```
      long switch_prob(long x, long n)
      x in %rdi, n in %rsi
 1    0000000000400590 <switch_prob>:
 2      400590:  48 83 ee 3c              sub    $0x3c,%rsi
 3      400594:  48 83 fe 05              cmp    $0x5,%rsi
 4      400598:  77 29                    ja     4005c3 <switch_prob+0x33>
 5      40059a:  ff 24 f5 f8 06 40 00     jmpq   *0x4006f8(,%rsi,8)
 6      4005a1:  48 8d 04 fd 00 00 00     lea    0x0(,%rdi,8),%rax
 7      4005a8:  00
 8      4005a9:  c3                       retq
 9      4005aa:  48 89 f8                 mov    %rdi,%rax
10      4005ad:  48 c1 f8 03              sar    $0x3,%rax
11      4005b1:  c3                       retq
12      4005b2:  48 89 f8                 mov    %rdi,%rax
13      4005b5:  48 c1 e0 04              shl    $0x4,%rax
14      4005b9:  48 29 f8                 sub    %rdi,%rax
15      4005bc:  48 89 c7                 mov    %rax,%rdi
16      4005bf:  48 0f af ff              imul   %rdi,%rdi
17      4005c3:  48 8d 47 4b              lea    0x4b(%rdi),%rax
18      4005c7:  c3                       retq
```

图 3-53　家庭作业 3.63 的反汇编代码

**3.64　考虑下面的源代码，这里 R、S 和 T 都是用#define声明的常数：

```
1  long A[R][S][T];
2
3  long store_ele(long i, long j, long k, long *dest)
4  {
5      *dest = A[i][j][k];
6      return sizeof(A);
7  }
```

在编译这个程序中，GCC 产生下面的汇编代码：

```
long store_ele(long i, long j, long k, long *dest)
i in %rdi, j in %rsi, k in %rdx, dest in %rcx
1   store_ele:
2     leaq    (%rsi,%rsi,2), %rax
3     leaq    (%rsi,%rax,4), %rax
4     movq    %rdi, %rsi
5     salq    $6, %rsi
6     addq    %rsi, %rdi
7     addq    %rax, %rdi
8     addq    %rdi, %rdx
9     movq    A(,%rdx,8), %rax
10    movq    %rax, (%rcx)
11    movl    $3640, %eax
12    ret
```

A. 将等式(3.1)从二维扩展到三维，提供数组元素 A[i][j][k]的位置的公式。

B. 运用你的逆向工程技术，根据汇编代码，确定 R、S 和 T 的值。

*3.65　下面的代码转置一个 $M \times M$ 矩阵的元素，这里 M 是一个用#define定义的常数：

```
1  void transpose(long A[M][M]) {
2      long i, j;
3      for (i = 0; i < M; i++)
4          for (j = 0; j < i; j++) {
5              long t = A[i][j];
6              A[i][j] = A[j][i];
7              A[j][i] = t;
8          }
9  }
```