实验报告

总体实现思路

该语义实验是基于语法实验构造的语法树完成的。首先基于语法分析的程序，在C++程序中构建语法树。之后分析语法树，在每一步归约动作后，添加相应的汇编代码逻辑。

汇编代码存储

对于每一步生成的汇编代码，我利用一个全局的vector存储汇编代码语句，并在每一步归约动作处理时将需要添加的代码逻辑添加到该vector中。

变量维护

变量存储

对于变量存储，我将源程序中定义的变量存储在栈中，将定义的全局变量定义在数据段，并利用寄存器%r8-%r15存储表达式中的临时变量。对于寄存器的分配，我利用一个哈希表维护空闲的寄存器。

变量信息

对于每一个变量，我定义如下结构体：

```
struct Variable {
    string name; // 变量名称
    int offset = 0, size, val; // 变量偏移量，大小，数值（常量）
    bool isGlobal = false, isConst; // 是否为全局变量，是否为常量
    vector<int> arr = vector<int>(); // 数组每一维大小
    string addrReg; // 存储偏移量的寄存器
    string trueName; // 该变量对应的真实变量名（临时变量）
};
```

同时，对于临时变量，需要一个<string, string>的哈希表，用于存储变量名与其对应的寄存器，同时有一个全局的变量tmp对每一个临时变量唯一的命名。

符号表

对于符号表，我用一个栈来维护整个符号表。当进入一个Block时，向栈中添加一个符号表，并在离开Block时将其弹出。每一个符号表，我用一个<string, Variable>的哈希表维护，通过变量名索引变量信息。

函数调用

对于函数调用，我首先将原先函数中调用的寄存器保存在栈中，之后保存%rbp，最后将%rdi与%rsi中传的参数保存到栈中作为函数中的变量调用。对于每一个函数，我开启一个新的哈希表存储空闲的寄存器。在函数返回时，我将保存的寄存器的值返回，并将存储空闲寄存器的哈希表弹出。

Break与Continue的实现

对于break与continue，我用一个栈来记录当前循环所需要跳转的位置，每次处理doWhile块时，将nextNum与backNum存储在栈中，若在循环中调用了break和continue，则直接跳转到该位置。并在每一个处理完doWhile块时将栈弹出。

实验结果

1.sy

```
.LC0:
    .string "%lld"
.LC1:
    .string "%lld\n"
    .globl  main
main:
    pushq   %rbp
    pushq   %rbx
    pushq   %rcx
    pushq   %r8
    pushq   %r9
    pushq   %r10
    pushq   %r11
    pushq   %r12
    pushq   %r13
    pushq   %r14
    pushq   %r15
    movq    %rsp, %rbp
    subq    $32, %rsp
    subq    $16, %rsp
    subq    $16, %rsp
    movq    -8(%rbp), %r8
    movq    $8, %rcx
    subq    $16, %rsp
    neg     %rcx
    leaq    (%rbp, %rcx), %rax
    neg     %rcx
    movq    %rax, %rsi
    leaq    .LC0(%rip), %rdi
    call    __isoc99_scanf@PLT
    addq    $16, %rsp
    movq    -24(%rbp), %r8
    movq    $24, %rcx
    subq    $16, %rsp
    neg     %rcx
    leaq    (%rbp, %rcx), %rax
    neg     %rcx
    movq    %rax, %rsi
    leaq    .LC0(%rip), %rdi
    call    __isoc99_scanf@PLT
    addq    $16, %rsp
    movq    -8(%rbp), %r8
    movq    $8, %rcx
    movq    -24(%rbp), %r9
    movq    $24, %rcx
    addq    %r9, %r8
    subq    $16, %rsp
    movq    %r8, -40(%rbp)
    movq    -40(%rbp), %r8
    movq    $40, %rcx
    subq    $16, %rsp
```

```
    movq    %r8, %rsi
    leaq    .LC1(%rip), %rdi
    call    printf@PLT
    addq    $16, %rsp
    movq    %rbp, %rsp
    popq     %r15
    popq     %r14
    popq     %r13
    popq     %r12
    popq     %r11
    popq     %r10
    popq     %r9
    popq     %r8
    popq    %rcx
    popq    %rbx
    popq    %rbp
    ret
```

