

Alex Sobiek
COMP 264
Professor Greenberg

Homework 4

A. Which registers hold program values `z` and `num`?

`%rdi` holds `z` and `%rsi` holds `num`

B. In which assembly code line number is `accum` updated on each iteration of the loop?

`accum` is updated on line 7.

C. In which assembly code line number is `loopvar` updated on each iteration of the loop?

`loopvar` is updated on line 5.

D. What are the initial values of `accum` and `loopvar`?

`loopvar` is initially set to 1 on line 1 and `accum` is set to 0 on line 2

```
movl    $1, %edx ; line #1    Move $1 into %edx
xorl    %eax, %eax ; line #2    XOR same register to set it to 0
```

E. What is the test condition for `loopvar`?

The test condition for `loopvar` is not equal to 0. This can be found on line 8

```
testq   %rdx, %rdx ; line #8    Is %rdx equal to 0?
```

F. How does `loopvar` get updated?

`loopvar` is updated on line 5 where it left shifts `loopvar` (`%rdx`) by `%cl`. `%cl` is the lower 8 bits of `%ecx`, which on line 3 is set to hold the lower 8 bits of `%rsi` (`num`)

```
salq    %cl, %rdx ; line #5    Left shift %rdx by %cl
```

G. How does `accum` get updated?

accum is updated with a XOR of itself and %rsi. %rsi (num) is set with an AND of itself and %rdi (z)

```
andq    %rdi, %rsi ; line #6    AND %rdi (long z), %rsi (int num)
xorq    %rsi, %rax ; line #7    XOR %rsi, %rax (%rax = long accum)
```

H. Fill in all the missing parts of the C code.

```
long loop(long z, int num) {
    long accum = 0;
    long loopvar;
    for (loopvar = 1; loopvar != 0; loopvar <= num) {
        accum ^= (z & num);
    }
    return accum;
}
```