

### ファイルを作成して、それに出力する。

`fprintf(stdout, "data");`は、標準出力ストリームと呼ばれる `stdout` に"data"を出力しました。`stdout` は、コンソール画面へつながるストリームなので、画面に表示したわけです。では、`stdout` でなく指定するとどうなるのでしょうか？

例えば、あるファイルへつながるストリームです。

そうです！ そうすると、ファイルに出力する（書き込む）ことになります

そしてそのためには、希望のファイルにつながる `stdout` に代わるストリーム管理用の変数が必要で、そこにファイルをセットしなければなりません。その変数の種類は `(FILE *)`型で、`stdio.h` をインクルードすれば使えるようになります。


そして、希望のファイルへつながるストリームの取得には `fopen` 関数を使います。これも `stdio.h` の中でプロトタイプ宣言があります。具体例で説明します。

```
#include <stdio.h>
main()
{
    FILE * fpw; /* ファイル出力ストリーム用変数 */
    fpw = fopen("test.txt" , "w" );//ストリーム取得
    fprintf(fpw, "hello\n");
    fprintf(fpw, " hello\n hello\n");
    fclose(fpw);//ストリームアクセス終了
}
```

変数 `fpw` の種類(型)です

hello  
hello  
hello

`fprintf` の `fpw` を、`stdout` にすると、これまで通りに、画面へ表示されます。

実行すると、画面には何も表示されませんが、代わりに、 test.txt テキストドキュメント 1 KB が作られて、右のように開けます。



なお、`fopen` で開いたファイルは、`fclose` で必ず閉じる必要があります。

オペレータにおいて、メモ帳で、データを書き込む場合も、ファイルを開いて、最後に閉じる必要があることと同じです。なお、`fopen` の行の意味は、次のようになります。

```
fpw = fopen("test.txt" , "w" );
```

作成する **ファイル名** で、**相対パス** や **絶対パス** で指定できます。

書き込みモードで、ファイルを **新規作成** する。ファイルが存在する場合は、そのファイル内容は、**無くなって**、空の状態にする指定です。

相対パスを指定すると、この作品を実行した位置を基準にファイルを作ります。Visual Studio .NET 2003 では、開発に使っているフォルダ (CTEST) の中に作ります。なお、既にファイルがあって、そのファイルを別のプログラムが開いていると、`fopen` の処理は

失敗します。また絶対パス指定などで存在しないドライブやディレクトリを指定したり、指定したディレクトリ内にファイル作る権限のない人が、実行した場合も失敗します。

このように fopen の処理が失敗した場合は、0 の値が戻り値として戻ります。

(よって、戻り値を if で評価して、エラーチェックするのが一般的です。)

**問題1** 0 から 100 の範囲となる乱数を、最初に入力した生成個数の数だけ生成し、生成した値が並ぶ『E01.txt』ファイルを作りなさい。(なお乱数の種は、123 を使ってください。)

<b>実行例</b>	<b>tE01.c のソースファイル名で作る</b>	
乱数生成個数>5 終了しました。	この実行で作成される E01.txt の内容	36 65 47 75 43

次のステップでプログラムを作成します。(ファイル化は最後のステップで、最初は画面に出します)

**ステップ1** 次のコードを作成して、動かします。

```
#include <stdio.h>
#include <stdlib.h> /* rand や srand 関数を使うため */
main()
{
    int n; /*      個数記憶用 */
    /* 1 */
    fprintf(stdout, "乱数生成個数>");
    fscanf(stdin, "%d" , &n);
    /* 2 */
    srand(123); /* 123 の種で乱数生成の準備を行う */
    fprintf(stdout, "%d¥n", rand() % 50);
    fprintf(stdout, "%d¥n", rand() % 50);
    fprintf(stdout, "%d¥n", rand() % 50);
    fprintf(stdout, "終了しました。¥n");
    /* 3 */
}
```

**ステップ2** 乱数を表示している上記2行を削除して、残った1行を、nの回数だけ繰り返す while にします。また、生成範囲が 0 ~ 100 になるように、%演算子で細工した式に変更し、実行確認します。

**ステップ3** ステップ2が正しく動いたらコメント/\*1\*/ で、FILE \* fpw; のストリーム変数を用意し、コメント/\*2\*/で fpw = fopen("E01.txt", "w"); によりファイルを作ってそのストリームを取得し、コメント/\*3\*/に fclose(fpw); の閉じる処理を書きます。あとは、ファイルに書き込みたい範囲の fprintf の stdout を fpw に変更すれば出来上がりです。

**問題2** 小学1年生の足し算学習ソフトを作ります。  
乱数を2つ 1から9の範囲で作り、それを表示します。  
その足し算の問題を出題し、オペレータの入力が  
合っていれば Good! 合わなければ 答えを表示します。  
これを4回行わせて、正解した回数を表示させます。

**tE02.c のソースファイル名で作る**

1回目	1+4	はいくつ?>5
		Good!
2回目	6+5	はいくつ?>12
		答えは11です
3回目	4+6	はいくつ?>10
		Good!
4回目	9+9	はいくつ?>18
		Good!
		3つ正解です

**問題3** 引き算も出題できるようにしましょう。 **ファイル名 tE03.c** また連続して4回正解しないと終わらないように変更しましょう。(正解回数でなくクリアまでの回数を表示させます)