ELEMENT 2: LITTLE MAN COMPUTER

Computer Architecture

Student Name: Neil Abraham
Student Number: 21299930

Contents

Introduction: Little Man Computer........2

Programs........4

Menu algorithm:........4

Countdown program (1):........6

Triangle number sequence program (2):........6

Fibonacci sequence program (3):........7

Introduction: Little Man Computer

The Little Man Computer also known as LMC was created by DR. Stuart Madnick at MIT. LMC models Von Neumann architecture, which means that LMC has all of the basic features that is required in modern computer. It can be programmed in assemble code (machine code). By using a LMC simulator, coders can run programs that they create.

Von Neumann Architecture is a computer architecture created by a mathematician and physicist called John Von Neumann. His Architecture consists of Input – Output Unit, Control Unit, Logic Unit and Memory Unit.

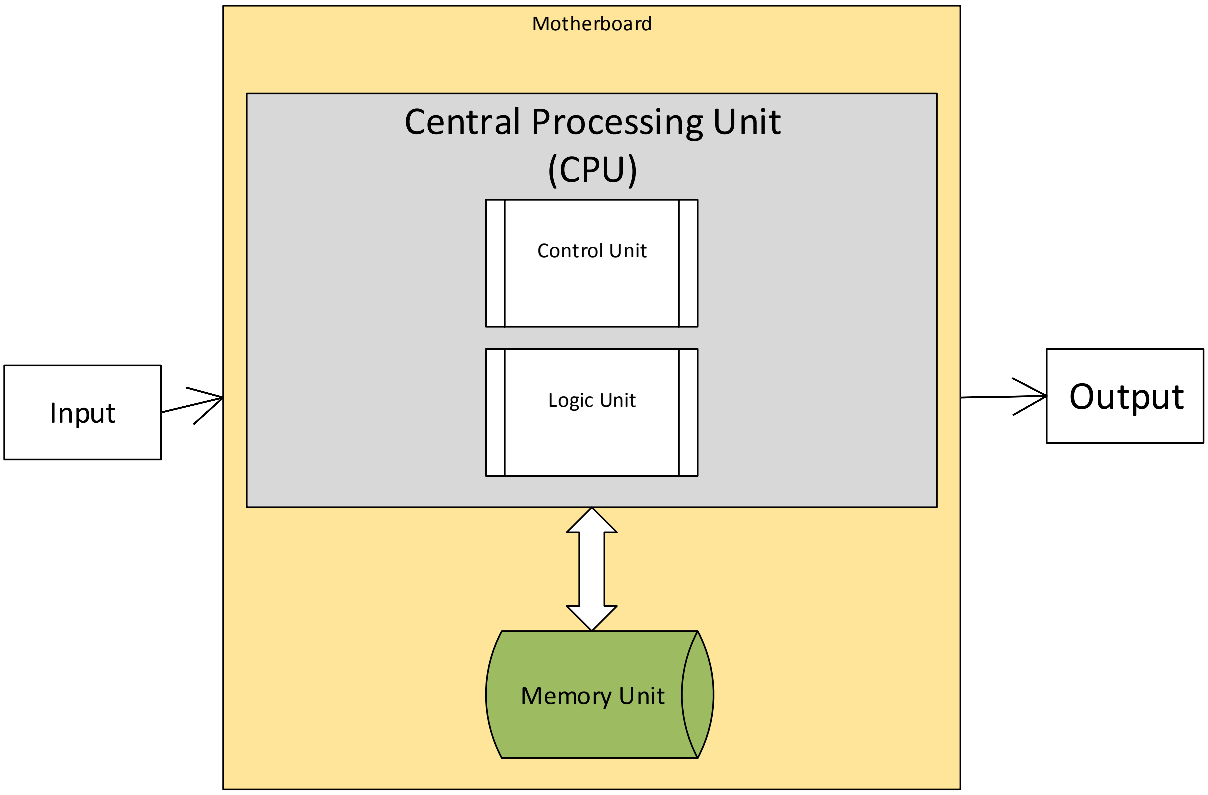
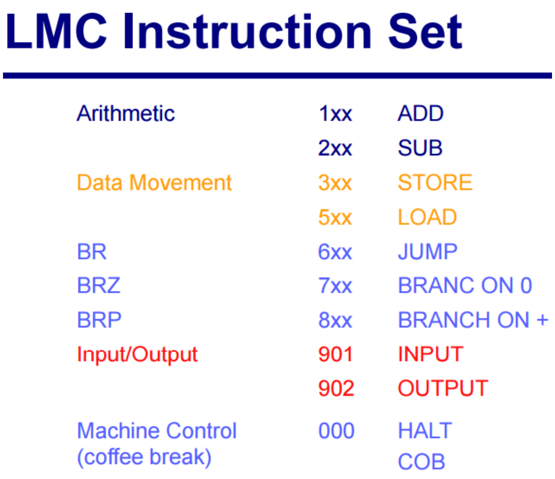


Figure 1 Vonn Neumann Architecture

The LMC works by interpreting the instruction set which contains 3-digit numeric codes or mnemonics codes (Opt. code). In this module, there are 10 different instruction sets that can be used whereas more complex and modern machines can allow programmers to carry out complicated instructions with simple code. The example of an instruction set for Little Man Computer is shown in figure 2.



