

オブジェクト指向と ゲームプログラミング

基礎編 - 第17回 コマンドライン引数

コマンドライン引数

main関数には、引数を渡すことができます。main関数に渡す引数のことをコマンドライン引数といいます。main関数へ渡せる引数は決まっており、以下の2つです。

- ・引数の総個数
- ・引数の文字列の配列

一般的にmain関数が引数を受け取る場合、

```
int main(int argc, char* argv[])
```

と記述します。それぞれの引数には、以下の情報が格納されます。

```
int  argc    引数の個数(プログラム名も含む)
char* argv[] 引数の文字列の配列
```

たとえば、コマンドラインから「D:¥>PROG ABC 123 /?」とプログラムを起動した場合、main関数の引数argcに4、argv[0]に"D:¥PROG.EXE"、argv[1]に"ABC"、argv[2]に"123"、argv[3]に"/?"が渡されます。

argc	4
argv[0]	D:¥PROG.EXE
argv[1]	ABC
argv[2]	123
argv[3]	/?

練習問題

- 1 main関数に渡された引数を表示する以下のプログラムを入力しましょう。

```
#include <stdio.h>

int main(int argc, char* argv[])
{
    printf("引数の個数 = %d\n", argc);
    for(int i = 0; i < argc; i++)
        printf("%2d番目の引数 = %s\n", i, argv[i]);

    return EXIT_SUCCESS;
}
```

- 2 1のプログラムをコマンドプロンプトから適当な引数を指定して実行しましょう。なお、引数の指定は「プロジェクト(P) 設定(S) デバッグ プログラムの引数(U)」でも行えます。

3 以下のプログラムは、指定されたファイルを暗号化または復号して出力するものです。プログラム中の ? を埋めて完成させましょう。

- crypt.cpp -

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char* argv[])
{
    // 引数チェック
    if(???? < 4) {
        puts("用法 : crypt 入力ファイル名 出力ファイル名 暗号鍵");
        return EXIT_FAILURE;
    }

    // ファイルを開く
    FILE* in_file = ?????(argv[?], "rb");           // 入力ファイル
    FILE* out_file = ?????(????????, "wb");        // 出力ファイル
    if(in_file == ??? || out_file == ???) {
        puts("ファイルが開けません");
        return EXIT_FAILURE;
    }

    // 暗号鍵で乱数初期化
    const int KEY = atoi(????????);                // 引数は文字列なので、数値に変換する
    srand(KEY);

    // 暗号化および復号
    char c;
    while(fread(&c, sizeof(char), 1, in_file) >= 1) { // 入力ファイルから1バイト読み込む
        const int r = rand() / 256;                 // 0~255の乱数を生成
        c ^= r;                                     // 乱数と排他的論理和をとり、暗号化する
        fwrite(&c, sizeof(char), 1, out_file);     // 出力ファイルに1バイト書き込む
    }

    // ファイルを閉じる
    ?????(out_file);
    ?????(in_file);

    puts("終了しました");

    return EXIT_SUCCESS;
}
```

このプログラムは、排他的論理和と乱数を組み合わせた簡単な暗号化のプログラムです。暗号化データを再度このプログラムにかけると元のデータに戻ります。

使用法は、コマンドプロンプトから「D:¥>crypt 入力ファイル名 出力ファイル名 暗号鍵」のように入力します。指定された「入力ファイル」を暗号化または復号し、「出力ファイル」に出力します。「暗号鍵」は、数値を指定します。ここで指定した数値から乱数を生成し、暗号化および復号します。復号の場合は、暗号化したときと同じ数値でなければなりません。

たとえば、以下のようにすると「text1.txt」が数値「1234」をもとに暗号化または復号され、「text2.txt」に出力されます。

```
D:¥>crypt text1.txt text2.txt 1234
```