竹

宁可食无肉，不可居无竹。无肉令人瘦。
无竹令人俗。
人瘦尚可肥，士俗不可医。

C语言中的数学

——C语言技术公开课第四讲

主讲人：朱兆祺

辗转相除法

- 又称为欧几里得算法，求解两个正整数最大公约数。
- 至今仍在使用的算法中历史最为悠久。
- 两个整数的最大公约数等于其中较小的数和两数的相除余数的最大公约数。……

- iNumberA = 135; iNumberB = 35;
- 135 / 35 = 3……20
- 35 / 20 = 1……15
- 20 / 15 = 1……5
- 15 / 5 = 3……0
-

```
int Gcd(int iNumberA, int iNumberB)
{
    int iNumberN = iNumberA % iNumberB;

    while (0 != iNumberN)
    {
        iNumberA = iNumberB;
        iNumberB = iNumberN;
        iNumberN = iNumberA % iNumberB;
    }

    return iNumberB;
}
```

N! 结果中有几位数

- N! = N * (N-1) *……* 2 * 1
- 假设N! = 10^A
- N!(1~9):A(0~1)
- N!(10~99):A(1~2)
- ……
- A = log(10)(N!)
- = log(10)(N * (N-1) *……* 2 * 1)
- = log(10)(N)+log(10)(N-1)+……+log(10)(1)

```c
int main(int argc, char *argv[])
{
    int iNumber , i = 0 ;
    double sum = 0 ;
    printf("请输入iNumber :");
    scanf( "%d" , &iNumber );
    for( i = 1 ; i < ( iNumber + 1 ) ; i++)
    {
        sum += log10(i) ;
    }
    printf(" N!有%d位 \n" , (int)sum  +  1  ) ;
    return 0;
}
```

实现开方函数

- C语言库中sqrt()：sqtr(x)，有理想精度、但没有理想速度。
- 卡马克快速平方根：速度最快开方算法，有理想速度、相对误差约为0.177585%。
- 牛顿迭代算法：有理想精度、数据越复杂速度越慢。

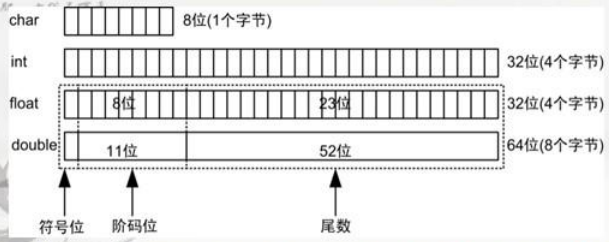
卡马克快速平方根

- 在3D图形编程中，经常要求平方根或平方根的倒数，例如：求向量的长度或将向量归一化。C数学函数库中的sqrt具有理想的精度，但对于3D游戏程式来说速度太慢。我们希望能够在保证足够的精度的同时，进一步提高速度。Carmack在QUAKE3（雷神之锤III ）中使用的卡马克快速平方根算法，它第一次在公众场合出现的时候，几乎震住了所有的人。据说该算法其实并不是Carmack发明的，它真正的作者是Nvidia的Gary Tarolli。

卡马克快速平方根

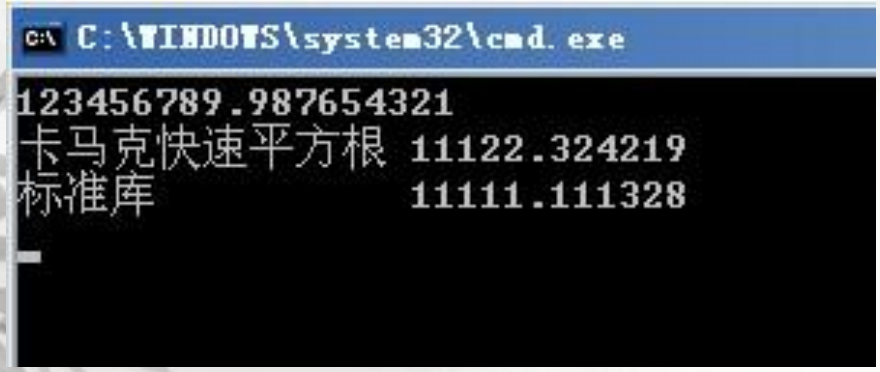
```
float InvSqrt2(float x)
{
    float xhalf = 0.5f*x;
    int i = *(int*)&x;          // get bits for floating VALUE
    i = 0x5f375a86- (i>>1); // gives initial guess y0
    x = *(float*)&i;            // convert bits BACK to float
    x = x*(1.5f-xhalf*x*x); // Newton
    return 1/x;
}
```