2.sy

```
.LC0:
    .string "%lld"
.LC1:
    .string "%lld\n"
N:
    .long   3
    .comm   a,160,32
    .globl  f
f:
    pushq   %rbp
    pushq   %rbx
    pushq   %rcx
    pushq   %r8
    pushq   %r9
    pushq   %r10
    pushq   %r11
    pushq   %r12
    pushq   %r13
    pushq   %r14
    pushq   %r15
    movq    %rsp, %rbp
    subq    $32, %rsp
    subq    $16, %rsp
    movq     %rdi, -8(%rbp)
    movq    -8(%rbp), %r8
    movq    $8, %rcx
    movq    $1, %r9
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r9, %r8
    cmove   %rcx, %rdx
    movq    %rdx, %r8
    movq    $0, %rcx
    cmpq    %r8, %rcx
    je  .L1
```

```
    movq    $1, %r9
    movq    %r9, %rax
    movq    %rbp, %rsp
    popq     %r15
    popq     %r14
    popq     %r13
    popq     %r12
    popq     %r11
    popq     %r10
    popq     %r9
    popq     %r8
    popq    %rcx
    popq    %rbx
    popq    %rbp
    ret
.L1:
    movq    $1, %r9
    subq    $16, %rsp
    movq    %r9, -24(%rbp)
    movq    -24(%rbp), %r9
    movq    $24, %rcx
    movq    -8(%rbp), %r10
    movq    $8, %rcx
    movq    -8(%rbp), %r11
    movq    $8, %rcx
    movq    $1, %r12
    subq    %r12, %r11
    movq    %r11, %rdi
    call    f
    imulq   %rax, %r10
    neg     %rcx
    movq    %r10, -24(%rbp)
    neg     %rcx
    movq    -24(%rbp), %r9
    movq    $24, %rcx
    movq    %r9, %rax
    movq    %rbp, %rsp
    popq     %r15
    popq     %r14
    popq     %r13
    popq     %r12
    popq     %r11
    popq     %r10
    popq     %r9
    popq     %r8
    popq    %rcx
    popq    %rbx
    popq    %rbp
    ret
    movq    %rbp, %rsp
    popq     %r15
    popq     %r14
    popq     %r13
    popq     %r12
    popq     %r11
    popq     %r10
```

```
    popq     %r9
    popq     %r8
    popq    %rcx
    popq    %rbx
    popq    %rbp
    ret
    .globl  main
main:
    pushq   %rbp
    pushq   %rbx
    pushq   %rcx
    pushq   %r8
    pushq   %r9
    pushq   %r10
    pushq   %r11
    pushq   %r12
    pushq   %r13
    pushq   %r14
    pushq   %r15
    movq    %rsp, %rbp
    subq    $32, %rsp
    movq    $10, %r8
    movq    %r8, %rdi
    call    f
    subq    $16, %rsp
    movq    %rax, -8(%rbp)
    movq    -8(%rbp), %r8
    movq    $8, %rcx
    movq    $3628800, %r9
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r9, %r8
    cmove   %rcx, %rdx
    movq    %rdx, %r8
    movq    -8(%rbp), %r9
    movq    $8, %rcx
    orq     %r9, %r8
    movq    $0, %rcx
    cmpq    %r8, %rcx
    je  .L2
    movq    $2333, %r9
    subq    $16, %rsp
    movq    %r9, %rsi
    leaq    .LC1(%rip), %rdi
    call    printf@PLT
    addq    $16, %rsp
    movq    -8(%rbp), %r9
    movq    $8, %rcx
    movq    $0, %rax
    movq    $1, %rcx
    cmpq    %r9, %rax
    cmove   %rcx, %rax
    movq    %rax, %r9
    movq    $0, %rcx
    cmpq    %r9, %rcx
    je  .L4
```

```
    movq    -8(%rbp), %r10
    movq    $8, %rcx
    movq    -8(%rbp), %r11
    movq    $8, %rcx
    movq    $10, %r12
    subq    %r12, %r11
    neg     %rcx
    movq    %r11, -8(%rbp)
    neg     %rcx
    jmp .L5
.L4:
    movq    -8(%rbp), %r10
    movq    $8, %rcx
    movq    -8(%rbp), %r11
    movq    $8, %rcx
    movq    $10, %r12
    addq    %r12, %r11
    neg     %rcx
    movq    %r11, -8(%rbp)
    neg     %rcx
.L5:
    jmp .L3
.L2:
    movq    -8(%rbp), %r10
    movq    $8, %rcx
    subq    $16, %rsp
    movq    %r10, %rsi
    leaq    .LC1(%rip), %rdi
    call    printf@PLT
    addq    $16, %rsp
.L3:
    movq    -8(%rbp), %r10
    movq    $8, %rcx
    subq    $16, %rsp
    movq    %r10, %rsi
    leaq    .LC1(%rip), %rdi
    call    printf@PLT
    addq    $16, %rsp
    movq    $0, %r10
    subq    $16, %rsp
    movq    %r10, -24(%rbp)
    movq    $0, %r10
    subq    $16, %rsp
    movq    %r10, -40(%rbp)
.L6:
    movq    -24(%rbp), %r10
    movq    $24, %rcx
    movq    $3, %r11
    movq    $0, %rcx
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r11, %r10
    cmovle  %rcx, %rdx
