

- [CPSC 275: Introduction to Computer Systems](#)

## [CPSC 275: Introduction to Computer Systems](#)

Fall 2025

- [Syllabus](#)
- [Schedule](#)
- [Resources](#)
- [Upload](#)
- [Solution](#)

# Homework 17

NOTE: You are not required to hand in the following exercises, but you are strongly encouraged to complete them to strengthen your understanding of the concepts covered in class.

## 1. For the C code

```

1  int dw_loop(int x, int y, int n) {
2      do {
3          x += n;
4          y *= n;
5          n--;
6      } while ((n > 0) && (y < n));
7      return x;
8  }
```

The gcc C compiler generates the following assembly code:

```

      x at %ebp+8, y at %ebp+12, n at %ebp+16
1      movl 8(%ebp), %eax
2      movl 12(%ebp), %ecx
3      movl 16(%ebp), %edx
4      .L2:
5      addl %edx, %eax
6      imull %edx, %ecx
7      subl $1, %edx
8      testl %edx, %edx
9      jle .L5
10     cmpl %edx, %ecx
11     jl .L2
12     .L5:
```

- State how the individual registers are used in the code.
- Identify *test-expr* and *body-statement* in the C code, and the corresponding lines in the assembly code.
- Add annotations to the assembly code describing the operation of the program.

## 2. A function, `fun_a`, has the following overall structure:

```

int fun_a(unsigned x) {
    int val = 0;
    while ( _____ ) {
        _____;
    }
}
```

```

    }
    return _____;
}

```

The gcc C compiler generates the following assembly code:

```

    x at %ebp+8
1      movl 8(%ebp), %edx
2      movl $0, %eax
3      testl %edx, %edx
4      je    .L7
5      .L10:
6      xorl  %edx, %eax
7      shr  %edx          # Shift right by 1
8      jne  .L10
9      .L7:
10     andl  $1, %eax

```

- A. Use the assembly-code version to fill in the missing parts of the C code.
- B. Describe in English what this function computes.

3. A function `fun_b` has the following overall structure:

```

int fun_b(unsigned x) {
    int val = 0;
    int i;
    for ( ____ ; ____ ; ____ ) {
        _____
    }
    return val;
}

```

The gcc C compiler generates the following assembly code:

```

    x at %ebp+8
1      movl 8(%ebp), %ebx
2      movl $0, %eax
3      movl $0, %ecx
4      .L13:
5      leal (%eax,%eax), %edx
6      movl %ebx, %eax
7      andl $1, %eax
8      orl  %edx, %eax
9      shr  %ebx          # Shift right by 1
10     addl $1, %ecx
11     cmpl $32, %ecx
12     jne  .L13

```

- A. Use the assembly-code version to fill in the missing parts of the C code.
- B. Describe in English what this function computes.

4. Assuming that `a` and `b` are in `%ebx` and `%ecx`, respectively, write assembly code for the following function:

```

int sum(int a, int b) {
    int val = 0;
    for (int i = a, i <= b; i++)
        val += i;
    return val;
}

```

5. In the following code fragment:

```
    call next  
next:  
    popl %eax
```

to what value does register %eax get set?

- **Welcome: Sean**

- [LogOut](#)

