1. Table

|  |  |
| --- | --- |
| **Operand** | **Value** |
| %rdi | 0x1000 |
| 0x1004 | 0xBB |
| $0x1008 | 0x1008 |
| (%rdi) | 0xAA |
| 4(%rdi) | 0xBB |
| 8(%rdi,%rcx) | 0xDD |
| 0x1002(%rdx,%rcx) | 0xCC |
| -4(%rdi,%rsi,4) | 0xAA |
| (%rdi,%rdx,4) | 0xCC |

1. Table

|  |  |  |
| --- | --- | --- |
| Instruction | Destination | Value |
| addq (%rdi),%rsi | %rsi | 0xAB |
| andq %rsi,%rdi | %rdi | 0x00 |
| subq %rsi,(%rdi) | 0x1000 | 0xA9 |
| incq %rsi | %rsi | 0x2 |
| decq %rdx | %rdx | 0x1 |
| xorq (%rdi,%rdx,4),%rcx | %rcx | 0xC8 |
| orq 0x1002(%rdx,%rcx),%rsi | %rsi | 0xCD |

1. Fill in the blank

unknown:

imulq %rdx, %rsi

leaq (%rsi,%rdi), %rax

ret

long unknown(long x, long y, long z) {

return **x + y \* z**;

}

unknown:

movq %rdi, %rax

salq $3, %rax

addq %rdi, %rax

ret

long unknown(long x) {

return **9 \* x;**

}

1. C Code

#include <stdio.h>

#include <stdlib.h>

// %rdx, %rsi, %rdi

long decode2(long x, long y, long z){

y -= z;

x \*= y;

return x ^ ((y << 63) >> 63);

}

int main(){

long x = 1, y = 2, z = 2;

decode(x, y , z);

}

1. Assembly program

leaq (%rsi, %rdi), %rcx

movq %rcx, %rdi

subq %rdx, %rdi

movq %rdi, %rsi

andq %rcx, %rsi

movq %rsi, %rax

imulq %rdi, %rax

ret