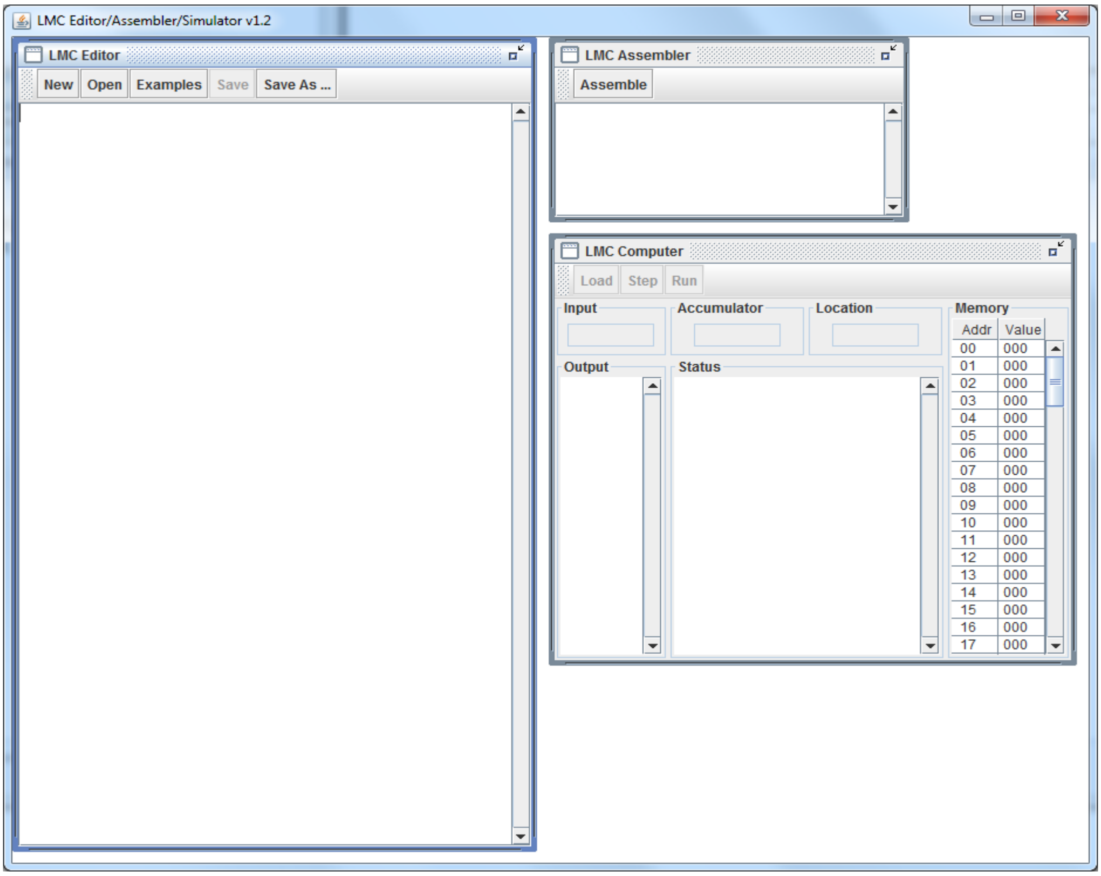
LMC Instruction Set

Arithmetic	1xx	ADD
	2xx	SUB
Data Movement	3xx	STORE
	5xx	LOAD
BR	6xx	JUMP
BRZ	7xx	BRANC ON 0
BRP	8xx	BRANCH ON +
Input/Output	901	INPUT
	902	OUTPUT
Machine Control (coffee break)	000	HALT COB

Figure 2 LMC Instruction set

The Little Man Computer simulator is used to test and write algorithms. This simulator checks if the assembly codes is right and provides feedback when assemble button is clicked. The figure 3 shows a screenshot of the LMC simulator.

