

Project 4: Hack “Compilation” Examples

CS 220

For an exam, be ready to “compile” any of the examples in the slides to Hack, not just these selected examples.

1 Selected Examples from Slide 8

1. Set D to A-1

```
D=A-1
```

2. Set D to 19

```
@19 // Load 19 in A
D=A
```

3. Set RAM[5034] to D-1

```
@5034 // Load 5034 into A
M=D-1 // M corresponds to RAM[A]
```

2 Selected Examples from Slide 9

1. sum = 0

```
@sum // Load sum's address into A
M=0
```

2. q = sum + 12 - j

```
@sum
D=M
@12
D=D+A
@j
D=D-M
@q
M=D
```

3. arr[3] = -1

```
@arr
D=A // Load arr's base address into D
@3
A=D+A // Add the index to the base address and store in A
M=-1
```

4. `arr[j] = 17`

This one is tricky; it seems as if we need three registers. The solution is to compute the target address (`arr[j]`'s location in RAM) and store it temporarily in virtual register R15.

```
@arr
D=A      // Load arr's base address into D
@j
D=D+M    // Add the index, j, to the base address and store in D;
          // D now holds the target address
@R15
M=D      // Store the computed target address in R15
@17
D=A      // The value 17 is now in D
@R15
A=M      // Load the computed target address into A
M=D      // Store 17 to arr[j]
```

3 Selected Examples from Slide 11

1. `goto 50`

```
@50
0;JMP
```

2. `if (D==0) goto 112`

```
@112
D;JEQ
```

3. `if (sum > 0) goto END`

```
@sum
D=M
@END
D;JGT
```

4 Control Structure Examples

See Slides 12 and 13 of the Project 4 Machine Language slide show for the high level background of the following.

4.1 Simple If Conditional

Pseudocode:

```
if (k < 0)
    k = 0;
more code...
```

Intermediate pseudocode replacing if with if/goto:

```
// Note that we've inverted the original condition's relational
// operator because of how if/goto works.
if (k >= 0) goto ENDIF1
k=0;
(ENDIF1)
more code...
```

Hack assembly language:

```
@k        // Load k's address into A.
D=M       // Load k's value into D. Remember, "M" is RAM[A].
@ENDIF1   // Load the jump address into A
D;JGE     // Jump past the if block if k >= 0
@k
M=0       // Set k's value in RAM to 0.
(ENDIF1)
more code...
```

4.2 If / Else Conditional

Pseudocode:

```
if (i <= j)
    min = i;
else
    min = j;
more code...
```

First, rewrite the condition as $(i - j \leq 0)$ because all Hack conditional jump instructions compare their operand to 0. Put the 0 on the right-hand side to match the way that the conditional jump instructions work in Hack. Intermediate pseudocode using `if/goto`:

```
// Again, note the condition's inverted relational operator.
if (i - j > 0) goto ELSE2
min = i;
jump to ENDIF2
(ELSE2)
min = j;
(ENDIF2)
more code...
```

Hack assembly language:

```
@i          // Load i's address into A
D=M         // Copy i's value into D
@j          // Load j's address into A
D=D-M      // Copy (i-j) into D
@ELSE2     // Load the jump address into A
D;JGT      // Jump to the else block if (i - j) > 0
           // If block
@i          // Load i's address into A
D=M         // Copy i's value into D
@min        // Load min's address into A
M=D        // Set min's value in RAM to i's value
@ENDIF2
0;JMP      // Jump around the else block.
(ELSE2)    // Else block
@j          // The next several Hack instructions perform
D=M        //     min = j;
@min
M=D
(ENDIF2)
more code...
```

4.3 While Loop

Pseudocode:

```
sum = 0;
i = 1;
while (i < 10) {
    sum = sum + i;
    i = i + 1;
}
more code...
```

Again, rewrite the condition as $(i - 10 < 0)$. You should be getting the hang of this by now, so, no more comments. Intermediate pseudocode using `if/goto`:

```
sum = 0;
i = 1;
(WHILE1)
    if (i - 10 >= 0) goto ENDWHILE1
    sum = sum + i;
    i = i + 1;
    jump to WHILE1
(ENDWHILE1)
more code...
```

Again, no more comments. Hack assembly language:

```
@sum
M=0
@i
M=1
(WHILE1)
@i
D=M
@10
D=D-A
@ENDWHILE1
D;JGE
@i
D=M
@sum
M=M+D
@i
M=M+1
@WHILE1
0;JMP
(ENDWHILE1)
more code...
```