- CPSC 275: Introduction to Computer Systems

CPSC 275: Introduction to Computer Systems

Fall 2025

- Syllabus
- Schedule
- Resources
- Upload
- Solution

Solution to Homework 17

1. A.
```
Register Variable
  %eax      x
  %ecx      y
  %edx      n
```

   B. The *body-statement* portion consists of lines 3 through 5 in the C code and lines 5 through 7 in the assembly code. The *test-expr* portion is on line 6 in the C code. In the assembly code, it is implemented by the instructions on lines 8 through 11.

   C.
```
x at %ebp+8, y at %ebp+12, n at %ebp+16
1       movl  8(%ebp), %eax     # Get x
2       movl  12(%ebp), %ecx    # Get y
3       movl  16(%ebp), %edx    # Get n
4   .L2:                       # loop:
5       addl  %edx, %eax        #   x += n
6       imull %edx, %ecx        #   y *= n
7       subl  $1, %edx          #   n--
8       testl %edx, %edx        #   Test n
9       jle   .L5               #   If <= 0, goto done
10      cmpl  %edx, %ecx        #   Compare y:n
11      jl    .L2               #   If <, goto loop
12  .L5:                       # done:
```

2. A.
```
int fun_a(unsigned x) {
    int val = 0;
    while (x) {
       val ^= x;
       x >>= 1;
    }
    return val & 0x1;
}
```

   B. This code computes the *parity* of argument `x`. That is, it returns 1 if there is an odd number of ones in `x` and 0 if there is an even number.

3. A.
```
int fun_b(unsigned x) {
      int val = 0;
      int i;
      for (i = 0; i < 32; i++) {
         val = (val << 1) | (x & 0x1);
         x >>= 1;
      }
      return val;
   }
```

   B. This code reverses the bits in `x`, creating a mirror image. It does this by shifting the bits of `x` from left to right, and then filling these bits in as it shifts `val` from right to left.

4.

5. `%eax` is set to the address of the `popl` instruction.

- **Welcome: Sean**
  - LogOut