    movq    %rdx, %r10
    movq    $0, %rcx
    cmpq    %r10, %rcx
```

```
    je  .L7
    movq    -24(%rbp), %r10
    movq    $24, %rcx
    movq    $1, %r11
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r11, %r10
    cmove   %rcx, %rdx
    movq    %rdx, %r10
    movq    $0, %rcx
    cmpq    %r10, %rcx
    je  .L8
    movq    -24(%rbp), %r11
    movq    $24, %rcx
    movq    -24(%rbp), %r12
    movq    $24, %rcx
    movq    $1, %r13
    addq    %r13, %r12
    neg     %rcx
    movq    %r12, -24(%rbp)
    neg     %rcx
    jmp .L6
.L8:
    movq    -24(%rbp), %r11
    movq    $24, %rcx
    movq    $3, %r12
    movq    $0, %rcx
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r12, %r11
    cmove   %rcx, %rdx
    movq    %rdx, %r11
    movq    $0, %rcx
    cmpq    %r11, %rcx
    je  .L9
    jmp .L7
.L9:
    movq    -24(%rbp), %r12
    movq    $24, %rcx
    subq    $16, %rsp
    movq    %r12, %rsi
    leaq    .LC1(%rip), %rdi
    call    printf@PLT
    addq    $16, %rsp
    movq    -24(%rbp), %r12
    movq    $24, %rcx
    movq    -24(%rbp), %r13
    movq    $24, %rcx
    movq    $1, %r14
    addq    %r14, %r13
    neg     %rcx
    movq    %r13, -24(%rbp)
    neg     %rcx
    jmp .L6
.L7:
    movq    $3, %r12
```

```
    movq    $0, %rcx
    movq    $3, %r12
    movq    $0, %rcx
    movq    $2, %r13
    imulq   %r13, %r12
    subq    $160, %rsp
    movq    -24(%rbp), %r12
    movq    $24, %rcx
    movq    $1, %r13
    neg     %rcx
    movq    %r13, -24(%rbp)
    neg     %rcx
    movq    -40(%rbp), %r12
    movq    $40, %rcx
    movq    -24(%rbp), %r13
    movq    $24, %rcx
    movq    -24(%rbp), %r14
    movq    $24, %rcx
    addq    %r14, %r13
    neg     %rcx
    movq    %r13, -40(%rbp)
    neg     %rcx
    movq    $0, %r12
    movq    $0, %r13
    movq    $6, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    $0, %r13
    movq    $1, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    $8 , %rcx
    imulq   %rcx, %r12
    addq    $56, %r12
    neg     %r12
    movq    (%rbp, %r12), %r13
    neg     %r12
    movq    -24(%rbp), %r14
    movq    $24, %rcx
    movq    -40(%rbp), %r15
    movq    $40, %rcx
    movq    $2, %rbx
    imulq   %rbx, %r15
    addq    %r15, %r14
    neg     %r12
    movq    %r14, (%rbp, %r12)
    neg     %r12
    movq    $0, %r12
    movq    -24(%rbp), %r13
    movq    $24, %rcx
    movq    $6, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    -40(%rbp), %r13
    movq    $40, %rcx
    movq    $1, %rcx
```

```
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    $8 , %rcx
    imulq   %rcx, %r12
    addq    $56, %r12
    neg     %r12
    movq    (%rbp, %r12), %r13
    neg     %r12
    movq    $3, %r14
    neg     %r12
    movq    %r14, (%rbp, %r12)
    neg     %r12
    movq    $0, %r12
    movq    $0, %r13
    movq    $6, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    $0, %r13
    movq    $1, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    $8 , %rcx
    imulq   %rcx, %r12
    addq    $56, %r12
    neg     %r12
    movq    (%rbp, %r12), %r13
    neg     %r12
    subq    $16, %rsp
    movq    %r13, %rsi
    leaq    .LC1(%rip), %rdi
    call    printf@PLT
    addq    $16, %rsp
    movq    $0, %r12
    movq    -24(%rbp), %r13
    movq    $24, %rcx
    movq    $6, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    -40(%rbp), %r13
    movq    $40, %rcx
    movq    $1, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    $8 , %rcx
    imulq   %rcx, %r12
    addq    $56, %r12
    neg     %r12
    movq    (%rbp, %r12), %r13
    neg     %r12
    subq    $16, %rsp
    movq    %r13, %rsi
    leaq    .LC1(%rip), %rdi
    call    printf@PLT
    addq    $16, %rsp
    movq    $0, %r12
    movq    -24(%rbp), %r13
```

```
    movq    $24, %rcx
    movq    $6, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    -40(%rbp), %r13
    movq    $40, %rcx
    movq    $1, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    $8 , %rcx
