4. 複素数関連

4.1 複素数を構造体で表現

複素数を以下のような構造体で表現することにします。

```
struct Complex{
    double Re;  /*  実数部(Real Part) */
    double Im;  /*  虚数部(Image Part) */
};
```

まず，値を設定する関数 setCom を用意します。また数値計算ではオイラーの公式で値を設定することが多いので，$e^{j\theta}$ における θ で値を設定するための expJCom を用意することにします。

```
struct Complex setCom(double Re, double Im){
    struct Complex C; C.Re = Re; C.Im = Im;   return C;
}
struct Complex expJCom(double TH){
    return setCom(cos(TH),sin(TH));
}
```

4.2 複素数を出力する

複素数の表現形式で文字列に変換したり，出力する関数です。確認用に，必要に応じて使って下さい。

```
int strCom(struct Complex A, char str[]){
  if(A.Im == 0) return(sprintf(str, "%lf", A.Re));
  if(A.Im <  0) return(sprintf(str, "%lf - %lf j", A.Re, fabs(A.Im)));
  else          return(sprintf(str, "%lf + %lf j", A.Re, A.Im));
}
void printCom(struct Complex A){
  if(A.Im <  0) printf("%8.5lf - %8.5lf j", A.Re, fabs(A.Im));
  else          printf("%8.5lf + %8.5lf j", A.Re, A.Im);
}
void fprintCom(FILE *fout,struct Complex A){
  if(A.Im <  0) fprintf(fout,"%8.5lf - %8.5lf j", A.Re, fabs(A.Im));
  else          fprintf(fout,"%8.5lf + %8.5lf j", A.Re, A.Im);
}
```

4.3 複素数の絶対値

複素数の絶対値は，複素数平面の原点からの距離です。そこで，

```
double absCom(struct Complex A){
       return sqrt(pow(A.Re, 2) + pow(A.Im, 2));
}
```

とすれば良さそうですが，大きな値のとき，指数部あふれが起きがちです。指数部あふれを少しでも減らすため，$\sqrt{x^2+y^2}$ の替りに