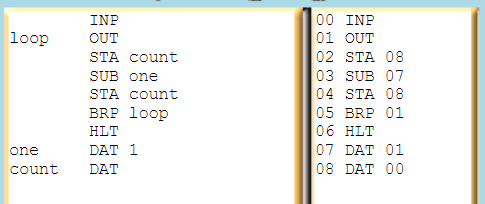
Name: REIMARC G. CORPUZ

Course/Sec: BSCPE IV G-MIX

Create a program which takes in a number and find and display the sum of its digits.

**PROGRAM:**

 INP

loop OUT

STA count

SUB one

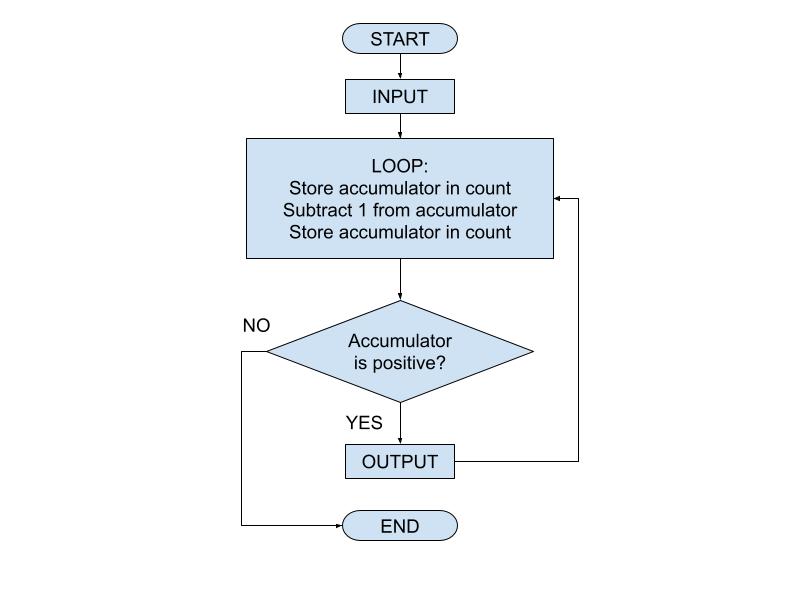
STA count

BRP loop

HLT

one DAT 1

count DAT



**PSEUDOCODE AND FLOWCHART:**

Input // Read a value from input and store it in the accumulator

Loop: // Label for the loop

Store accumulator in memory location count

Subtract 1 from the accumulator

Store accumulator in memory location count

If the accumulator is positive, jump to Loop

Halt the program

one: // Data location for the value 1

Data 1

count: // Data location for count

Data 0

**MNEMONIC ASSEMBLER:**

|  |  |
| --- | --- |
| **MAILBOX** | **MNEMONIC** |
| 00 | NP |
| 01 | OUT |
| 02 | STA 08 |
| 03 | SUB 07 |
| 04 | STA 08 |
| 05 | BRP 01 |
| 06 | HLT |
| 07 | DAT 01 |
| 08 | DAT 00 |

