

C言語講座 第5回

ファイル入出力

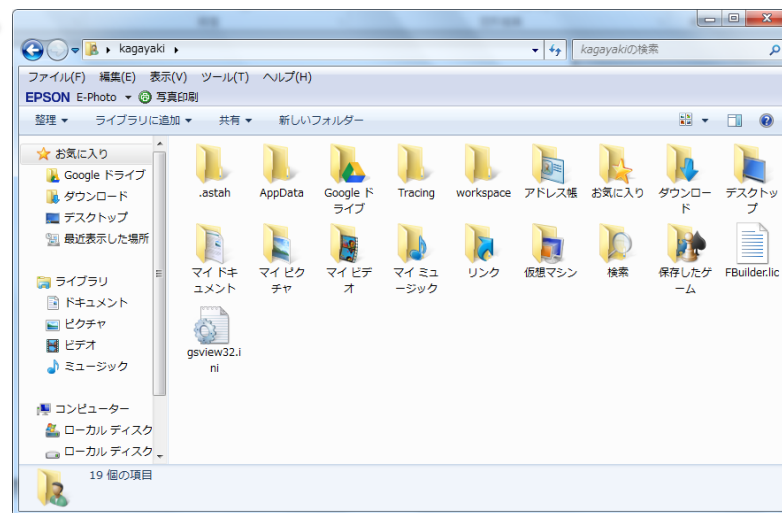
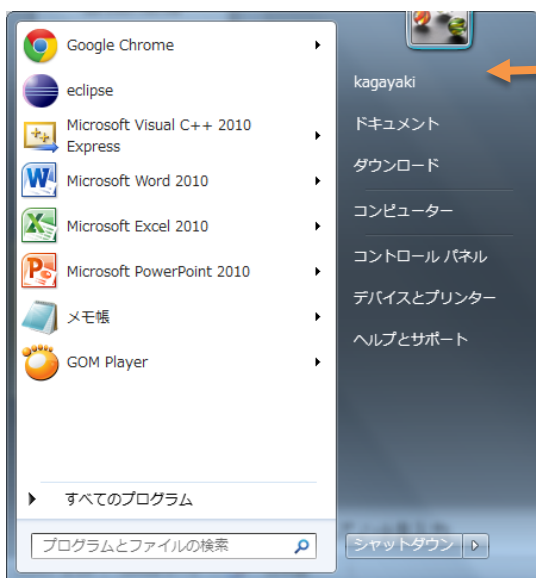
担当:おさない、水谷、山崎

ファイル入出力とは？

- ファイルから入力进行もらう
- ファイルに出力する
- 具体的には
 - 大きな配列を、統計データなどで初期化する時
 - プログラムの实行結果をグラフに表す时などに使われる