- ADD: Takes the value from whatever is stored in mailbox “xy” and then adds it to the value stored in the accumulator.
- SUB: Takes the value from whatever is stored in mailbox “xy” and then subtracts it to the value stored in the accumulator.
- STO: Stores the value in the accumulator to the mailbox “xy”. This will overwrite previously stored value in mailbox.
- LDA: Loads the value stored from the mailbox “xy” and then enters it to accumulator.
- BR (branch): Jumps to mailbox “xy” and then executes whatever command is stored.
- BRZ: If the accumulator holds the value of 0, then it jumps to mailbox “xy” and then executes the command stored in the mailbox.
- BRP: If the accumulator holds positive value, then it jumps to mailbox “xy” and then executes the command stored in the mailbox.
- IN: Input a number
- OUT: Output current number from the accumulator.
- HLT and COB: Halts the program normally.
- DAT: Used to assign or declare variables.



Addr	Value
00	000
01	000
02	000
03	000
04	000
05	000
06	000
07	000
08	000
09	000
10	000
11	000
12	000
13	000
14	000
15	000
16	000
17	000

Figure 3 LMC Simulator

The memory in LMC has 99 addresses in which data is stored.

Programs

I have 3 programs that consists of Countdown, Triangle number sequence and Fibonacci sequence within menu. The menu code is used so that users can select which program they want to use or run. My algorithms run perfectly with no errors only if the value inputted is in the region of -500 to 499. Overflow is avoided if the value entered is within the range.

Menu algorithm:

The menu algorithm resets the value entered to the default value as well as enabling the user to choose on what program to execute. These algorithm is based on the user's input. Figure 4 shows how my program will work.

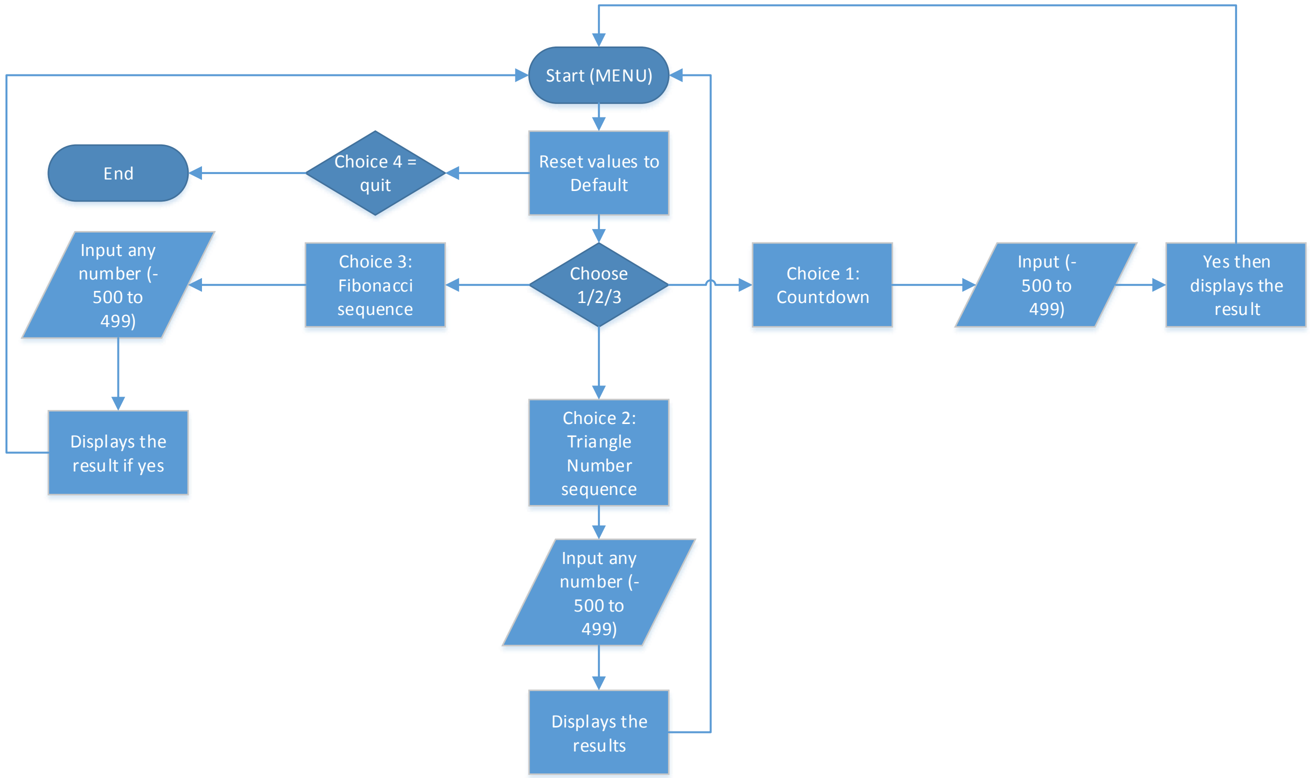

Start (MENU)
Reset values to Default
Choice 4 = quit
End
Choose 1/2/3
Choice 3: Fibonacci sequence
Input any number (-500 to 499)
Displays the result if yes
Choice 1: Countdown
Input (-500 to 499)
Yes then displays the result
Choice 2: Triangle Number sequence
Input any number (-500 to 499)
Displays the results