**LMC:**

**INP**

**Loop OUT** *//This instruction outputs the value in the accumulator to the console.*

**STA 08***//This instruction stores the value in the accumulator to memory address 08.*

**SUB 07***//This instruction subtracts the value at address 07 from the accumulator.*

**STA 08***//This instruction stores the result of the subtraction back in memory address 08.*

**BRP 01***//This instruction branches to address 01 if the value in the accumulator is positive or zero.*

**HLT***//This instruction halts the program.*

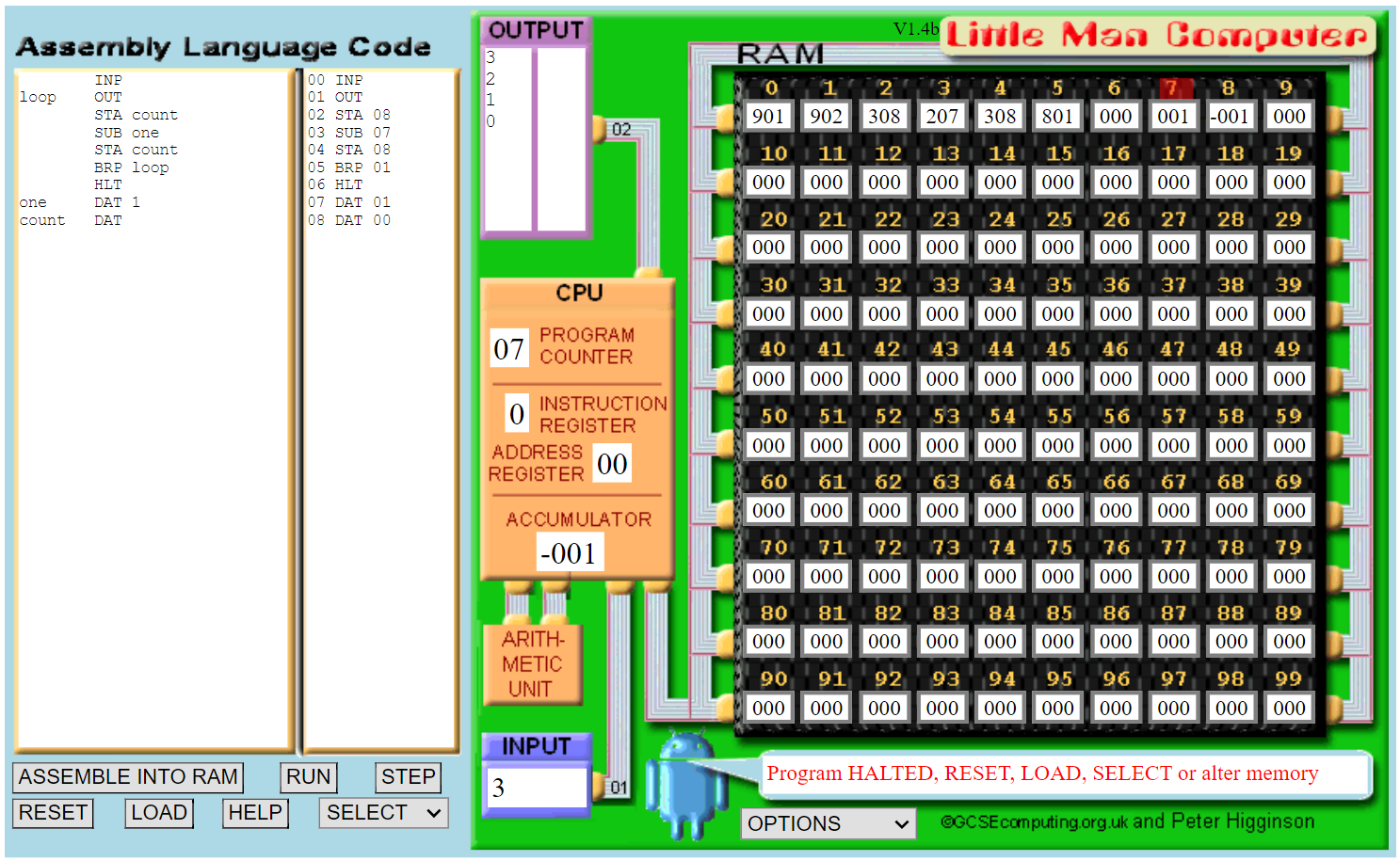
**one DAT 1**

**count DAT**

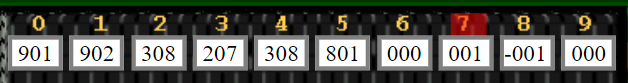
**ADDRESS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ADDRESS** | **OPCODE** | **OPERAND** |  |
| 00 | 901 | - | INP |
| 01 | 902 | - | OUT |
| 02 | 301 | 09 | STA count |
| 03 | 511 | 07 | SUB one |
| 04 | 301 | 09 | STA count |
| 05 | 600 | 00 | BRP loop |
| 06 | 000 | - | HLT |
| 07 | - | 1 | DAT 1 |
| 08 | - | 0 | DAT 0 |
| 09 | - | 0 | DAT 0 |

