```
        .file   "matmult.c"
        .comm   Seed,4,4
        .comm   ArrayA,400,32
        .comm   ArrayB,400,32
        .comm   ResultArray,400,32
        .text
.globl main
        .type   main, @function
main:
3       pushl   %ebp
1       movl    %esp, %ebp
1       andl    $-16, %esp
1       subl    $16, %esp
3       call    InitSeed
3       movl    $ResultArray, 8(%esp)
3       movl    $ArrayB, 4(%esp)
3       movl    $ArrayA, (%esp)
3       call    Test
3       leave
5       ret
        .size   main, .-main
.globl InitSeed
        .type   InitSeed, @function
InitSeed:
3       pushl   %ebp
1       movl    %esp, %ebp
3       movl    $0, Seed
4       popl    %ebp
5       ret
        .size   InitSeed, .-InitSeed
.globl Test
        .type   Test, @function
Test:
3       pushl   %ebp
1       movl    %esp, %ebp
1       subl    $24, %esp
1       movl    8(%ebp), %eax
3       movl    %eax, (%esp)
3       call    Initialize
1       movl    12(%ebp), %eax
3       movl    %eax, (%esp)
3       call    Initialize
1       movl    16(%ebp), %eax
3       movl    %eax, 8(%esp)
1       movl    12(%ebp), %eax
3       movl    %eax, 4(%esp)
1       movl    8(%ebp), %eax
3       movl    %eax, (%esp)
3       call    Multiply
3       leave
5       ret
        .size   Test, .-Test
.globl Initialize
        .type   Initialize, @function
Initialize:
3       pushl   %ebp
1       movl    %esp, %ebp
3       pushl   %esi
3       pushl   %ebx
1       subl    $16, %esp
```

```
3   movl    $0, -16(%ebp)
4   jmp .L8
.L11:
3   movl    $0, -12(%ebp)
4   jmp .L9
.L10:
1   movl    -16(%ebp), %edx
1   movl    %edx, %eax
4   sall    $2, %eax
4   addl    %edx, %eax
4   sall    $3, %eax
3   movl    %eax, %ebx
4   addl    8(%ebp), %ebx
1   movl    -12(%ebp), %esi
3   call    RandomInteger
1   movl    %eax, (%ebx,%esi,4)
1   addl    $1, -12(%ebp)
.L9:
4   cmpl    $9, -12(%ebp)
1   jle .L10
1   addl    $1, -16(%ebp)
.L8:
4   cmpl    $9, -16(%ebp)
1   jle .L11
1   addl    $16, %esp
4   popl    %ebx
4   popl    %esi
4   popl    %ebp
5   ret
    .size   Initialize, .-Initialize
.globl RandomInteger
    .type   RandomInteger, @function
RandomInteger:
3   pushl   %ebp
1   movl    %esp, %ebp
3   pushl   %ebx
3   movl    Seed, %eax
1   imull   $133, %eax, %eax
2   leal    81(%eax), %ecx
3   movl    $271652039, %edx
1   movl    %ecx, %eax
1   imull   %edx
4   sarl    $9, %edx
1   movl    %ecx, %eax
4   sarl    $31, %eax
1   movl    %edx, %ebx
1   subl    %eax, %ebx
1   movl    %ebx, %eax
1   imull   $8095, %eax, %eax
1   movl    %ecx, %edx
1   subl    %eax, %edx
1   movl    %edx, %eax
1   movl    %eax, Seed
3   movl    Seed, %eax
4   popl    %ebx
4   popl    %ebp
5   ret
    .size   RandomInteger, .-RandomInteger
.globl Multiply
    .type   Multiply, @function
```

```
Multiply:
3   pushl   %ebp
1   movl    %esp, %ebp
3   pushl   %edi
3   pushl   %esi
3   pushl   %ebx
4   subl    $12, %esp
3   movl    $0, %ebx
4   jmp .L16
.L21:
3   movl    $0, %esi
4   jmp .L17
.L20:
1   movl    %ebx, %edx
1   movl    %edx, %eax
4   sall    $2, %eax
4   addl    %edx, %eax
4   sall    $3, %eax
1   addl    16(%ebp), %eax
1   movl    %esi, %edx
3   movl    $0, (%eax,%edx,4)
3   movl    $0, %edi
4   jmp .L18
.L19:
1   movl    %ebx, %edx
1   movl    %edx, %eax
4   sall    $2, %eax
4   addl    %edx, %eax
4   sall    $3, %eax
1   movl    16(%ebp), %edx
1   addl    %eax, %edx
1   movl    %edx, -24(%ebp)
1   movl    %esi, -20(%ebp)
1   movl    %ebx, %edx
1   movl    %edx, %eax
4   sall    $2, %eax
4   addl    %edx, %eax
4   sall    $3, %eax
1   addl    16(%ebp), %eax
1   movl    %esi, %edx
1   movl    (%eax,%edx,4), %eax
1   movl    %eax, -16(%ebp)
1   movl    %ebx, %edx
1   movl    %edx, %eax
4   sall    $2, %eax
1   addl    %edx, %eax
4   sall    $3, %eax
1   movl    %eax, %edx
1   addl    8(%ebp), %edx
1   movl    %edi, %eax
1   movl    (%edx,%eax,4), %ecx
1   movl    %edi, %edx
1   movl    %edx, %eax
4   sall    $2, %eax
1   addl    %edx, %eax
4   sall    $3, %eax
1   movl    %eax, %edx
1   addl    12(%ebp), %edx
1   movl    %esi, %eax
1   movl    (%edx,%eax,4), %eax
```

```
1   imull   %ecx, %eax
1   addl    -16(%ebp), %eax
1   movl    -20(%ebp), %ecx
1   movl    -24(%ebp), %edx
1   movl    %eax, (%edx,%ecx,4)
1   addl    $1, %edi
.L18:
4   cmpl    $9, %edi
1   jle .L19
1   addl    $1, %esi
.L17:
4   cmpl    $9, %esi
1   jle .L20
1   addl    $1, %ebx
.L16:
4   cmpl    $9, %ebx
1   jle .L21
1   addl    $12, %esp
4   popl    %ebx
4   popl    %esi
4   popl    %edi
4   popl    %ebp
5   ret
    .size   Multiply, .-Multiply
    .ident  "GCC: (Debian 4.4.5-8) 4.4.5"
    .section    .note.GNU-stack,"",@progbits
```