3. 分数関連

3.1 分数を構造体で表現

分数を以下のような構造体で表現することにします。

```
struct fraction{
    double nume;  /* 分子(numeration)  */
    double deno;  /* 分母(denominator) */
};
```

分母・分子は long でもよいのですが，何回も計算を繰り返すうちに桁あふれが生じることがあります。このように表現することで，実数／実数も分数として一般化することができます。値を交換する関数 swap，最大公約数を求める関数 GCD も double に変更します。

```
void swap(double* A, double* B){
    double t = *A; *A = *B; *B = t;
}
double DMod(double A, double B){ /* %演算子は整数用のため本関数追加 */
    double P = floor(A/B); return A - B * P;
}
double GCD(double A, double B){
    double W; A = fabs(A); B = fabs(B);
    if(A < B)  swap(&A, &B);
    while(B != 0){ W = DMod(A, B); A = B; B = W;}
    return A;
}
```

関数 scanf の書式文字列中にマッチングする文字を書いてもよいことになっていますので，分数は次のようにして入力できます。

```
struct fraction A;
scanf("%lf/%lf", &A.nume, &A.deno);
```

たとえば 12/15[ENTER]と入力すると，A.nume=12，A,deno=15 となります。さらに，分数の形の文字列を分数に変換するには，以下のようにすれば構いません。

```
sscanf(str, "%lf/%lf", &A.nume, &A.deno);  /*  str は char *  */
```

関数 printf や sprintf を使うことで，分数の表示や文字列への変換も簡単にできます。。

```
printf("%lf/%lf", A.nume, A.deno);

sprintf(str, "%lf/%lf", A.nume, A.deno);
```