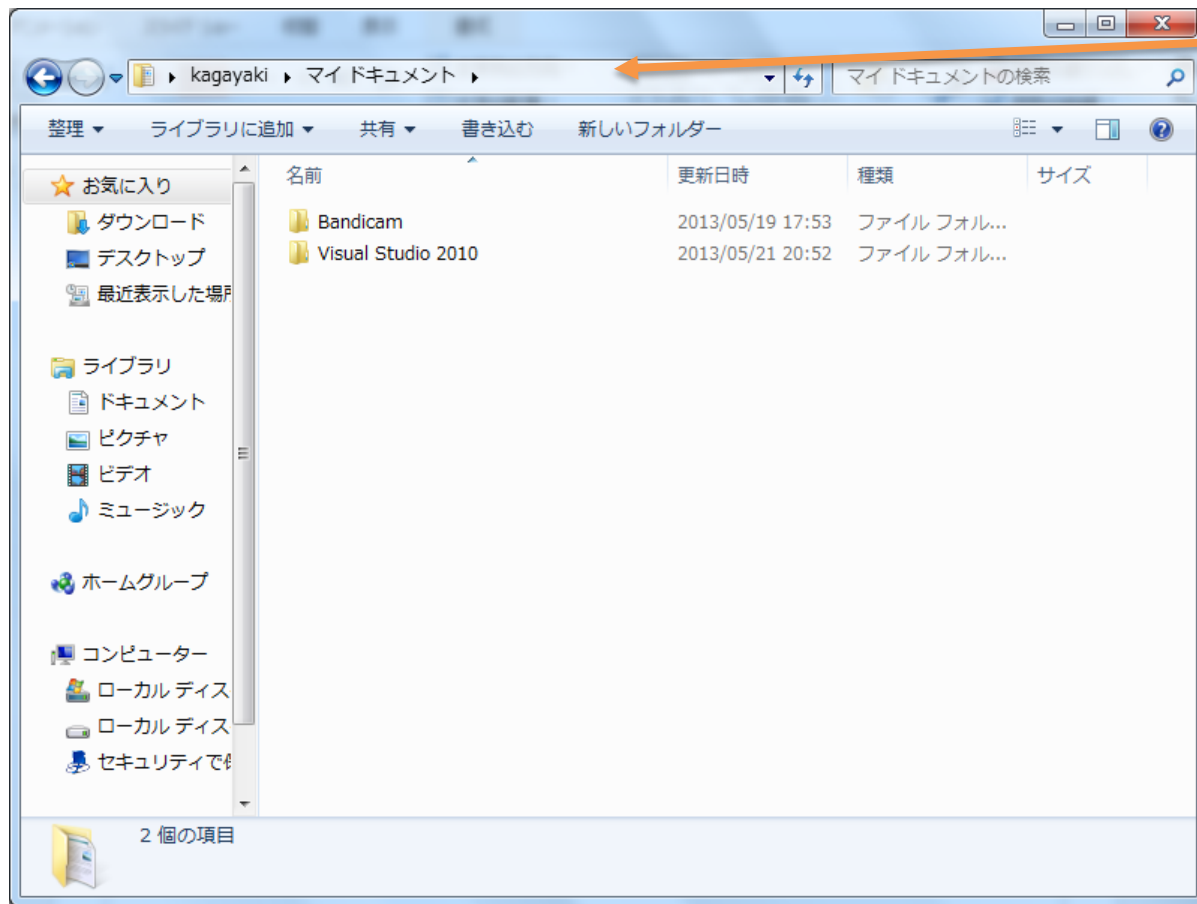
ファイル入出力のプログラム例1/5

- 何も説明していませんが、ファイル入出力がどんな動作をするのか実行してみましよう！
- まず、左下のスタートボタン→(パソコンのユーザー名)→マイドキュメント



ファイル入出力のプログラム例2/5

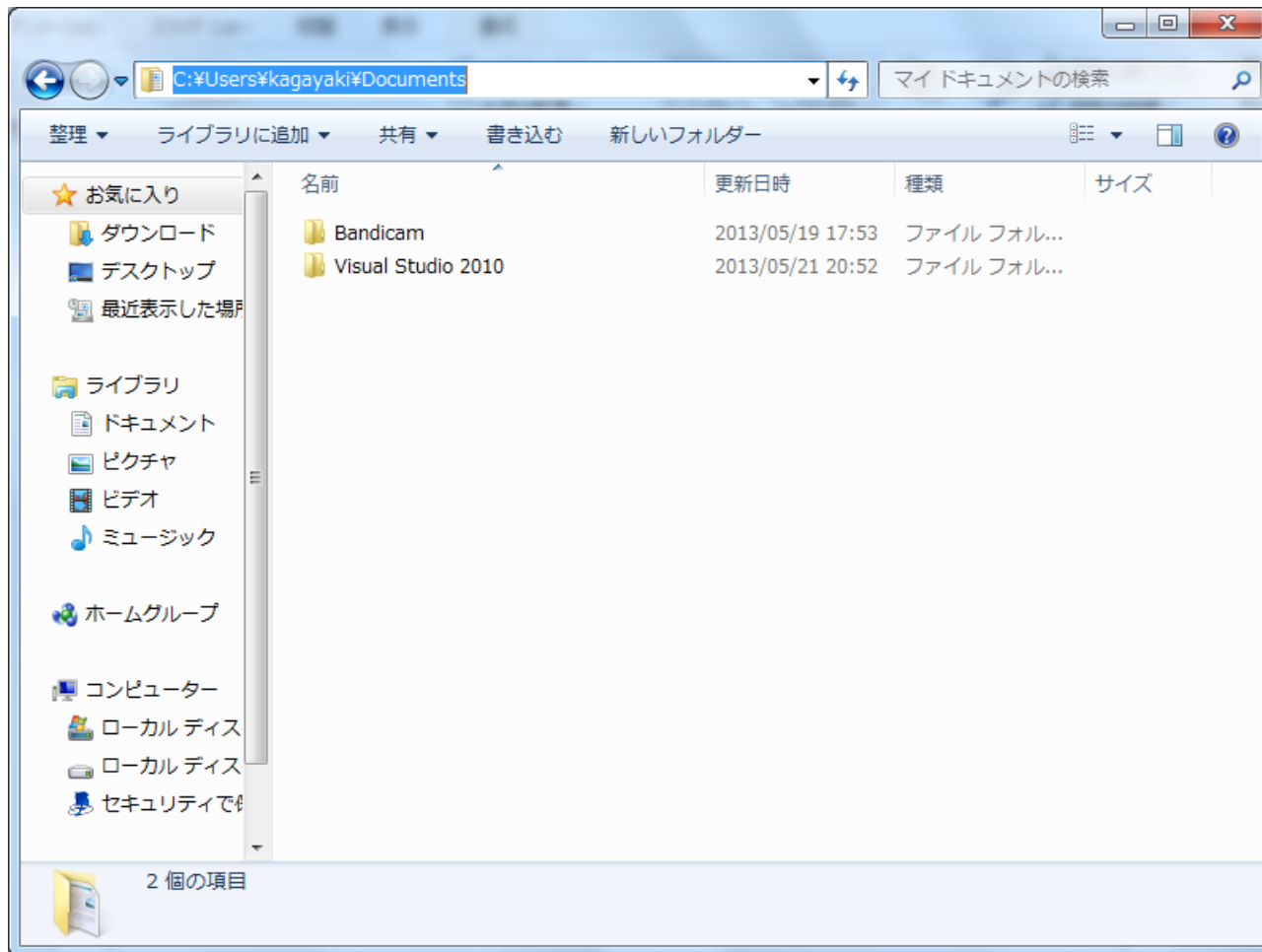
- こんな画面で



このあたりを
クリックすると、

ファイル入出力のプログラム例3/5


- パスが出てくるのでコピーしておく。



ファイル入出力のプログラム例4/5

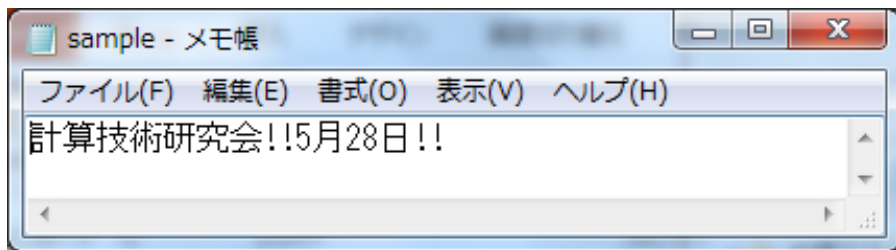
