SANGI AWARD IT 競技会 事前学習 第４ 回

ファイルの入出力及び文字列のチェックを含めた問題のサンプルです
この問題を通してファイルの入出力の方法を確認してください。

2012 年度[問題 3]より

C言語で記述されたソースプログラムをファイルから読み込んで，注釈を除去して別のファイルへ出力するプログラムを作成せよ。

(1) プログラムの仕様

C言語で記述されたソースプログラムをファイル「in_data.txt」から読み込んで、注釈を除去し、ファイル「out_data.txt」へ出力する。

なお、in_data.txt の内容は、(4)－①のとおりである。

「in_data.txt」と「out_data.txt」は用意されているものを利用し「in_data.txt」の変更は許されない。

(2) ソースプログラムの表記に関する説明

① このプログラムで扱う注釈は，文字列リテラル（"）又は注釈の中を除いて，“/*”で始まり“*/”で終わる文字の列と、“//”で始まり行末で終わる文字の列である。

② 使用可能文字の種類は，JIS X 0201（7 ビット及び 8 ビットの情報交換用符号化文字集合）である。ただし，“￥”は“￥”と表記する。

③ 次のような注釈の入れ子記述は、無いものとする。

例 /* aaaaa /* bbbbbb */ ccccc */

④ 文法上の誤りはないものとする。

(3) プログラムの作成についてのヒント。

① 二重引用符（"）を検出すると，文字列リテラルの開始と解釈し，対応する二重引用符を検出するまで，文字の列を読み込んでそのまま出力する。

② “/*”を検出すると，注釈の開始と解釈し，最初に現れる“*/”までの文字の列を読み飛ばす。

③ “//”を検出すると，注釈の開始と解釈し，行末までの文字の列を読み飛ばす。

(4) プログラムによる注釈除去の実行例。

① 入力ソースプログラム 「in_data.txt」

```
/* This Program is File I/O Process *
  * File Line Read & Write. */
#include <stdio.h>

void main( void )
{
    FILE *fp;                    /* file pointer */
    char buf[256];    /* input buffer */

    if(( fp = fopen( "input_data.txt", "r" )) != NULL ){

        if( fgets( buf, 256, fp ) == NULL )    /* file read */
            printf( "fgets function is /*error*/¥n" ); // error message
        else
            printf( "%s", buf );  // file write
        fclose( fp );
    }

}
```

② 注釈を除去した後の出力結果 「out_data.txt」

```
#include <stdio.h>

void main( void )
{
    FILE *fp;
    char buf[256];

    if(( fp = fopen( "input_data.txt", "r" )) != NULL ){

        if( fgets( buf, 256, fp ) == NULL )
            printf( "fgets function is /*error*/¥n" );
        else
```

```
            printf( "%s", buf );
        fclose( fp );
    }

}
```

[Check Point]

・ファイル「in_data.txt」が取り込まれている

・ファイル「out_data.txt」に出力がされている

・文字列処理が適切にされている

解答例　及び　解説(プログラム中のコメント)

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

void main()
{
        int   i, j;
        FILE  *fin, *fout;  //入力ファイル及び出力ファイル用のファイルポインタ
        char  buf[256], out[256];     //入力データ及び出力データ格納用配列（１行分最大２５５文字)
        //quote:引用符 "～" チェック用  comment: コメント/*～*/チェック用
        //  comment2:一行コメント//～¥n チェック用
        //それぞれ途中の状態の時は1を代入
        int   quote = 0, comment = 0, comment2 = 0;

        //入力ファイルを読み取りモード、出力ファイルを書き込みモードでオープン、
        //正常にオープンできたときのみ処理を行う。
        if ((fin=fopen("in_data.txt", "r")) != NULL &&
                        (fout=fopen("out_data.txt", "w")) !=NULL ){ //C言語標準関数の使用時
        //セキュリティ強化版のfopen関数使用時（現在はこちらを推奨)
        //if (fopen_s(&fin, "in_data.txt", "r") == 0 &&
                                        fopen_s(&fout, "out_data.txt", "w") == 0){

                //入力ファイルを一行ずつ読み取り（読み取る行がなくなれば終了)
                while (fgets(buf, 256, fin) != NULL){
                        comment2 = 0;
                        //読み取った一行分のデータを先頭から一文字ずつチェックしていく
                        //（行の終わりは¥0)
                        for (i = 0, j = 0; i<256 && buf[i] != '¥0'; i++){
                                //引用符 " のチェック
                                if (buf[i] == '"'){
                                        if (quote == 0) quote = 1;    //引用符 " の始まり
                                        else            quote = 0;    //引用符 " の終わり
                                }
                                //改行¥nが来たとき
                                if (buf[i] == '¥n'){
                                        comment2 = 0;        //一行コメント//～の終了
```

```
                    }
                    //引用符 " "の外部で/*（コメントの開始）が来たとき
                    if (quote == 0 && buf[i] == '/'  && buf[i + 1] == '*'){
                        comment = 1;        //コメントの開始
                        i++;
                    }
                    //引用符の外部で*/（コメントの終了）が来たとき
                    else if (quote == 0 && buf[i] == '*'  && buf[i + 1] == '/'){
                        comment = 0;        //コメントの終了
                        i++;
                    }
                    //引用符の外部で//(一行コメント）が来たとき
                    else if (quote == 0 && buf[i] == '/'  && buf[i + 1] == '/'){
                        comment2 = 1;       //一行コメントの開始
                                            //(終了は¥nが来たとき)
                        i++;
                    }
                    //コメント内の文字でなければ出力バッファに出力
                    else if (comment == 0 && comment2 == 0){
                        out[j] = buf[i];
                        j++;
                    }
                }
                //出力バッファの最後に¥0を付加
                out[j] = '¥0';
                //コメント内の文字が除かれている一行分の出力バッファを
                //出力ファイルに出力
                fputs(out, fout);
            }
            //すべての行の処理が終了したので入力ファイル、出力ファイルをクローズ
            fclose(fin);
            fclose(fout);
        }
    }
```