

Training 11

特殊な処理

株式会社イーシーエス 出版事業推進委員会

Lesson1 ファイルの入出力



Poi nt◆◆ファイルの入出力を理解しよう!!

fopen 関数は、ファイルの読み書きをするため、指定されたモードでファイルのオープンを行います。
fclose 関数は、fopen 関数にてオープンしたファイルのクローズを行います。
ファイルのオープンとクローズはセットなので、ファイルを使い終わったら、クローズしましょう。

※ファイルのオープンモードについては巻末資料参照

【問題1】 次の問題の□を埋めて fopen 関数を完成させなさい。

① 読み込み専用

```
fopen("test.c", "□");
```

③ 追加書き込み

```
fopen("test.c", "□");
```

⑤ データの更新(読み書き)
ファイルがなければ自動作成

```
fopen("test.c", "□");
```

⑦ バイナリモードでの読み込み専用

```
fopen("test.c", "□");
```

⑨ バイナリモードでの追加書き込み

```
fopen("test.c", "□");
```

② 書き込み専用

```
fopen("test.c", "□");
```

④ データの更新(読み書き)
ファイルがなければエラー

```
fopen("test.c", "□");
```

⑥ 追加書き込み(③の別書式)

```
fopen("test.c", "□");
```

⑧ バイナリモードでの書き込み専用

```
fopen("test.c", "□");
```

【問題2】 絶対パス指定を使い、読み込み専用モードで"C: ¥Source¥test.c"ファイルを開く fopen 関数を書きなさい。

```
fopen("□①", "□②")
```

【問題3】 次の□部分を埋め、プログラムを完成させなさい。

【処理内容】

test.c ファイルを開き、処理を行わずにそのまま閉じる。

```
int main(void)
{
    FILE *fp

    if ( ( □① = fopen( "test.c", "r" ) ) == NULL )
    {
        printf( "ファイルを開けません" );
        return 1;
    }

    fclose( □② );
    return 0;
}
```

Lesson2 ファイルの操作



Point◆◇ファイル操作を理解しよう!!

ファイルの操作をする関数として、レコード単位で入力する `fread` 関数、出力する `fwrite` 関数があります。ファイル操作での読み/書きは1レコード読むごとに自動的に読み/書きする位置がずれるのが特徴です。そこで任意の読み/書き位置に移動させる場合に使用する `fseek` 関数があります。しっかりと使いこなしましょう。

【問題1】 次の□部分を埋め、プログラムを完成させなさい。

【処理内容】

test.c ファイルを開き、先頭から5つの `int` 型データを読み出して出力する。

```
#include <stdio.h>

#define LISTNUM 5

void main(void)
{
    FILE *fp;
    int list[LISTNUM];
    int num;
    int i;

    /* バイナリ読み込み モードでファイルを開く */
    if ( ( fp = fopen( "test.c", □① ) ) != NULL )
    {
        /* データを読み込む */
        num = fread( □②, □③, □④, □⑤ );
        printf( "読み出したデータ数は %d 個です。¥n", num );

        for ( i = 0; i < LISTNUM; i++ )
        {
            printf( "バッファの内容: %2d 個目 = %d¥n", i+1, □⑥ );
        }

        fclose( fp );
    }
    else
    {
        printf( "ファイルを開けませんでした。¥n" );
    }
}
```

【問題2】 次の□部分を埋め、プログラムを完成させなさい。

【処理内容】

test.c ファイルを開き、5 つの int 型データを書き込む。

```
#include <stdio.h>
#define LISTNUM 5

void main(void)
{
    FILE *fp;
    int list[LISTNUM];
    int num;
    int i;

    /* バイナリ書き込み モードでファイルを開く */
    if ( (fp = fopen( "test.c", □①□ ) ) != NULL )
    {
        for ( i = 0; i < LISTNUM; i++ )
        {
            printf( "書き込むデータを入力して下さい%2d 個目:", i+1 );
            scanf( "%d", &list[i] );
        }

        /* データを書き込む */
        num = fwrite( □②□, □③□, □④□, □⑤□ );
        printf( "%d 個のデータを書き込みました。¥n", num );
        fclose( fp );
    }
    else
    {
        printf( "ファイルを開けませんでした。¥n" );
    }
}
```

【問題3】 次の□部分を埋め、プログラムを完成させなさい。

【処理内容】

test.c ファイルを開き、先頭から1つずつ飛ばしで5文字を読み出して出力する。

```
#include <stdio.h>

void main(void)
{
    FILE *fp;
    char list[6] = {"%0"};
    char moji;
    long int pos;
    int num = 0;
    int i;

    /* バイナリ読み込み モードでファイルを開く */
    if ( ( fp = fopen( "test.c", □①□ ) ) != NULL )
    {
        for ( i = 0; i < □②□ ; □③□ )
        {
            /* 文字を読み込む */
            fread( &moji, sizeof( char ), □④□, fp );

            /* 文字を配列に格納 */
            list[□⑤□] = moji;

            /* 次の読み出し位置に移動 */
            pos = □⑥□;
            fseek( □⑦□, pos, SEEK_CUR );
        }
        list[□⑧□] = '%0';

        printf( "読み出した文字数は %d 個です。%n", num );
        printf( "バッファの内容 = %s%n", list );
        fclose( fp );
    }
    else
    {
        printf( "ファイルを開けませんでした。%n" );
    }
}
```


解答

Training11 特殊な処理

Lesson1 ファイルの入出力

問題 1	①r	②w	③a	④r+
	⑤w+	⑥a+	⑦rb	⑧wb
	⑨ab			

問題 2	①C: ¥¥Source¥¥test.c または C: /Source/test.c
	②r

問題 3	①fp	②fp
------	-----	-----

Lesson2 ファイルの操作

問題 1	①"rb"	②list	③sizeof(int)
	④LISTNUM	⑤fp	⑥list[i]

問題 2	①"wb"	②list	③sizeof(int)
	④LISTNUM	⑤fp	

問題 3	①"rb"	②5	③i++	④1
	⑤num++	⑥sizeof(char)		
	⑦fp	⑧num		

巻末資料

ファイルのオープンモード

"r"	テキスト読み込みモード	"rb"	バイナリ読み込みモード
"w"	テキスト書き込みモード	"wb"	バイナリ書き込みモード
"a"	テキスト追加書き込みモード	"ab"	バイナリ追加書き込みモード
"r+"	テキスト読み取り+書き込みモード	"r+b" or "rb+"	バイナリ読み込み+書き取りモード
"w+"	テキスト書き込み+読み取りモード	"w+b" or "wb+"	バイナリ書き込み+読み取りモード
"a+"	テキスト追加書き込み+読み込みモード	"a+b" or "ab+"	バイナリ追加書き込み+読み取りモード