```
# iki tamsayinin carpimi
 .data
 x: .word  6  # $t1
 y:  .word   -23    # $t2
 ms_sum: .word  0  # $t3
 ls_sum: .word   0   # $t4
 mask:  .word   1  # $t6

 .text
 .globl main

main:
 la $t0, x
 lw $t1, 0($t0)
 lw $t2, 4($t0)
 lw $t3, 8($t0)
 lw $t4, 12($t0)
 lw $t6, 16($t0)

geri: and $t7, $t2, $t6  # strip off appropriate multiplier bit
 beqz $t7, shift  # skip addition if multiplier is zero
 add $t3, $t3, $t1  # add partial sum
shift:  andi $t5, $t3, 1   # determine lsb of ms_sum
 or $t4, $t4, $t5 # place lsb of ms_sum in lsb of ls_sum
 ror $t4, $t4, 1  # shift ls_sum, moving new bit into msb
 sra $t3, $t3, 1  # shift ms_sum, maintaining sign
 sll $t6, $t6, 1  # update index
 bne $t6,$0,geri  # branch if not last iteration
 sw $t3, 8($t0)
 sw $t4, 12($t0)
 li $v0, 10   # code for program end
 syscall
```