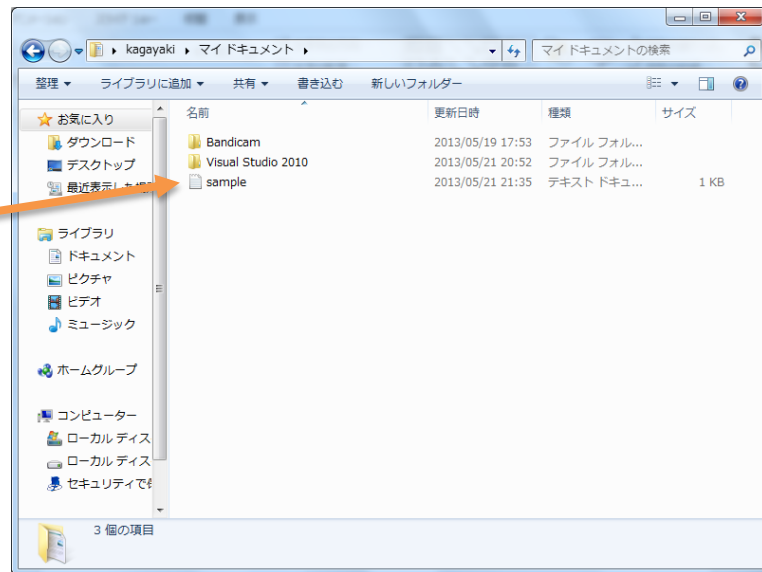
```
#include<stdio.h>
int main(){
    FILE *fp;
    fp=fopen("C:¥¥Users¥¥kagayaki¥¥Documents¥¥sample.txt","a");
    if(fp==NULL){
        printf("ファイルオープンに失敗しました¥n");
        return 0;
    }
    fprintf(fp,"計算技術研究会!!5月28日!!");
    fclose(fp);
    return 1;
}
```