**SCREENSHOT OF LMC SIMULATOR ASSEMBLER:**



**SCREENSHOT OF LMC MAILBOXES:**



**CONSOLE MESSAGE:**

1. INPUT required

2. INPUT value loaded into accumulator

3. STORE value in accumulator at RAM address 00

4. OUTPUT value stored in accumulator

5. STORE value in accumulator at RAM address 00

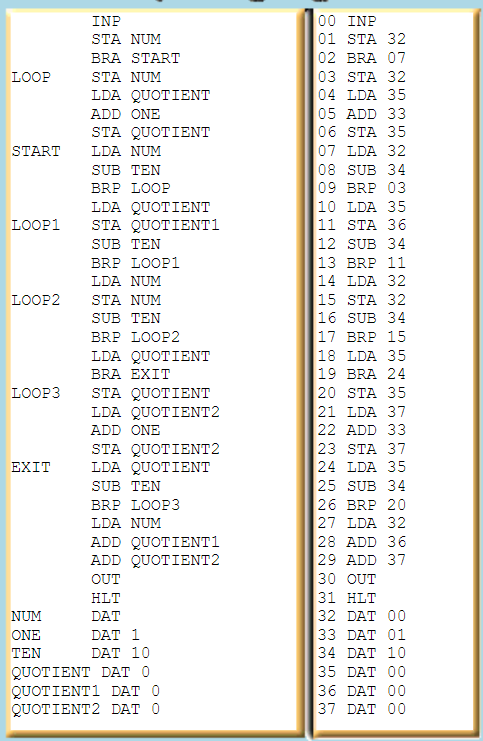
6. SUBTRACT from accumulator the contents of RAM address 01

7. STORE value in accumulator at RAM address 00

8. BRANCH (if zero or positive) to memory address 02

9. Program HALTED, RESET, LOAD, SELECT or alter memory

Create a program that will accept large number and find and display the sum of its digits.

**PROGRAM**

INP

STA NUM

BRA START

LOOP STA NUM

LDA QUOTIENT

ADD ONE

STA QUOTIENT

START LDA NUM

SUB TEN

BRP LOOP

LDA QUOTIENT

LOOP1 STA QUOTIENT1

SUB TEN

BRP LOOP1

LDA NUM

LOOP2 STA NUM

SUB TEN

BRP LOOP2

LDA QUOTIENT

BRA EXIT

LOOP3 STA QUOTIENT

LDA QUOTIENT2

ADD ONE

STA QUOTIENT2

EXIT LDA QUOTIENT

SUB TEN

BRP LOOP3

LDA NUM

ADD QUOTIENT1

ADD QUOTIENT2

OUT

HLT

NUM DAT

ONE DAT 1

TEN DAT 10

QUOTIENT DAT 0

QUOTIENT1 DAT 0

QUOTIENT2 DAT 0

**PSEUDOCODE:**

Input NUM

Store NUM in memory location NUM

Branch to START

LOOP:

Store NUM in memory location NUM

Load QUOTIENT

Add ONE

Store the result in QUOTIENT

START:

Load NUM

Subtract TEN

If result is positive, branch to LOOP

LOOP1:

Store QUOTIENT in memory location QUOTIENT1

Subtract TEN

If result is positive, branch to LOOP1

Load NUM

LOOP2:

Store NUM in memory location NUM

Subtract TEN

If result is positive, branch to LOOP2

Load QUOTIENT

LOOP3:

Store QUOTIENT in memory location QUOTIENT

Load QUOTIENT2

Add ONE

Store the result in QUOTIENT2

Subtract TEN

If result is positive, branch to LOOP3

EXIT:

Load QUOTIENT

Subtract TEN

If result is positive, branch to LOOP3

Load NUM

Add QUOTIENT1

Add QUOTIENT2

Output the result

Halt the program

Data Section:

NUM Data

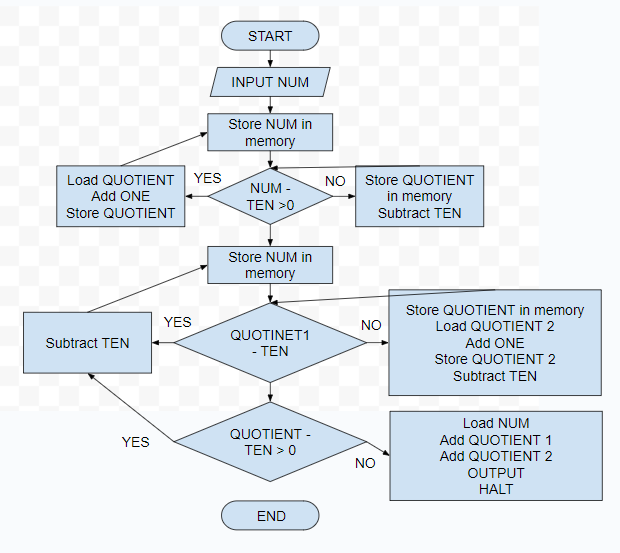
ONE Data 1

TEN Data 10

QUOTIENT Data 0

QUOTIENT1 Data 0

QUOTIENT2 Data 0

**FLOWCHART:**

**MNEMONIC ASSEMBLER:**

|  |  |
| --- | --- |
| **MAILBOX** | **MNEMONIC** |
| 00 | INP |
| 01 | STA 32 |
| 02 | BRA 07 |
| 03 | STA 32 |
| 04 | LDA 35 |
| 05 | ADD 33 |
| 06 | STA 35 |
| 07 | LDA 32 |
| 08 | SUB 34 |
| 09 | BRP 03 |
| 10 | LDA 35 |
| 11 | STA 36 |
| 12 | SUB 34 |
| 13 | BRP 11 |
| 14 | LDA 32 |
| 15 | STA 32 |
| 16 | SUB 34 |
| 17 | BRP 15 |
| 18 | LDA 35 |
| 19 | BRA 24 |
| 20 | STA 35 |
| 21 | LDA 37 |
| 22 | ADD 33 |
| 23 | STA 37 |
| 24 | LDA 35 |
| 25 | SUB 34 |
| 26 | BRP 20 |
| 27 | LDA 32 |
| 28 | ADD 36 |
| 29 | ADD 37 |
| 30 | OUT |
| 31 | HLT |
| 32 | DAT 00 |
| 33 | DAT 01 |
| 34 | DAT 10 |
| 35 | DAT 00 |
| 36 | DAT 00 |
| 37 | DAT 00 |

