

 **练习题 3.40** 下面的 C 代码将定长数组的对角线上的元素设置为 val:

```
/* Set all diagonal elements to val */
void fix_set_diag(fix_matrix A, int val) {
    long i;
    for (i = 0; i < N; i++)
        A[i][i] = val;
}
```

当以优化等级 -O1 编译时, GCC 产生如下汇编代码:

```
1  fix_set_diag:
   void fix_set_diag(fix_matrix A, int val)
   A in %rdi, val in %rsi
2      movl    $0, %eax
3      .L13:
4          movl    %esi, (%rdi,%rax)
5          addq    $68, %rax
6          cmpq    $1088, %rax
7          jne     .L13
8          rep; ret
```

创建一个 C 代码程序 `fix_set_diag_opt`, 它使用类似于这段汇编代码中所使用的优化, 风格与图 3-37b 中的代码一致。使用含有参数 N 的表达式, 而不是整数常量, 使得如果重新定义了 N , 你的代码仍能够正确地工作。

3.8.5 变长数组

历史上, C 语言只支持大小在编译时就能确定的多维数组(对第一维可能有些例外)。程序员需要变长数组时不得不用 `malloc` 或 `calloc` 这样的函数为这些数组分配存储空间, 而且不得不显式地编码, 用行优先索引将多维数组映射到一维数组, 如公式(3.1)所示。ISO C99 引入了一种功能, 允许数组的维度是表达式, 在数组被分配的时候才计算出来。

在变长数组的 C 版本中, 我们可以将一个数组声明如下:

```
int A[expr1][expr2]
```

它可以作为一个局部变量, 也可以作为一个函数的参数, 然后在遇到这个声明的时候, 通过对表达式 `expr1` 和 `expr2` 求值来确定数组的维度。因此, 例如要访问 $n \times n$ 数组的元素 i, j , 我们可以写一个如下的函数:

```
int var_ele(long n, int A[n][n], long i, long j) {
    return A[i][j];
}
```

参数 n 必须在参数 `A[n][n]` 之前, 这样函数就可以在遇到这个数组的时候计算出数组的维度。

GCC 为这个引用函数产生的代码如下所示:

```
int var_ele(long n, int A[n][n], long i, long j)
n in %rdi, A in %rsi, i in %rdx, j in %rcx
1  var_ele:
2      imulq    %rdx, %rdi           Compute  $n \cdot i$ 
3      leaq     (%rsi,%rdi,4), %rax   Compute  $x_A + 4(n \cdot i)$ 
4      movl     (%rax,%rcx,4), %eax    Read from  $M[x_A + 4(n \cdot i) + 4j]$ 
5      ret
```