コピーしたパスの、¥を2つにしたもの




ファイル入出力のプログラム例5/5

- デバッグなしで開始→実行結果には何も出ていないが...
- ドキュメントの中にsample.txtができています
- ダブルクリックして開くと、



今日の講座はこんなプログラムについてです。

ファイル入出力のプログラムの流れ

- ファイルポインタを定義する
 - FILE *fp;
 - ファイルポインタを初期化する “ ” を忘れずに!
 - fp=fopen(“(パス)¥¥(ファイル名.拡張子)”,”(モード)”);
 - ファイルポインタの初期化に失敗すると、
NULLが代入される
 - if(fp==NULL){
 printf(“ファイルのオープンに失敗しました¥n”);
 return 0;//main関数を抜けてプログラムを終了する
}
 - 処理 - fprintf(~); など
 - ファイルポインタを解放する (約束事のようなもの)
 - fclose(fp);
- 

fopen関数の引数の「パス」について 参考までに

- パスはファイルの場所を示すもの `cf:path` 道

- 絶対パス

- 例) `C:\Users\kagayaki\Documents\sample.txt`
- >ドキュメントの中の`sample.txt`ファイルを示す

- 相対パス

- 例) `sample.txt`
- >ワークスペースがデフォルトだと、ドキュメントの中の
Visual Studio 2010 → Projects → (作成したプロジェクト)
→ ...
- プログラムがある場所の`sample.txt`を示す

fopen関数の引数の「モード」について

- テキストファイル(文字とか数字とかのファイル)を扱う時

モード	意味
r	読み込み専用; ファイルを編集できない
w	書き込み専用; ファイルの中身を見れない
a	ファイルの最後に追加の書き込み
r+	読み込みと書き込み(ファイルがないとエラー)
w+	読み込みと書き込み(ファイルを作成する)
a+	読み込みと追加の書き込み(ファイルがないと作成)

- バイナリファイル(画像とか音楽とかのファイル)を扱う時

モード	意味
rb	バイナリモードで読み込み専用
wb	バイナリモードで書き込み専用
ab	バイナリモードで追加の書き込み

問題1

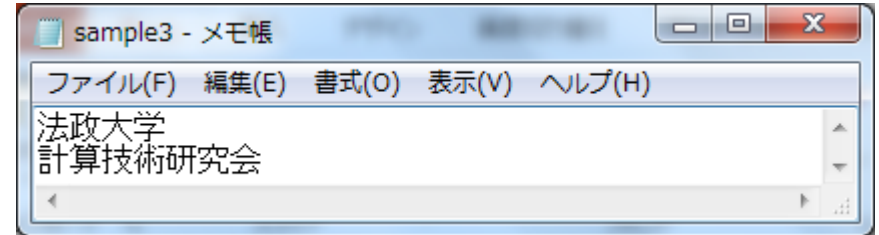
- デスクトップに、自分の名前が書いてある `sample2.txt` を プログラムを実行することによって、作成してください。 ファイルポインタの初期化(ファイルのオープン)に失敗した場合も考慮してください。(必要に応じて6ページの例を見てください)
また、モードを変更してどういう挙動になるか見てください。
- 注意 最後のほうのページに答え

ファイル操作の標準関数

- fopen関数やfclose関数,fprintf関数
- fgetc(ファイルポインタ)
 - 先頭の一文字を得る。
 - 実行後はファイルポインタが次の一文字を指す。
 - ・・・サンプルプログラムあります。
- fputc(char型の文字,ファイルポインタ)
 - 一文字書き込む。
 - 実行後はファイルポインタがひとつ先を指す。
 - ・・・サンプルプログラムあります。