**LMCl**

INP

STA 32

BRA 07

LOOP STA 32

LDA 35

ADD 33

STA 35

START LDA 32

SUB 34

BRP 03

LDA 35

LOOP1 STA 36

SUB 34

BRP 11

LDA 32

LOOP2 STA 32

SUB 34

BRP 15

LDA 35

BRA 24

LOOP3 STA 35

LDA 37

ADD 33

STA 37

EXIT LDA 35

SUB 34

BRP 20

LDA 32

ADD 36

ADD 37

OUT

HLT

NUM DAT 00

ONE DAT 01

TEN DAT 10

QUOTIENT DAT 00

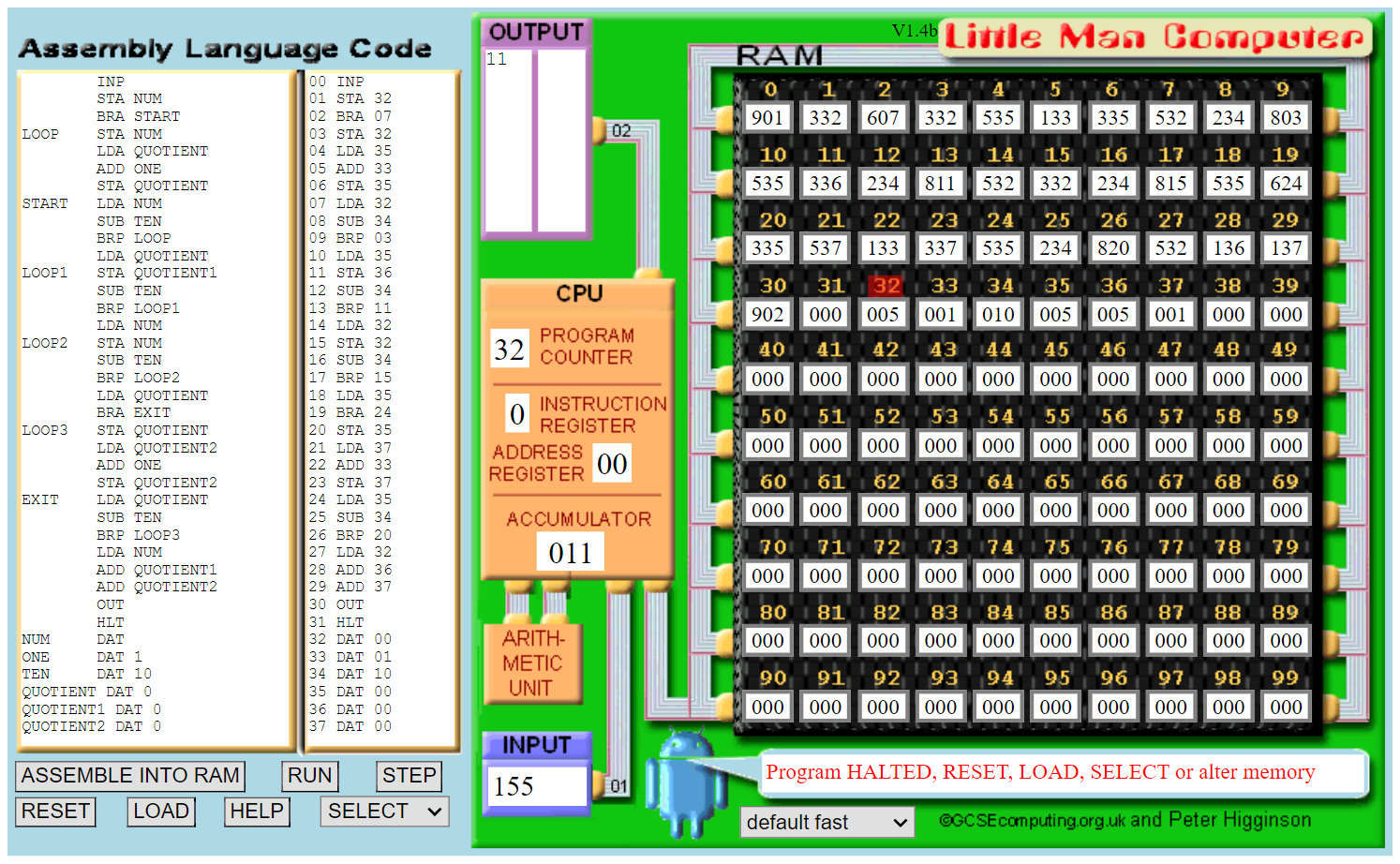
QUOTIENT1 DAT 00

QUOTIENT2 DAT 00

**ADDRESS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ADDRESS** | **OPCODE** | **OPERAND** | **COMMENT** |
| 0 | 901 | - | INP |
| 1 | 201 | 40 | STA |
| 2 | 600 | 10 | BRA |
| 3 | 201 | 40 | STA |
| 4 | 501 | 41 | LDA |
| 5 | 301 | 42 | ADD |
| 6 | 201 | 41 | STA |
| 7 | 501 | 40 | LDA |
| 8 | 801 | 43 | SUB |
| 9 | 600 | 3 | BRP |
| 10 | 501 | 41 | LDA |
| 11 | 201 | 44 | STA |
| 12 | 801 | 43 | SUB |
| 13 | 600 | 11 | BRP |
| 14 | 501 | 40 | LDA |
| 15 | 201 | 40 | STA |
| 16 | 801 | 43 | SUB |
| 17 | 600 | 14 | BRP |
| 18 | 501 | 41 | LDA |
| 19 | 600 | 24 | BRA |
| 20 | 201 | 41 | STA |
| 21 | 501 | 45 | LDA |
| 22 | 301 | 42 | ADD |
| 23 | 201 | 45 | STA |
| 24 | 501 | 41 | LDA |
| 25 | 801 | 43 | SUB |
| 26 | 600 | 20 | BRP |
| 27 | 501 | 40 | LDA |
| 28 | 301 | 44 | ADD |
| 29 | 301 | 45 | ADD |
| 30 | 901 | - | OUT |
| 31 | 700 | - | HLT |
| 32 | 0 | 0 | DAT |
| 33 | 1 | 1 | DAT |
| 34 | 10 | 10 | DAT |
| 35 | 0 | 0 | DAT |
| 36 | 0 | 0 | DAT |
| 37 | 0 | 0 | DAT |
| 38 | 0 | 0 | DAT |
| 39 | 0 | 0 | DAT |
| 40 | 0 | 0 | DAT |
| 41 | 0 | 0 | DAT |
| 42 | 1 | 0 | DAT |
| 43 | 10 | 1 | DAT |
| 44 | 0 | 10 | DAT |
| 45 | 0 | 0 | DAT |

**SCREENSHOT OF LMC SIMULATOR ASSEMBLER:**



**SCREENSHOT OF LMC MAILBOXES:**



**CONSOLE MESSAGE:**

1. **INP** - Input 155
2. **STA NUM** - Store the input in the location labeled NUM