    imulq   %rcx, %r12
    addq    $56, %r12
    neg     %r12
    movq    (%rbp, %r12), %r13
    neg     %r12
    movq    $0, %r14
    movq    $0, %r15
    movq    $6, %rcx
    imulq   %rcx, %r15
    addq    %r15, %r14
    movq    $0, %r15
    movq    $1, %rcx
    imulq   %rcx, %r15
    addq    %r15, %r14
    movq    $8 , %rcx
    imulq   %rcx, %r14
    addq    $56, %r14
    neg     %r14
    movq    (%rbp, %r14), %r15
    neg     %r14
    addq    %r15, %r13
    subq    $16, %rsp
    movq    %r13, -216(%rbp)
    movq    -216(%rbp), %r12
    movq    $216, %rcx
    subq    $16, %rsp
    movq    %r12, %rsi
    leaq    .LC1(%rip), %rdi
    call    printf@PLT
    addq    $16, %rsp
    movq    $0, %r12
    movq    $0, %r13
    movq    $6, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    $0, %r13
    movq    $1, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    leaq    a(%rip), %rax
    movq    $8 , %rcx
    imulq   %rcx, %r12
    addq    $0, %r12
    addq    %r12, %rax
    movq    (%rax), %r13
    movq    $5, %r15
```

```
    leaq    a(%rip), %rax
    addq    %r12, %rax
    movq    %r15, (%rax)
    movq    $0, %r12
    movq    $1, %r13
    movq    $6, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    $1, %r13
    movq    $1, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    leaq    a(%rip), %rax
    movq    $8 , %rcx
    imulq   %rcx, %r12
    addq    $0, %r12
    addq    %r12, %rax
    movq    (%rax), %r13
    movq    $2, %r15
    leaq    a(%rip), %rax
    addq    %r12, %rax
    movq    %r15, (%rax)
    movq    $0, %r12
    movq    $0, %r13
    movq    $6, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    $0, %r13
    movq    $1, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    leaq    a(%rip), %rax
    movq    $8 , %rcx
    imulq   %rcx, %r12
    addq    $0, %r12
    addq    %r12, %rax
    movq    (%rax), %r13
    subq    $16, %rsp
    movq    %r13, %rsi
    leaq    .LC1(%rip), %rdi
    call    printf@PLT
    addq    $16, %rsp
    movq    $0, %r12
    movq    $1, %r13
    movq    $6, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    $1, %r13
    movq    $1, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    leaq    a(%rip), %rax
    movq    $8 , %rcx
    imulq   %rcx, %r12
    addq    $0, %r12
    addq    %r12, %rax
```

```
    movq    (%rax), %r13
    subq    $16, %rsp
    movq    %r13, %rsi
    leaq    .LC1(%rip), %rdi
    call    printf@PLT
    addq    $16, %rsp
    movq    $0, %r12
    movq    $1, %r13
    movq    $6, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    $1, %r13
    movq    $1, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    leaq    a(%rip), %rax
    movq    $8 , %rcx
    imulq   %rcx, %r12
    addq    $0, %r12
    addq    %r12, %rax
    movq    (%rax), %r13
    subq    $16, %rsp
    leaq    a(%rip), %rax
    addq    %r12, %rax
    movq    %rax, %rsi
    leaq    .LC0(%rip), %rdi
    call    __isoc99_scanf@PLT
    addq    $16, %rsp
    movq    $0, %r12
    movq    $1, %r13
    movq    $6, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    movq    $1, %r13
    movq    $1, %rcx
    imulq   %rcx, %r13
    addq    %r13, %r12
    leaq    a(%rip), %rax
    movq    $8 , %rcx
    imulq   %rcx, %r12
    addq    $0, %r12
    addq    %r12, %rax
    movq    (%rax), %r13
    subq    $16, %rsp
    movq    %r13, %rsi
    leaq    .LC1(%rip), %rdi
    call    printf@PLT
    addq    $16, %rsp
    movq    $0, %r12
    movq    %r12, %rax
    movq    %rbp, %rsp
    popq     %r15
    popq     %r14
    popq     %r13
    popq     %r12
    popq     %r11
```

```
    popq     %r10
    popq     %r9
    popq     %r8
    popq    %rcx
    popq    %rbx
    popq    %rbp
    ret
    movq    %rbp, %rsp
    popq     %r15
    popq     %r14
    popq     %r13
    popq     %r12
    popq     %r11
    popq     %r10
    popq     %r9
    popq     %r8
    popq    %rcx
    popq    %rbx
    popq    %rbp
    ret
```

