1.

Address	Value	--	Register	Value
0x1000	0xAA	--	%rdi	0x1000
0x1004	0xBB	--	%rsi	0x1
0x1008	0xCC	--	%rdx	0x2
0x100C	0xDD	--	%rcx	0x3

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

2.

Instruction	Destination	Value
addq (%rdi), %rsi	%rsi	0xAB
andq %rsi, %rdi	%rdi	0x00
subq %rsi, (%rdi)	0x0	0x01
incq %rsi	%rsi	0x2
decq %rdx	%rdx	0x1
xorq (%rdi, %rdx, 4), %rcx	%rcx	0x5
orq 0x1002(%rdx, %rcx), %rsi	%rsi	0xCE

3.
```
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 x*9;
}
```

4.

```
#include <stdio.h>

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

        long a;

        y=y-z;

        x=x*y;

        a=y;

        a=a<<63;

        a=a>>63;

        a=a^x;

        return a;

}


int main(){

        long x,y,z,n;

        printf(" Enter the value of x:");

        scanf("%ld",&x);

        printf(" Enter the value of y:");

        scanf("%ld",&y);
```

```
        printf(" Enter the value of z:");
        scanf("%ld",&z);

        n=decode2(x,y,z);
        printf(" Result of decode2(%ld,%ld,%ld) is %ld \n",x,y,z,n);
        return 0;
}
```

```
[appa@fedora 4hwfold]$ ./prob4.out
 Enter the value of x:1
 Enter the value of y:2
 Enter the value of z:3
 Result of function decode2(1,2,3) -> 0
[appa@fedora 4hwfold]$ ./prob4.out
 Enter the value of x:5
 Enter the value of y:3
 Enter the value of z:6
 Result of function decode2(5,3,6) -> 14
[appa@fedora 4hwfold]$ ./prob4.out
 Enter the value of x:3
 Enter the value of y:4
 Enter the value of z:5
 Result of function decode2(3,4,5) -> 2
[appa@fedora 4hwfold]$
```

5.

```
long arith(long x, long y, long z) {
                long t1,t2,t3,t4;
                t1 = x + y;
                t2 = t1 - z;
                t3 = t1 & t2;
                t4 = t2 * t3;

                return t4;
```

```
}

arith:
        leaq (%rsi,%rdi), %rcx # t1
        movq %rcx, %rdi
        subq %rdx, %rdi # t2
        movq %rcx, %rsi
        andq %rdi, %rsi # t3
        movq %rdi, %rax
        mulq %rsi, %rax # t4
        re
```