

```

1 #include "basic_math.h"
2
3
4 float Sqrt(float x)
5 {
6     union
7     {
8         int i;
9         float x;
10    } u;
11    u.x = x;
12    u.i = (1<<29) + (u.i >> 1) - (1<<22);
13
14    // Two Babylonian Steps (simplified from:)
15    // u.x = 0.5f * (u.x + x/u.x);
16    // u.x = 0.5f * (u.x + x/u.x);
17    u.x =      u.x + x/u.x;
18    u.x = 0.25f*u.x + x/u.x;
19
20    return u.x;
21 }
22
23
24 //-----
-----
25 // Fast inverse square-root
26 // See: http://en.wikipedia.org/wiki/Fast\_inverse\_square\_root
27
28 float invSqrt(float x)
29 {
30     float halfx = 0.5f * x;
31     float y = x;
32     long i = *(long*)&y;
33     i = 0x5f3759df - (i>>1);
34     y = *(float*)&i;
35     y = y * (1.5f - (halfx * y * y));
36     return y;
37 }

```

1/√x の近似計算…ベクトルノルムの正規化などに使う。