3.sy

```
.LC0:
    .string "%lld"
.LC1:
    .string "%lld\n"
N:
    .long   10
    .comm   a,96,32
    .comm   b,96,32
    .comm   c,96,32
    .globl  main
main:
    pushq   %rbp
    pushq   %rbx
    pushq   %rcx
    pushq   %r8
    pushq   %r9
    pushq   %r10
    pushq   %r11
    pushq   %r12
    pushq   %r13
    pushq   %r14
    pushq   %r15
    movq    %rsp, %rbp
    subq    $32, %rsp
    subq    $16, %rsp
    subq    $16, %rsp
    movq    -8(%rbp), %r8
    movq    $8, %rcx
    subq    $16, %rsp
    neg     %rcx
    leaq    (%rbp, %rcx), %rax
    neg     %rcx
```

```
    movq    %rax, %rsi
    leaq    .LC0(%rip), %rdi
    call    __isoc99_scanf@PLT
    addq    $16, %rsp
    movq    -24(%rbp), %r8
    movq    $24, %rcx
    subq    $16, %rsp
    neg     %rcx
    leaq    (%rbp, %rcx), %rax
    neg     %rcx
    movq    %rax, %rsi
    leaq    .LC0(%rip), %rdi
    call    __isoc99_scanf@PLT
    addq    $16, %rsp
    movq    $0, %r8
    subq    $16, %rsp
    movq    %r8, -40(%rbp)
.L1:
    movq    -40(%rbp), %r8
    movq    $40, %rcx
    movq    -8(%rbp), %r9
    movq    $8, %rcx
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r9, %r8
    cmovl   %rcx, %rdx
    movq    %rdx, %r8
    movq    $0, %rcx
    cmpq    %r8, %rcx
    je  .L2
    movq    $0, %r8
    movq    -40(%rbp), %r9
    movq    $40, %rcx
    movq    $1, %rcx
    imulq   %rcx, %r9
    addq    %r9, %r8
    leaq    a(%rip), %rax
    movq    $8 , %rcx
    imulq   %rcx, %r8
    addq    $0, %r8
    addq    %r8, %rax
    movq    (%rax), %r9
    subq    $16, %rsp
    leaq    a(%rip), %rax
    addq    %r8, %rax
    movq    %rax, %rsi
    leaq    .LC0(%rip), %rdi
    call    __isoc99_scanf@PLT
    addq    $16, %rsp
    movq    -40(%rbp), %r8
    movq    $40, %rcx
    movq    -40(%rbp), %r9
    movq    $40, %rcx
    movq    $1, %r10
    addq    %r10, %r9
    neg     %rcx
```

```
    movq    %r9, -40(%rbp)
    neg     %rcx
    jmp .L1
.L2:
    movq    -40(%rbp), %r8
    movq    $40, %rcx
    movq    $0, %r9
    neg     %rcx
    movq    %r9, -40(%rbp)
    neg     %rcx
.L3:
    movq    -40(%rbp), %r8
    movq    $40, %rcx
    movq    -24(%rbp), %r9
    movq    $24, %rcx
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r9, %r8
    cmovl   %rcx, %rdx
    movq    %rdx, %r8
    movq    $0, %rcx
    cmpq    %r8, %rcx
    je  .L4
    movq    $0, %r8
    movq    -40(%rbp), %r9
    movq    $40, %rcx
    movq    $1, %rcx
    imulq   %rcx, %r9
    addq    %r9, %r8
    leaq    b(%rip), %rax
    movq    $8 , %rcx
    imulq   %rcx, %r8
    addq    $0, %r8
    addq    %r8, %rax
    movq    (%rax), %r9
    subq    $16, %rsp
    leaq    b(%rip), %rax
    addq    %r8, %rax
    movq    %rax, %rsi
    leaq    .LC0(%rip), %rdi
    call    __isoc99_scanf@PLT
    addq    $16, %rsp
    movq    -40(%rbp), %r8
    movq    $40, %rcx
    movq    -40(%rbp), %r9
    movq    $40, %rcx
    movq    $1, %r10
    addq    %r10, %r9
    neg     %rcx
    movq    %r9, -40(%rbp)
    neg     %rcx
    jmp .L3
.L4:
    movq    $0, %r8
    subq    $16, %rsp
    movq    %r8, -56(%rbp)
```

```
    movq    -40(%rbp), %r8
    movq    $40, %rcx
    movq    $0, %r9
    neg     %rcx
    movq    %r9, -40(%rbp)
    neg     %rcx
.L5:
    movq    -40(%rbp), %r8
    movq    $40, %rcx
    movq    -8(%rbp), %r9
    movq    $8, %rcx
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r9, %r8
    cmovl   %rcx, %rdx
    movq    %rdx, %r8
    movq    $0, %rcx
    cmpq    %r8, %rcx
    je  .L6
    movq    -56(%rbp), %r8
    movq    $56, %rcx
    movq    $0, %r9
    neg     %rcx
    movq    %r9, -56(%rbp)
    neg     %rcx
.L7:
    movq    -56(%rbp), %r8
    movq    $56, %rcx
    movq    -24(%rbp), %r9
    movq    $24, %rcx