数据的存储

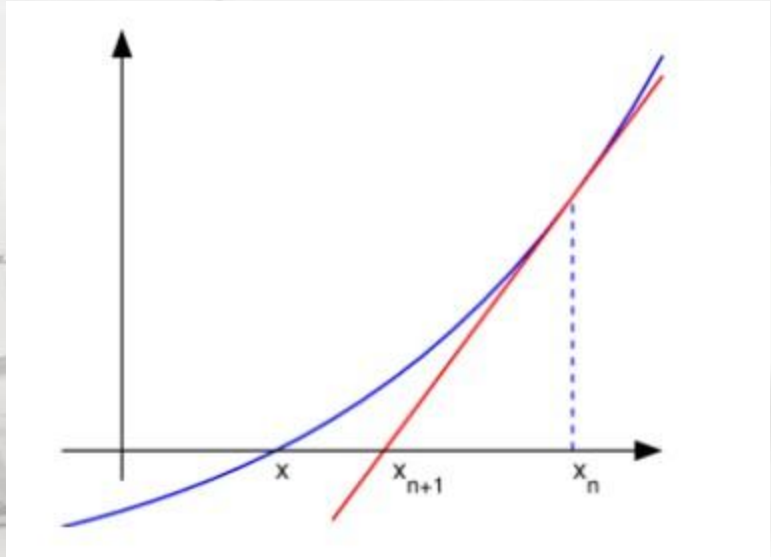


i=3.000000，那么3.000000(10进制) = 11(2进制) = 1.1*(2^1) (二进制)。
数据在电脑中存储都是二进制，这个应该都没有疑问。
数符为：0，阶码为：E – 127 = 1，
阶码为：E = 128 即为：1000 0000 (2进制)，
尾数为：100 0000 0000 0000 0000 0000 。
存储形式就是：0100 0000 0100 0000 0000 0000 0000 0000。
转换成16进制就是0x4040 0000。

卡马克快速平方根



牛顿迭代算法



$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{n}{x_k}\right), \quad \text{(迭代公式)}$$

牛顿迭代算法

```
float InvSqrt1(float x)
{
    float da = 1;
    float pre = x;
    if(x == 0){
        return 0;
    }else if(x < 0){
        printf("Error\n");
        return -1.0;
    }
    while(fabs(pre - da) > precision){   /*精度控制*/
        pre = da;
        da = (da+x/da)/(2);
    }
    return da;
}
```

卡尔曼滤波算法

- 卡尔曼滤波器是一个“optimal recursive data processing algorithm（最优化自回归数据处理算法）”。对于解决很大部分的问题，他是最优，效率最高甚至是最有用的。他的广泛应用已经超过30年，包括机器人导航，控制，传感器数据融合甚至在军事方面的雷达系统以及导弹追踪等等。近年来更被应用于计算机图像处理，例如头脸识别，图像分割，图像边缘检测等等。

卡尔曼滤波算法

- 最优状态估计（最优状态预测），卡尔曼滤波器是最优的信息处理器。

- 最小二乘法（又称最小平方法）是一种数学优化技术。它通过最小化误差的平方和寻找数据的最佳函数匹配。利用最小二乘法可以简便地求得未知的数据，并使得这些求得的数据与实际数据之间误差的平方和为最小。最小二乘法还可用于曲线拟合。

卡尔曼滤波算法

```
MatrixMul(F, tOpt.XPreOpt, X, ORDER, ORDER, ORDER);

MatrixCal(F, tCov.PPreOpt, Temp1, ORDER);
MatrixAdd(Temp1, Q, P, ORDER, ORDER);

MatrixCal(H, P, Temp1, ORDER);
MatrixAdd(Temp1, R, Temp1, ORDER, ORDER);
Gauss_Jordan(Temp1, ORDER);
MatrixTrans(H, Temp2, ORDER, ORDER);
MatrixMul(P, Temp2, Temp3, ORDER, ORDER, ORDER);
MatrixMul(Temp1, Temp3, K, ORDER, ORDER, ORDER);

MatrixMul(H, X, Temp1, ORDER, ORDER, ORDER);
MatrixMinus(Z, Temp1, Temp1, ORDER, ORDER);
MatrixMul(K, Temp1, Temp2, ORDER, ORDER, ORDER);
MatrixAdd(X, Temp2, tOpt.XNowOpt, ORDER, ORDER);

MatrixMul(K, H, Temp1, ORDER, ORDER, ORDER);
MatrixMinus(I, Temp1, Temp1, ORDER, ORDER);
MatrixMul(Temp1, P, tCov.PNowOpt, ORDER, ORDER, ORDER);
```

1.预估计

$X(k|k-1) = F(k,k-1)*X(k-1|k-1)$

2.计算预估计协方差矩阵

$P(k|k-1) = F(k,k-1)*P(k-1|k-1)*F(k,k-1)'+Q(k)$

$Q(k) = U(k)\times U(k)'$

3.计算卡尔曼增益矩阵

$Kg(k) = P(k|k-1)*H' / (H*P(k|k-1)*H' + R(k))$

$R(k) = N(k)\times N(k)'$

4.更新估计

$X(k|k) = X(k|k-1)+Kg(k)*(Z(k)-H*X(k|k-1))$

5.计算更新后估计协防差矩阵

$P(k|k) = (I-Kg(k)*H)*P(k|k-1)$

相关资料下载

雄鹰计划-卓越工程师炼成记：

http://bbs.elecfans.com/jishu_400775_1_1.html

朱兆祺ForARM步步为营官方群：110291944

- 《攻破C语言笔试与机试难点》C语言资料、雄鹰计划资料、嵌入式Linux实用教程、配套视频、Linux相关资料下载地址：
  http://pan.baidu.com/share/link?shareid=3562495290&uk=3996269986

- 视频配套OK6410、AM335X、STM32开发板，蓝牙4.0等淘宝店（明志电子科技），网址：http://mzkeji.taobao.com

竹

宁可食无肉，不可居无竹。无肉令人瘦。
无竹令人俗。
人瘦尚可肥，士俗不可医。

谢谢