fgetc関数の例

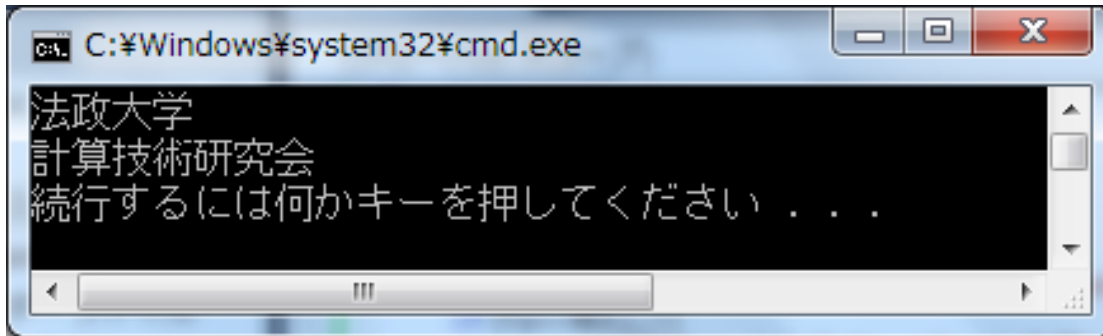
- あらかじめ、ドキュメントの中にsample3.txtを作成しておく



```
#include<stdio.h>
int main(){
    char c;
    FILE *fp;
    fp=fopen("C:¥¥Users¥¥kagayaki¥¥Documents¥¥sample3.txt","r");
    if(fp==NULL){
        printf("ファイルのオープンに失敗しました¥n");
        return 0;
    }
    while((c=fgetc(fp))!=EOF){
        printf("%c",c);
    }
    printf("¥n");
    fclose(fp);
    return 1;
}
```

各自のパス

実行結果



```
C:\¥Windows¥system32¥cmd.exe
法政大学
計算技術研究会
続行するには何かキーを押してください...
```

- sample3.txtファイルの中身が結果に表示される。
- sample3.txtから先頭の一文字読み取る
→それを表示→ファイルポインタが次の文字を指す
→ファイルポインタの指している一文字を読み取る
- ファイルポインタがEOFを指すまで上記を繰り返す。
- EOF (End Of File) はファイルの最後に必ずあるもの。
- fgetc関数はファイルポインタの指す一文字を返す。

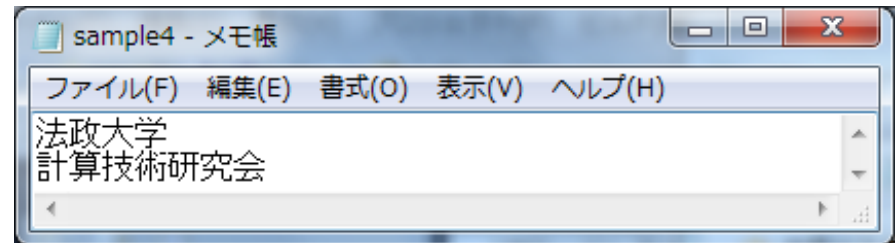
fputc関数の例

```
#include<stdio.h>
int main(){
    char c;
    FILE *fp,*cfp;
    fp=fopen("C:¥¥Users¥¥kagayaki¥¥Documents¥¥sample3.txt","r");
    cfp=fopen("C:¥¥Users¥¥kagayaki¥¥Documents¥¥sample4.txt","w");
    if(fp==NULL || cfp==NULL){
        printf("ファイルのオープンに失敗しました");
        return 0;
    }
    while((c=fgetc(fp))!=EOF){
        fputc(c,cfp);
    }
    fclose(fp);
    fclose(cfp);
    return 1;
}
```