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r9, %r8
    cmovl   %rcx, %rdx
    movq    %rdx, %r8
    movq    $0, %rcx
    cmpq    %r8, %rcx
    je  .L8
    movq    $0, %r8
    movq    -40(%rbp), %r9
    movq    $40, %rcx
    movq    $1, %rcx
    imulq   %rcx, %r9
    addq    %r9, %r8
    leaq    a(%rip), %rax
    movq    $8 , %rcx
    imulq   %rcx, %r8
    addq    $0, %r8
    addq    %r8, %rax
    movq    (%rax), %r9
    movq    $0, %r10
    movq    -56(%rbp), %r11
    movq    $56, %rcx
    movq    $1, %rcx
    imulq   %rcx, %r11
    addq    %r11, %r10
```

```
    leaq    b(%rip), %rax
    movq    $8 , %rcx
    imulq   %rcx, %r10
    addq    $0, %r10
    addq    %r10, %rax
    movq    (%rax), %r11
    imulq   %r11, %r9
    subq    $16, %rsp
    movq    %r9, -72(%rbp)
    movq    $0, %r8
    movq    -40(%rbp), %r9
    movq    $40, %rcx
    movq    -56(%rbp), %r11
    movq    $56, %rcx
    addq    %r11, %r9
    movq    $1, %rcx
    imulq   %rcx, %r9
    addq    %r9, %r8
    leaq    c(%rip), %rax
    movq    $8 , %rcx
    imulq   %rcx, %r8
    addq    $0, %r8
    addq    %r8, %rax
    movq    (%rax), %r9
    movq    $0, %r11
    movq    -40(%rbp), %r12
    movq    $40, %rcx
    movq    -56(%rbp), %r13
    movq    $56, %rcx
    addq    %r13, %r12
    movq    $1, %rcx
    imulq   %rcx, %r12
    addq    %r12, %r11
    leaq    c(%rip), %rax
    movq    $8 , %rcx
    imulq   %rcx, %r11
    addq    $0, %r11
    addq    %r11, %rax
    movq    (%rax), %r12
    movq    -72(%rbp), %r13
    movq    $72, %rcx
    addq    %r13, %r12
    leaq    c(%rip), %rax
    addq    %r8, %rax
    movq    %r12, (%rax)
    movq    -56(%rbp), %r8
    movq    $56, %rcx
    movq    -56(%rbp), %r9
    movq    $56, %rcx
    movq    $1, %r11
    addq    %r11, %r9
    neg     %rcx
    movq    %r9, -56(%rbp)
    neg     %rcx
    jmp .L7
.L8:
```

```
    movq    -40(%rbp), %r8
    movq    $40, %rcx
    movq    -40(%rbp), %r9
    movq    $40, %rcx
    movq    $1, %r11
    addq    %r11, %r9
    neg     %rcx
    movq    %r9, -40(%rbp)
    neg     %rcx
    jmp .L5
.L6:
    movq    -40(%rbp), %r8
    movq    $40, %rcx
    movq    $0, %r9
    neg     %rcx
    movq    %r9, -40(%rbp)
    neg     %rcx
.L9:
    movq    -40(%rbp), %r8
    movq    $40, %rcx
    movq    -8(%rbp), %r9
    movq    $8, %rcx
    movq    -24(%rbp), %r11
    movq    $24, %rcx
    addq    %r11, %r9
    movq    $1, %r11
    subq    %r11, %r9
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r9, %r8
    cmovl   %rcx, %rdx
    movq    %rdx, %r8
    movq    $0, %rcx
    cmpq    %r8, %rcx
    je  .L10
    movq    $0, %r8
    movq    -40(%rbp), %r9
    movq    $40, %rcx
    movq    $1, %rcx
    imulq   %rcx, %r9
    addq    %r9, %r8
    leaq    c(%rip), %rax
    movq    $8 , %rcx
    imulq   %rcx, %r8
    addq    $0, %r8
    addq    %r8, %rax
    movq    (%rax), %r9
    subq    $16, %rsp
    movq    %r9, %rsi
    leaq    .LC1(%rip), %rdi
    call    printf@PLT
    addq    $16, %rsp
    movq    -40(%rbp), %r8
    movq    $40, %rcx
    movq    -40(%rbp), %r9
    movq    $40, %rcx
```

```
    movq    $1, %r11
    addq    %r11, %r9
    neg     %rcx
    movq    %r9, -40(%rbp)
    neg     %rcx
    jmp .L9
.L10:
    movq    $0, %r8
    movq    %r8, %rax
    movq    %rbp, %rsp
    popq     %r15
    popq     %r14
    popq     %r13
    popq     %r12
    popq     %r11
    popq     %r10
    popq     %r9
    popq     %r8
    popq    %rcx
    popq    %rbx
    popq    %rbp
    ret
    movq    %rbp, %rsp
    popq     %r15
    popq     %r14
    popq     %r13
    popq     %r12
    popq     %r11
    popq     %r10
    popq     %r9
    popq     %r8
    popq    %rcx
    popq    %rbx
    popq    %rbp
    ret
```

