2. 実数関連

2.1 機械イプシロン

機械イプシロン（machine ε）とは，実行するコンピュータの実数のうち最も小さな正の数です。「実数(1 + ε)-1 がとりうる最小の正の値」として定義されます。機械の実数の精度の指標や乱数発生の最小単位として用いられます。

```
double machineEP(){
    double E = 1, ED;
    do{ ED = E; E = E / 2;} while((E + 1) > 1);
    E = 1 + ED; return E - 1;
}
```

2.2 自前の Power

通常のコンピュータでは，機械語として乗算は用意されますが，べき乗の命令は皆無です。一方，言語処理系では一般にべき乗を計算する必要がありますので，これをソフトウェアで用意する必要があります。そこで乗算を繰り返しますが，指数の値が大きい時，繰返し回数が多くなりますので，工夫する必要があります。以下は $\log_2 N$ 回のオーダの乗算で済ます方法です。さらに，整数の 2 による除算はシフト演算で高速化します。

```
double myPower(double X,int N){
    double P=1.0;
    for(int abN = abs(N); abN > 0; abN >>=1, X *= X) if(abN & 1) P *= X;
    if(N < 0) return 1/P; else return P;
}
```

2.3 実数を連分数に変換

連分数（continued fraction）とは，次のように，分数の中にさらに分数を含む形の分数です。

$$x = a_0 + b_0/(a_1 + b_1/(a_2 + b_2/a_3))$$

特に分子がすべて 1 （上の例では b_0=b_1=b_2=1）のとき正則連分数（regular continued fraction）といいます。正則連分数は，通常，以下のように略記されます。

$$x = [\, a_0 \,;\, a_1 \,,\, a_2 \,,\, a_3]$$

さらに，級数と同様，無限に続く連分数も考えることができます。以下は，倍精度浮動少数点を正則連分数に変換し，略記法の値を配列に代入するプログラム例です。

```
int continuedFraction(float X,int N,int A[]){
    double B = floor(X); A[0]=(int)B;
    if(fabs(B - X) < 0.00000001) return 1;
    for(int i=1;i<N;i++){  X = 1 / (X - B); B = floor(X); A[i]=(int) B;}
    return 20;
}
```