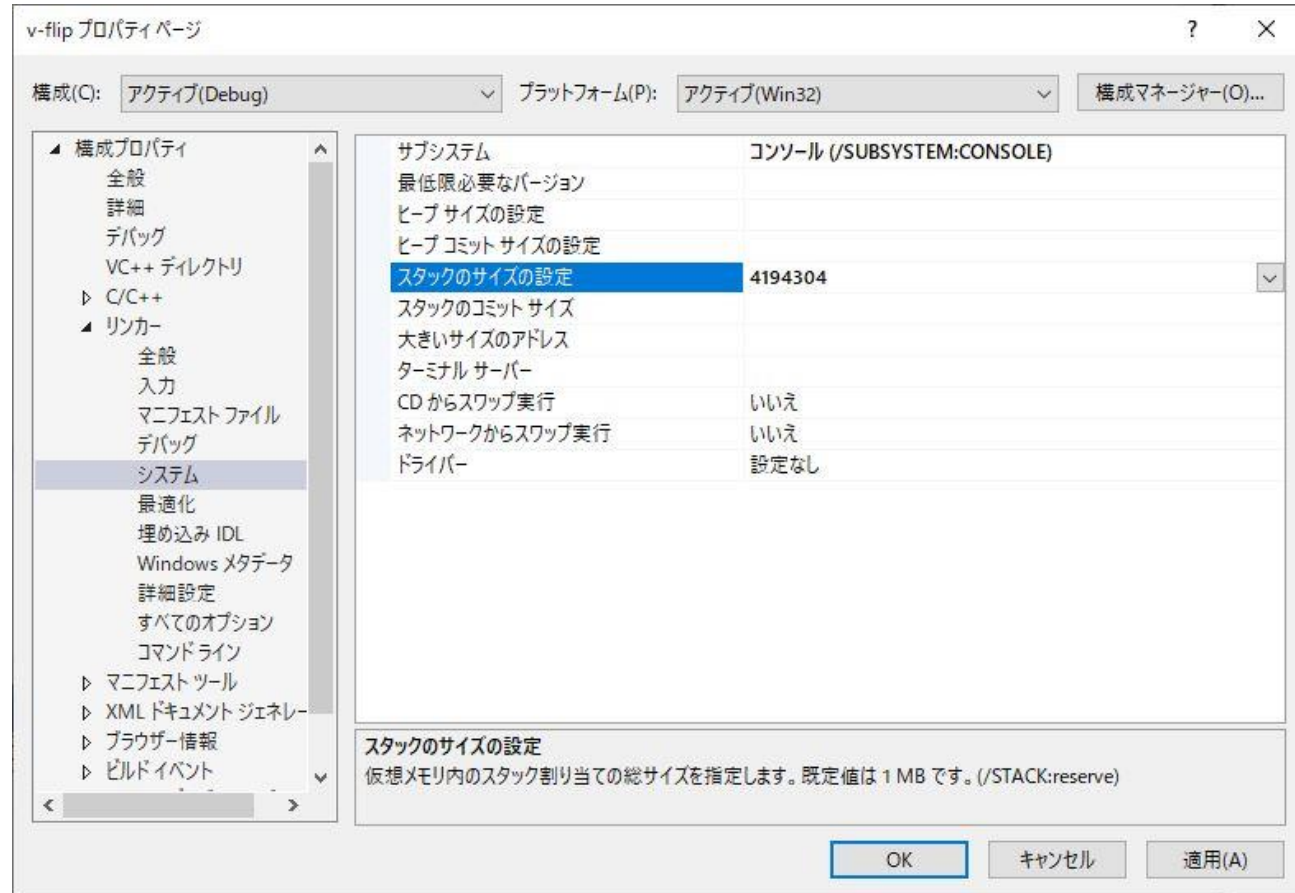
->[リンカー]

->[システム]

を選択

「スタックのサイズの
設定」でスタックサ
イズを入力
(4194304)

[適用(A)]を押し、
[OK]で終了。



【例題】 プログラム例 （抜粋）

```
81 // 上下反転画像生成
82 for (ii = 0; ii < YSIZE; ii++) {
83
84     // コピーする y 座標の先頭インデックスを計算する
85     py1 = HEADERSIZE + XSIZE * 3 * ii;
86     py2 = HEADERSIZE + XSIZE * 3 * (YSIZE - ii - 1);
87
88     for (jj = 0; jj < XSIZE; jj++) {
89
90         // コピーするドットの先頭インデックスを計算する
91         px1 = py1 + jj * 3; // コピー元ドットのインデックス
92         px2 = py2 + jj * 3; // コピー先ドットのインデックス
93
94         // BGR( Blue / Green / Red ) の3バイトをコピー
95         for (kk = 0; kk <= 2; kk++) {
96             buf2[px2 + kk] = buf1[px1 + kk];
97         }
98     }
99 }
100
101 /*
102     ここに buf2[] を画像ファイル baboon.bmp に
103     書き込むプログラムを追加せよ。
104
105 */
106
107
108
109
110 return 0;
111 }
112
```


プログラミング演習 (おみくじプログラム)

プログラム作成の要件

- おみくじは1回の試行で1つのくじがでること
- 引いたくじの内容はコマンドプロンプト上に表示させる
- くじの確率は、**大吉が1/10, 中吉が2/10, 吉3/10, 末吉が3/10, 凶が1/10**とする
- 本プログラムは擬似乱数を用いる
 - ✓ 擬似乱数生成にはrand()関数を用いる
 - ✓ 必要なヘッダーファイルについては自分で調べる
 - ✓ rand()関数の記述の前には, srand(time(NLL));を記述すること
 - ✓ rand()を用いてランダムに0から9の整数を生成する

実行結果（おみくじプログラム）

```
コマンドプロンプト
Microsoft Windows [Version 10.0.19042.1055]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Nakano>cd C:\Users\Nakano\source\repos\omikuj\Debug

C:\Users\Nakano\source\repos\omikuj\Debug>omikuj
おみくじプログラム開始する
末吉

C:\Users\Nakano\source\repos\omikuj\Debug>omikuj
おみくじプログラム開始する
末吉

C:\Users\Nakano\source\repos\omikuj\Debug>omikuj
おみくじプログラム開始する
大吉

C:\Users\Nakano\source\repos\omikuj\Debug>omikuj
おみくじプログラム開始する
吉

C:\Users\Nakano\source\repos\omikuj\Debug>omikuj
おみくじプログラム開始する
末吉

C:\Users\Nakano\source\repos\omikuj\Debug>omikuj
おみくじプログラム開始する
大吉

C:\Users\Nakano\source\repos\omikuj\Debug>omikuj
おみくじプログラム開始する
凶

C:\Users\Nakano\source\repos\omikuj\Debug>omikuj
おみくじプログラム開始する
吉
```

何回か試行してみて、設定している確率でおみくじが出ているか確認すること！

大吉の確率1/10

中吉の確率2/10

吉の確率が3/10

末吉の確率3/10

凶の確率1/10

プログラム例や解説については次週行う！