4.sy

```
.LC0:
    .string "%lld"
.LC1:
    .string "%lld\n"
    .globl  main
main:
    pushq   %rbp
    pushq   %rbx
    pushq   %rcx
    pushq   %r8
    pushq   %r9
    pushq   %r10
    pushq   %r11
    pushq   %r12
    pushq   %r13
    pushq   %r14
    pushq   %r15
```

```
    movq    %rsp, %rbp
    subq    $32, %rsp
    movq    $6, %r8
    movq    $6, %r8
    subq    $304, %rsp
    movq    $1, %r8
    subq    $16, %rsp
    movq    %r8, -312(%rbp)
    movq    $1, %r8
    subq    $16, %rsp
    movq    %r8, -328(%rbp)
.L1:
    movq    -312(%rbp), %r8
    movq    $312, %rcx
    movq    $5, %r9
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r9, %r8
    cmovl   %rcx, %rdx
    movq    %rdx, %r8
    movq    -328(%rbp), %r9
    movq    $328, %rcx
    movq    $5, %r10
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r10, %r9
    cmovl   %rcx, %rdx
    movq    %rdx, %r9
    andq    %r9, %r8
    movq    $0, %rcx
    cmpq    %r8, %rcx
    je  .L2
    movq    $0, %r8
    movq    -312(%rbp), %r9
    movq    $312, %rcx
    movq    $6, %rcx
    imulq   %rcx, %r9
    addq    %r9, %r8
    movq    -328(%rbp), %r9
    movq    $328, %rcx
    movq    $1, %rcx
    imulq   %rcx, %r9
    addq    %r9, %r8
    movq    $8 , %rcx
    imulq   %rcx, %r8
    addq    $8, %r8
    neg     %r8
    movq    (%rbp, %r8), %r9
    neg     %r8
    movq    -312(%rbp), %r10
    movq    $312, %rcx
    movq    -328(%rbp), %r11
    movq    $328, %rcx
    addq    %r11, %r10
    neg     %r8
    movq    %r10, (%rbp, %r8)
```

```
    neg     %r8
    movq    -328(%rbp), %r8
    movq    $328, %rcx
    movq    -328(%rbp), %r9
    movq    $328, %rcx
    movq    $1, %r10
    addq    %r10, %r9
    neg     %rcx
    movq    %r9, -328(%rbp)
    neg     %rcx
    jmp .L1
.L2:
.L3:
    movq    -312(%rbp), %r8
    movq    $312, %rcx
    movq    $5, %r9
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r9, %r8
    cmovl   %rcx, %rdx
    movq    %rdx, %r8
    movq    -328(%rbp), %r9
    movq    $328, %rcx
    movq    $6, %r10
    movq    $0, %rdx
    movq    $1, %rcx
    cmpq    %r10, %r9
    cmovl   %rcx, %rdx
    movq    %rdx, %r9
    andq    %r9, %r8
    movq    $0, %rcx
    cmpq    %r8, %rcx
