

Vim 版本：

```
lx-161220076@ubuntu:~/workspace/lab01/161220076$ man vim
lx-161220076@ubuntu:~/workspace/lab01/161220076$ vim --version
VIM - Vi IMproved 7.4 (2013 Aug 10, compiled Nov 24 2016 16:44:48)
Included patches: 1-1689
Extra patches: 8.0.0056
Modified by pkg-vim-maintainers@lists.ubuntu.com
Compiled by pkg-vim-maintainers@lists.ubuntu.com
Huge version without GUI.  Features included (+) or not (-):
```

Git 版本：

```
lx-161220076@ubuntu:~$ git --version
git version 2.7.4
```

Gcc 版本：

```
lx-161220076@ubuntu:~$ gcc --version
gcc (Ubuntu 5.4.0-6ubuntu1~16.04.4) 5.4.0 20160609
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

As 版本：

```
lx-161220076@ubuntu:~$ as --version
GNU assembler (GNU Binutils for Ubuntu) 2.26.1
Copyright (C) 2015 Free Software Foundation, Inc.
This program is free software; you may redistribute it under the terms of
the GNU General Public License version 3 or later.
This program has absolutely no warranty.
This assembler was configured for a target of `i686-linux-gnu'.
```

Objdump 版本：

```
lx-161220076@ubuntu:~$ objdump --version
GNU objdump (GNU Binutils for Ubuntu) 2.26.1
Copyright (C) 2015 Free Software Foundation, Inc.
This program is free software; you may redistribute it under the terms of
the GNU General Public License version 3 or (at your option) any later version.
This program has absolutely no warranty.
```

Gdb 版本：

```
lx-161220076@ubuntu:~$ gdb --version
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.04) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
```

使用objdump的-D选项反汇编heart.o文件，学号的位置

```
36  20:  31 36          xor     %esi,%esi
37  22:  31 32          xor     %esi,%edx
38  24:  32 30          xor     (%eax),%dh
39  26:  30 37          xor     %dh,%edi
40  28:  36 0a 0a      or      %ss:(%edx),%cl
```

(找到 16 进制的学号表示 (161220076))

Sqr 输出结果：

```
lx-161220076@ubuntu:~/workspace/lab01/161220076$ vim sqr.c
lx-161220076@ubuntu:~/workspace/lab01/161220076$ gcc -o sqr sqr.c
lx-161220076@ubuntu:~/workspace/lab01/161220076$ ./sqr
The 40000*40000 is 1600000000
The 50000*50000 is -1794967296
```

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 $-2^{31} \sim 2^{31}-1$ ，为了时数据结果不超出 int 类型表示的范围，数据不超过 $\sqrt{2^{31}-1}$ ，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：

```
lx-161220076@ubuntu:~/workspace/lab01/161220076$ vim matrix.c
lx-161220076@ubuntu:~/workspace/lab01/161220076$ gcc -o matrix matrix.c
lx-161220076@ubuntu:~/workspace/lab01/161220076$ ./matrix
copyij 0.022630 s
copyji 0.127169 s
```

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 $a[0], a[1] \dots \dots, a[99]$ 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。