2枚前で作ったsample3.txtがあるパス

実行すると

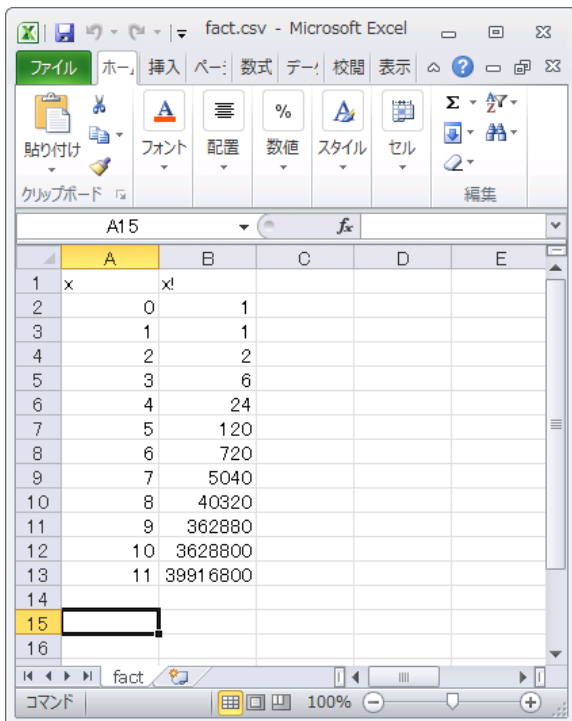
- sample3のコピーのsample4が作成される。



- sample3からfgetc関数を実行→先頭の一文字を変数cに代入→fputc関数を使って、sample4に変数cの一文字を書き込む
- ファイルポインタfpがEOFを指すまで上記を繰り返す。
- fputcはファイルポインタの指している場所に一文字を書き込む

もうすこし実用的に

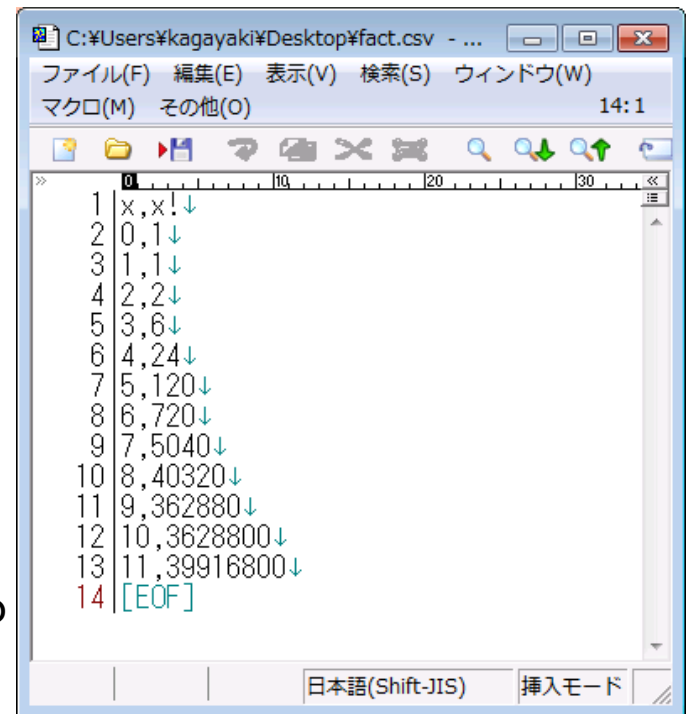
- .txtファイルを操作してきましたが、.csvファイルを操作してみましょう。
- csvファイルはこんなものということを軽く頭の隅に