    je  .L4
    movq    $0, %r8
    movq    -312(%rbp), %r9
    movq    $312, %rcx
    movq    $6, %rcx
    imulq   %rcx, %r9
    addq    %r9, %r8
    movq    -328(%rbp), %r9
    movq    $328, %rcx
    movq    $1, %rcx
    imulq   %rcx, %r9
    addq    %r9, %r8
    movq    $8 , %rcx
    imulq   %rcx, %r8
    addq    $8, %r8
    neg     %r8
    movq    (%rbp, %r8), %r9
    neg     %r8
    movq    -312(%rbp), %r10
    movq    $312, %rcx
    movq    -328(%rbp), %r11
    movq    $328, %rcx
    subq    %r11, %r10
    neg     %r8
```

```
    movq    %r10, (%rbp, %r8)
    neg     %r8
    movq    -312(%rbp), %r8
    movq    $312, %rcx
    movq    -312(%rbp), %r9
    movq    $312, %rcx
    movq    $1, %r10
    addq    %r10, %r9
    neg     %rcx
    movq    %r9, -312(%rbp)
    neg     %rcx
    jmp .L3
.L4:
    movq    $0, %r8
    movq    $1, %r9
    movq    $6, %rcx
    imulq   %rcx, %r9
    addq    %r9, %r8
    movq    $1, %r9
    movq    $1, %rcx
    imulq   %rcx, %r9
    addq    %r9, %r8
    movq    $8 , %rcx
    imulq   %rcx, %r8
    addq    $8, %r8
    neg     %r8
    movq    (%rbp, %r8), %r9
    neg     %r8
    movq    $0, %r10
    movq    $1, %r11
    movq    $6, %rcx
    imulq   %rcx, %r11
    addq    %r11, %r10
    movq    $4, %r11
    movq    $1, %rcx
    imulq   %rcx, %r11
    addq    %r11, %r10
    movq    $8 , %rcx
    imulq   %rcx, %r10
    addq    $8, %r10
    neg     %r10
    movq    (%rbp, %r10), %r11
    neg     %r10
    addq    %r11, %r9
    movq    $0, %r8
    movq    $4, %r11
    movq    $6, %rcx
    imulq   %rcx, %r11
    addq    %r11, %r8
    movq    $5, %r11
    movq    $1, %rcx
    imulq   %rcx, %r11
    addq    %r11, %r8
    movq    $8 , %rcx
    imulq   %rcx, %r8
    addq    $8, %r8
```

```
    neg     %r8
    movq    (%rbp, %r8), %r11
    neg     %r8
    addq    %r11, %r9
    subq    $16, %rsp
    movq    %r9, %rsi
    leaq    .LC1(%rip), %rdi
    call    printf@PLT
    addq    $16, %rsp
    movq    %rbp, %rsp
    popq     %r15
    popq     %r14
    popq     %r13
    popq     %r12
    popq     %r11
    popq     %r10
    popq     %r9
    popq     %r8
    popq    %rcx
    popq    %rbx
    popq    %rbp
    ret
```