```
.text
.globl main
move $fp $sp
la $a0 ProgStart
li $v0 4
syscall

# stored a new variable in the next offset

# new statement
# stored a new variable in the next offset

# new statement
li $t1, 1 # assigning a true value
sw $t1, -8
lw $t1 -8
sw $t1 -4 # assign to variable

# new statement
li $t1, 10 # assinging an int value
sw $t1, -8
lw $t1 -8
sw $t1 0 # assign to variable

# new statement
# stored a new variable in the next offset

# new statement
li $t1, 0 # assinging an int value
sw $t1, -12
lw $t1 -12
sw $t1 -8 # assign to variable

# new statement
Loop: lw $t0 -8 # load value
sw $t0 -12
lw $t0 -8 # load value
sw $t0 -16
lw $t0 -16
lw $t1 -12
mult $t2 $t1 $t0
mflo $t2
sw $t2 -12
li $t1, 10 # assinging an int value
sw $t1, -16
lw $t0 -16
lw $t1 -12
blt $t1 $t0 Skip
```

```
sw $t1 -12
lw $t0 -8 # load value
sw $t0 -12
li $t1, 1 # assinging an int value
sw $t1, -16
lw $t0 -16
lw $t1 -12
add $t2 $t1 $t0
sw $t2 -12
lw $t0 -12
sw $t0 -8 # assign to variable
j Loop
Skip:
# new statement
# stored a new variable in the next offset

# new statement
li $t1, 0 # assinging an int value
sw $t1, -16
lw $t1 -16
sw $t1 -12 # assign to variable

# new statement
Loop: lw $t0 -12 # load value
sw $t0 -16
lw $t0 -8 # load value
sw $t0 -20
lw $t0 -20
lw $t1 -16
blt $t1 $t0 Skip
sw $t1 -16
lw $t0 -4 # load value
sw $t0 -16
li $t1, 1 # assigning a true value
sw $t1, -20
lw $t0 -20
lw $t1 -16
sw $t1 -16
lw $t0 -12
lw $t1 -8
and $t2 $t1 $t0
mflo $t2
sw $t2 -8
# stored a new variable in the next offset

# new statement
lw $t0 -12 # load value
sw $t0 -16
```

```
lw $t0 -16
sw $t0 -12 # assign to variable

# new statement
Loop: lw $t0 -12 # load value
sw $t0 -16
lw $t0 0 # load value
sw $t0 -20
lw $t0 -20
lw $t1 -16
blt $t1 $t0 Skip
sw $t1 -16
lw $t0 -12 # load value
sw $t0 -16
lw $t0 -12 # load value
sw $t0 -20
lw $t0 -20
lw $t1 -16
mult $t2 $t1 $t0
mflo $t2
sw $t2 -16
lw $t0 0 # load value
sw $t0 -20
lw $t0 -20
lw $t1 -16
bne $t1 $t0 Skip
sw $t1 -16
lw $t0 -12
lw $t1 -8
or $t2 $t1 $t0
sw $t2 -8
lw $t0 -12 # load value
sw $t0 -12
li $t1, 1 # assinging an int value
sw $t1, -16
lw $t0 -16
lw $t1 -12
add $t2 $t1 $t0
sw $t2 -12
lw $t0 -12
sw $t0 -12 # assign to variable
j Loop
Skip:
# new statement
lw $t0 -12 # load value
sw $t0 -12
lw $t0 -12 # load value
sw $t0 -16
```

```
lw $t0 -16
lw $t1 -12
mult $t2 $t1 $t0
mflo $t2
sw $t2 -12
lw $t0 0 # load value
sw $t0 -16
lw $t0 -16
lw $t1 -12
sw $t1 -12
li $t1, 0 # assigning a false value
sw $t1, -12
lw $t1 -12
sw $t1 -4 # assign to variable
Skip:
# new statement
lw $t0 -12 # load value
sw $t0 -12
lw $t0 -12 # load value
sw $t0 -16
lw $t0 -16
lw $t1 -12
mult $t2 $t1 $t0
mflo $t2
sw $t2 -12
lw $t0 0 # load value
sw $t0 -16
lw $t0 -16
lw $t1 -12
bne $t1 $t0 Skip
sw $t1 -12
lw $t0 -12 # load value
sw $t0 -12
li $t1, 1 # assinging an int value
sw $t1, -16
lw $t0 -16
lw $t1 -12
add $t2 $t1 $t0
sw $t2 -12
lw $t0 -12
sw $t0 -12 # assign to variable
Skip:
# new statement
j Loop
Skip:
# new statement
lw $t0 -4 # load value
sw $t0 -12
```

```
li $t1, 1 # assigning a true value
sw $t1, -16
lw $t0 -16
lw $t1 -12
sw $t1 -12
lw $a0 -12
li $v0 1
sw $t1, -12
syscall
Skip:
# new statement
lw $t1 -4 # load value
sw $t1 -16
lw $t0 -16
lw $t1 -12
sw $t1 -12
Skip:
la $a0 ProgEnd
li $v0 4
syscall
li $v0 10
syscall
.data
ProgStart: .asciiz "Program Start\n"
ProgEnd: .asciiz "Program End\n"
```