Nate Flasher

3/29/21

Assembly Language

3.58)

```
Long decode2(long x, long y, long z){
Long temp = y-z;
Long temp2 = temp*x;
Temp << 63;
Temp >>63;
Temp3 = temp2 ^ temp;
Return temp3;
}
```

3.60)

```
Long result = 0;
Long mask;
For (mask = 1; mask != 0; mask = mask << n){
        Result |= mask&x;
}
Return result;
```

A) x in %rdi, n in %esi, result in %rax, mask in %rdx

B) result = 0, mask = 1

C) mask != 0

D) mask = mask << n

E) mask&x in %r8, result = result | %r8

3.62)

```
Typedef enum {MODE_A, MODE_B, MODE_C, MODE_D, MODE_E} mode_t;
Long switch3(long *p1, long *p2, mode_t action){
Long result = 0;
Switch(action){
```

```
        Case MODE_A:
                Result = *p2;
                Long temp = *p1;
                *p2 = temp;
                Break;
        Case MODE_B:
                Result = *p1+*p2;
                *p1=result;
                Break;
        Case MODE_C:
                *p1=59;
                Result=*p2;
                Break;
        Case MODE_D:
                *p1=*p2;
        Case MODE_E:
                Result=27;
                Break;
        Default:
                Result=12;
                Break;
        }
Return result;
}
```

3.63)

```
Long switch_prob(long x, long n){
Long result=x;
Switch(n){
        Case 60:
```

```
        Case 62:
        Result*=8;
        Break;
        Case 63:
        Result >>=3;
        Break;
        Case 64:
        X = (x<<4) -x;
        Case 65:
        X*=x;
        Default:
        Result = x+75;
        Break;
}
Return result;
}
```