エクセルで開く

秀丸エディタで開く

カンマで区切られていて、
行末には改行記号がある



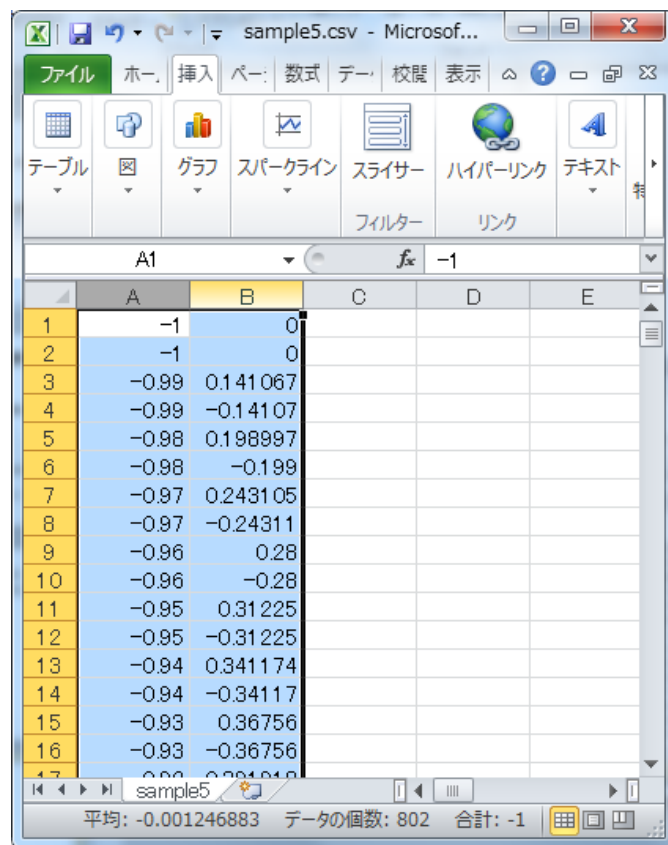
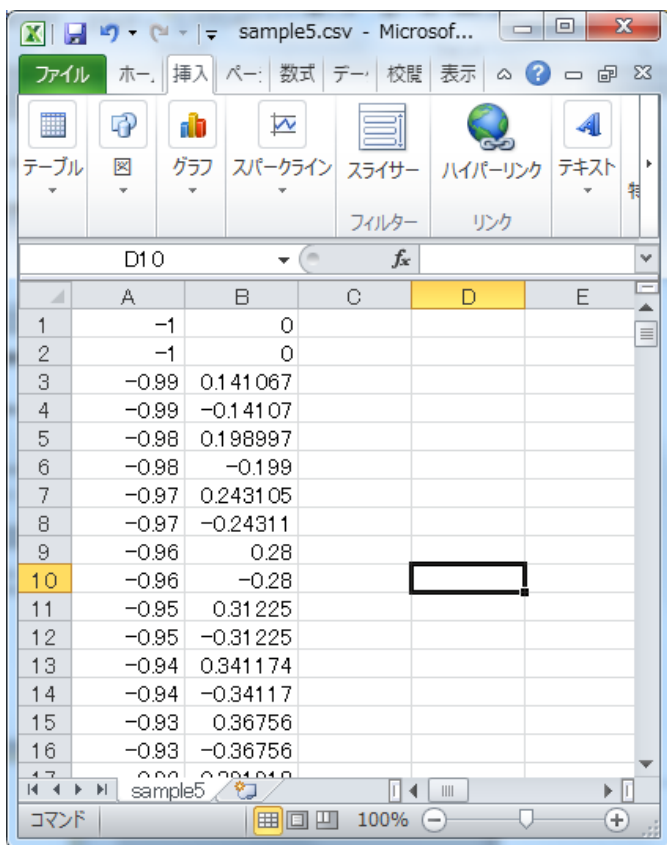
csvファイルにプログラムの実行結果を出力

```
#include<stdio.h>
#include<math.h>
double Circle(double x){
    return sqrt(1-x*x); //x^2+y^2=1 半径が1の円の関数をyについて解いた
}
int main(){
    double x;
    FILE *fp;
    fp=fopen("C:¥¥Users¥¥kagayaki¥¥Desktop¥¥sample5.csv","w");
    if(fp==NULL){
        printf("ファイルのオープンに失敗しました。");
        return 0;
    }
}
```

```
for(x=0.0;x<1.0;x+=0.01){  
    fprintf(fp,"%lf,%lf¥n",x,Circle(x));  
    fprintf(fp,"%lf,%lf¥n",x,-Circle(x));  
}  
fprintf(fp,"%lf,%lf¥n",1.0,0.0);  
fclose(fp);  
return 1;  
}
```

