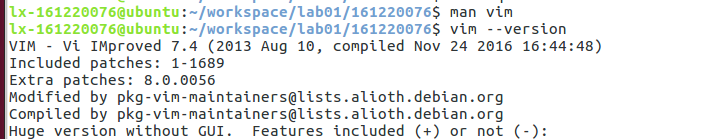
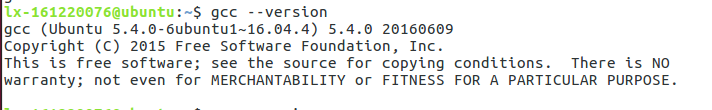
Vim版本：



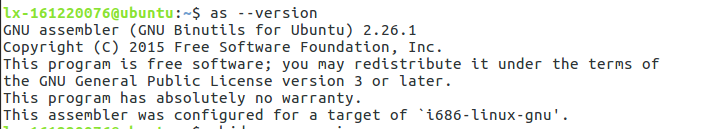
Git版本：

C:\Users\Administrator\Desktop\git version.png

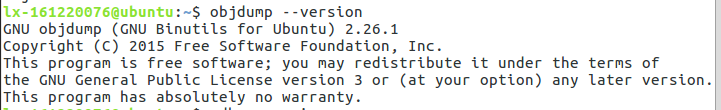
Gcc版本：



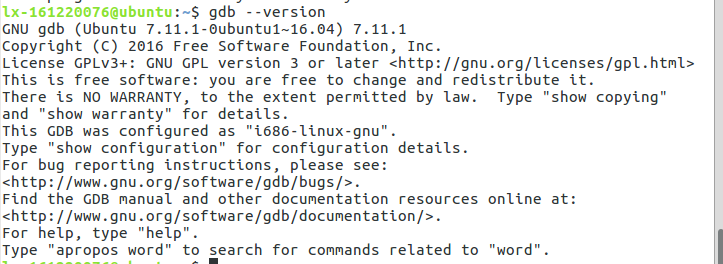
As版本;



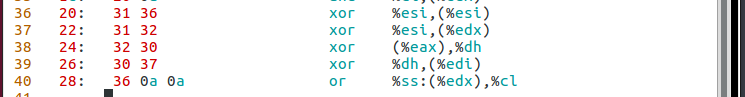
Objdump版本：



Gdb版本：

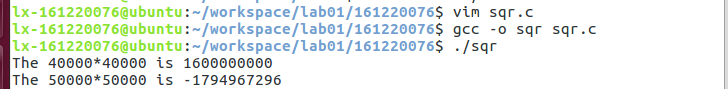


使用objdump的-D选项反汇编heart.o文件，学号的位置



（找到16进制的学号表示（161220076））

Sqr输出结果：



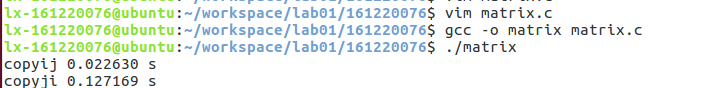
寻找在该程序中保证结果正确的最大整数值：

int占用4字节，32比特，数据范围为-2147483648~2147483647[-2^31~2^31-1],为了时数据结果不超出int类型表示的范围，数据不超过，为46340，即最大为46340.

修改程序，保证结果都正确:

将j的类型改为unsigned int类型。

矩阵运行时间比较:



两个函数中采用了两种不同的赋值顺序。一个二维数组（举例）a[100][100]，由100个一维数组a[0], a[1] … … , a[99],组成，在实际查找过程中，将某个一维数组读入缓存中，如果接下来的要查找的元素在缓存中，可以更为高效地查找到这个元素，提升缓存命中率。在一个一维数组从第一个元素查找到最后一个元素和从一个一维数组的某个位置跳到下一个一维数组的这个位置两种方式的缓存命中率不一样。从实际运行结果来看，前一种的运行时间代价更小。