

# Model 1 Low-Level Languages

The following program, shown in three different languages, calculates the sum of numbers from 1 to 10. In other words, it adds  $1 + 2 + \dots + 10 = 55$ .

Machine Code (1st Generation)	Y86-64 Assembly (2nd Generation)	Standard C (3rd Generation)
0x000: 0x000: 70000100000000000000	.pos 0 code jmp _start	
0x100: 0x100: 0x100: 30f00b0000000000000000 0x10a: 30f3010000000000000000 0x114: 30f1020000000000000000 0x11e: 30f2010000000000000000	.pos 0x100 code _start: irmovq \$0xb, %rax irmovq \$0x1, %rbx irmovq \$0x2, %rcx irmovq \$0x1, %rdx	int main() { int upper = 11; int sum = 1; int val = 2;
0x128: 2017 0x12a: 6107 0x12c: 73460100000000000000	rrmovq %rcx, %rdi subq %rax, %rdi je done	while (val < upper) { sum = sum + val; val++; }
0x135: 0x135: 6013 0x137: 6021	loop: addq %rcx, %rbx addq %rdx, %rcx	}
0x139: 2017 0x13b: 6107 0x13d: 74350100000000000000	rrmovq %rcx, %rdi subq %rax, %rdi jne loop	
0x146: 0x146: 00	done: halt	

## Questions (15 min)

Start time: \_\_\_\_\_

1. Compare the length of each program. Do not count labels (e.g, 0x000:, .pos 0 code) or punctuation (e.g., {, }).

a) How many instructions of machine code?

b) How many instructions of assembly code?

c) How many non-blank, non-brace lines of C code?

2. All data values for this program are stored in registers named %rax, %rbx, etc.

- a) In which register is the sum stored?
  - b) In which register is the next value to add stored?
3. The instruction `irmovq` means “move immediate value to register”. Immediate values begin with a dollar sign (\$), and registers begin with a percent sign (%).
- a) What is the value 11 in assembly code?
  - b) Does assembly use decimal or hexadecimal?
  - c) Does Standard C use decimal or hexadecimal?
4. In terms of the machine, what does an assignment statement do? As part of your answer, name the instructions in Model 1 that perform assignment.
5. Consider the line `rrmovq %rcx, %rdi`. The instruction `rrmovq` means “move (copy) register to register”.
- a) What is stored in register `%rcx`?
  - b) Where is this value copied to?
6. The instruction `subq` means “subtract”. Given two registers  $R$  and  $T$ , `subq` performs  $R - T$  and stores the result in  $T$ .
- a) What is stored in register `%rax`?
  - b) In what case would  $\%rax - \%rdi$  be zero?
7. The instruction `je` means “jump if the last operation’s result equals 0”, and the instruction `jne` means “jump if the last operation’s result does not equal 0”. Circle the portion of assembly code that corresponds to the `while` loop in C.