### Instruction

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Action</th>
<th>Addressing Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>R2 = R2 + R3</td>
<td>Register</td>
</tr>
<tr>
<td>ADD</td>
<td>R2 = R2 + 1</td>
<td>Immediate</td>
</tr>
<tr>
<td>AND</td>
<td>R2 = R2 AND R3</td>
<td>Register</td>
</tr>
<tr>
<td>AND</td>
<td>R2 = R2 AND 0</td>
<td>Immediate</td>
</tr>
<tr>
<td>BR[n][z][p]</td>
<td>If [n][z][p] Go to LABEL</td>
<td>CC, PC-relative</td>
</tr>
<tr>
<td>JMP</td>
<td>PC = R1</td>
<td>Register</td>
</tr>
<tr>
<td>JSR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JSRR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD</td>
<td>R2 = m[LABEL]</td>
<td>Register, PC-relative</td>
</tr>
<tr>
<td>LDI</td>
<td>R2 = m[m[LABEL]]</td>
<td>Register, Indirect</td>
</tr>
<tr>
<td>LDR</td>
<td>R2 = m[R0+n]</td>
<td>Base Register</td>
</tr>
<tr>
<td>LEA</td>
<td>R2 = LABEL</td>
<td>Register, PC-relative</td>
</tr>
<tr>
<td>NOT</td>
<td>R2 = NOT(R1)</td>
<td>Register</td>
</tr>
<tr>
<td>RET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td>m[LABEL] = R2</td>
<td>Register, PC-relative</td>
</tr>
<tr>
<td>STI</td>
<td>m[m[LABEL]] = R2</td>
<td>Register, Indirect</td>
</tr>
<tr>
<td>STR</td>
<td>m[R0 + n] = R2</td>
<td>Register, Base Register</td>
</tr>
<tr>
<td>TRAP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Today:**

### CS Information Session at 4:00p

ALL majors/minors should attend. (Check your e-mail for sign up)

### Talk: What is Data Science?

At 6:00p (check your email for sign up)

Lab #2 is posted.
**Problem:** Write a complete LC-3 program to perform integer division on two numbers, A and B. Assume that A is stored in location x6000 and B is in x6001. The integer division should produce two results: a quotient (stored in R0) and a remainder (stored in R1). For example,

Suppose A = 22 and B = 7. Then, after division R0 will contain 3 and R1 will contain 1.

```
Algorithm

A = 22, B = 7
Q = 0
while A >= B do
  A = A - B
  Q = Q + 1
R = A

* Q + R are the two answers

Register Allocation

Q: R0, R: R1
A: R1, B: R2 - b: R3
R1 = A, R2 = B
R0 = 0, R3 = -B
while R1 >= R2 do
  R1 = R1 + R3
  R0 = R0 + 1

.pseudocode

LC-3 pseudocode

LC-3 code

```

```

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

.ORIG x6000
A .FILL 22
B .FILL 7
.END

.ORIG x600A

START
LD R1, A ; R1 = A
LD R2, B ; R2 = B
JMP NOT R3, R2 ; R3 = -B
ADD R3, R3, #1
AND R0, R0, #0 ; R0 = 0
JMP R1 >= R2 ; R4 = R1 - R2 = R4 = R1 + R3
LOOP
ADD R4, R1, R3
BRn DONE
J 0 do
ADD R1, R1, R3 ; R1 = R1 + R3
ADD R0, R0, #1 ; R0 = R0 + 1
BR LOOP
DONE
HALT
.END
```
Problem: Given a string (e.g. "BRYN MAWR") and an input character (e.g. 'R')
Output: # of occurrences of input character in string (e.g. 2)
Assume # occurrences will be less than 10 (why?)

Algorithm:

\[
\text{count} = 0 \\
\text{c = input a char} \\
\text{nextChar = first char in strip} \\
\text{while nextChar != NULL do} \\
\quad \text{if nextChar = c then} \\
\quad \quad \text{count = count + 1} \\
\quad \text{nextChar = next char in strip} \\
\text{output count}
\]
count = 0
nextCH = next character in string
while nextCH != NULL do
    if nextCH = c then
        count = count + 1
    nextCH = next character in string
output count

---

R2 = 0
R0 = input from keyboard
R3 = addr of string (x4000)
R1 = M[R3 + 0]

while R1 ≠ 0 do
    if R1 = c then : R1 = R1 - R0
    R2 = R2 + 1
; nextCH ...
R3 = R3 + 1

output R2
; Program to count the occurrences of a character in a string.
; Program inputs the character to count from keyboard.
; The string is provided starting from memory location x4000 (NULL terminated).
; e.g. if the string is "BRYN MAWR"
; and the input char is 'R'
; the output displayed will be 2
;
; Assume the count will always be less than 10 (why??)

.ORIG x3000

START AND R2, R2, #0 ; R2 = 0 (count)
LD R3, PTR ; Address of string
TRAP x23 ; R0 = input char
LDR R1, R3, #0 ; R1 = next char

; WHILE R1 is not NULL DO
LOOP BRz OUTPUT ; next char is null
;
; IF char in R1 = R0
NOT R1, R1
ADD R1, R1, #1 ; R1 = -R1
ADD R1, R1, R0 ; R1 = R0 - R1
BRnp NEXTCH
;
; THEN
ADD R2, R2, #1 ; R2 = R2 + 1 (incr count)
;
NEXTCH ADD R3, R3, #1 ; incr char pointer
LDR R1, R3, #0 ; R1 = next char
BR LOOP

; Output count
OUTPUT LD R0, ASCII ; Load ASCII template in R0
ADD R0, R0, R2 ; Add count to R0 (ASCII count)
TRAP x21 ; Output count in ASCII
HALT

; STORAGE
ASCII .FILL x0030
PTR .FILL x4000

.END

.ORIG x4000

MESG .STRINGZ "BRYN MAWR"
.END