3. **BRA START** - Unconditional branch to the START label
4. **LOOP** - Store NUM in the location labeled LOOP, which is not explicitly used in the code
5. **LDA QUOTIENT** - Load the QUOTIENT value
6. **ADD ONE** - Add 1 to the QUOTIENT
7. **STA QUOTIENT** - Store the result in QUOTIENT
8. **START** - Load NUM
9. **SUB TEN** - Subtract 10 from NUM
10. **BRP LOOP** - Branch to LOOP if the result is positive

At this point, the loop subtracts 10 from NUM until it becomes negative. The number of times it subtracts 10 is stored in the QUOTIENT.

1. **LDA NUM** - Load NUM
2. **LOOP1** - Store NUM in LOOP1 (unused)
3. **STA QUOTIENT1** - Store NUM in QUOTIENT1
4. **SUB TEN** - Subtract 10 from NUM
5. **BRP LOOP1** - Branch to LOOP1 if the result is positive

This loop subtracts 10 from QUOTIENT1 until it becomes negative. The number of times it subtracts 10 is stored in QUOTIENT1.

1. **LDA NUM** - Load NUM
2. **LOOP2** - Store NUM in LOOP2 (unused)
3. **STA NUM** - Store NUM in NUM
4. **SUB TEN** - Subtract 10 from NUM
5. **BRP LOOP2** - Branch to LOOP2 if the result is positive

This loop subtracts 10 from NUM until it becomes negative.

1. **LDA QUOTIENT** - Load QUOTIENT
2. **LOOP3** - Store QUOTIENT in LOOP3 (unused)
3. **STA QUOTIENT** - Store QUOTIENT in QUOTIENT
4. **LDA QUOTIENT2** - Load QUOTIENT2
5. **ADD ONE** - Add 1 to QUOTIENT2
6. **STA QUOTIENT2** - Store the result in QUOTIENT2
7. **EXIT** - Load QUOTIENT
8. **SUB TEN** - Subtract 10 from QUOTIENT
9. **BRP LOOP3** - Branch to LOOP3 if the result is positive

This loop subtracts 10 from QUOTIENT until it becomes negative.

1. **LDA NUM** - Load NUM
2. **ADD QUOTIENT1** - Add QUOTIENT1 to NUM
3. **ADD QUOTIENT2** - Add QUOTIENT2 to NUM
4. **OUT** - Output NUM (should be 11)
5. **HLT** - Halt the program