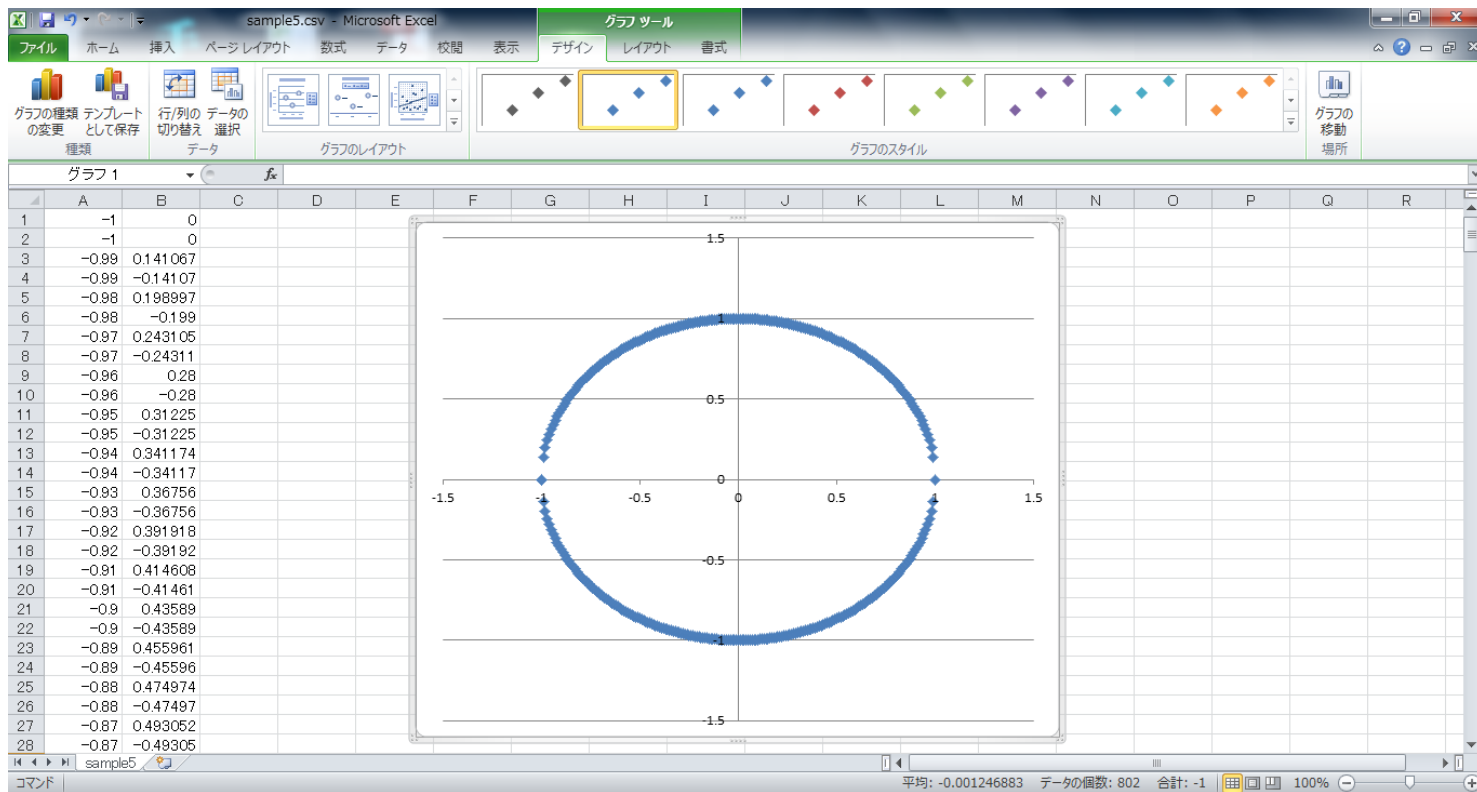
実行結果

- デスクトップにできたファイルをエクセルで開いて、A列とB列を選択し散布図(マーカーのみ)でグラフにすると



実行結果から

- 円のグラフができる。



問題2


- $f(x)=x^3$ を $-5 \leq x \leq 2$ の範囲で図示してください。
($x=-4.99$ の時は $f(4.99)$, $x=-4.98$ の時は...の様に)
ただし、double型の引数 x を持ち、double型の値を返す関数 $f(x)$ を定義すること。
(必要に応じて18~21ページを見てください)
- 変曲点が $-3, -2, 0$ のグラフができるはず

参考文献

- 参考文献:これならわかるC 入門の入門
– 坂下 夕里 著

問題1 解答例

モードを変更してどういう結果になるか見てみてください。
ファイルがない場合と、すでにある場合なども。



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

```
int main(){
```

```
FILE *fp;
```

```
fp=fopen("C:¥¥Users¥¥kagayaki¥¥Desktop¥¥sanple2.txt","w");
```

```
if(fp==NULL){
```

```
    printf("ファイルのオープンに失敗しました¥n");
```

```
    return 0;
```

```
}
```

```
fprintf(fp,"おさない じゅんぺい");
```

```
fclose(fp);
```

```
return 1;
```

```
}
```


問題2 解答例

```
#include<stdio.h>
#include<math.h>
double f(double x){
    return x*x*x+5*x*x+6*x;
}
int main(){
    double x;
    FILE *fp;
    fp=fopen("C:¥¥Users¥¥kagayaki¥¥Desktop¥¥sample6.csv","w");
    if(fp==NULL){printf("ファイルのオープンに失敗しました。");return 0;}
    for(x=-5.0;x<=2.0;x+=0.01)
        fprintf(fp,"%lf,%lf¥n",x,f(x));
    fclose(fp);
    return 1;
}
```