Figure 4 Flowchart for my program using LMC

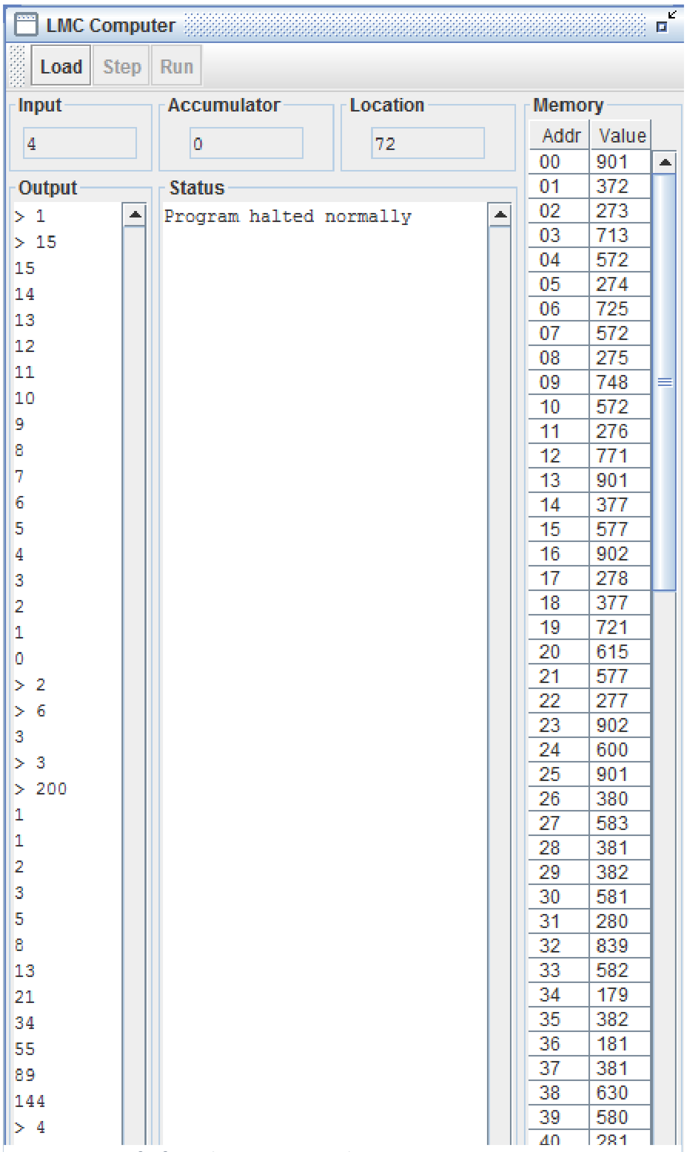
The user has to select one of four following options:

1. Countdown: This program will display numbers counting down from what the user has inputted. This program runs if the user inputs "1".
2. Tri-number: This program will display the nth term for the number which is part of the Triangle sequence($nth\ number\ in\ the\ trangle\ squence\ by\ us8ing\ the\ main\ rule\ x_n = \frac{n(n+1)}{2}$). The program runs only if the user inputs "2".
3. Fibonacci sequence: This program will display part of Fibonacci sequence but does not include the value entered as the input when the user inputs "3". This program is explained later on in this report.
4. Quit program: this program stops the algorithm from working whenever the user inputs "0".

Table 1 Menu code

Menu program with mnemonics	Opt. code	Address
Menu IN	901	00
STO ch	372	01
SUB one	273	02
BRZ choice1	713	03
LDA ch	572	04
SUB two	274	05
BRZ choice2	725	06
LDA ch	572	07
SUB three	275	08
BRZ choice3	748	09
LDA ch	572	10
SUB four	276	11
BRZ choice4	771	12

Figure 5 shows that this menu code is working.



Input: 4

Accumulator: 0

Location: 72

Output

```
> 1
> 15
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
> 2
> 6
3
> 3
> 200
1
1
2
3
5
8
13
21
34
55
89
144
> 4
```

Status

```
Program halted normally
```

Memory

Addr	Value
00	901
01	372
02	273
03	713
04	572
05	274
06	725
07	572
08	275
09	748
10	572
11	276
12	771
13	901
14	377
15	577
16	902
17	278
18	377
19	721
20	615
21	577
22	277
23	902
24	600
25	901
26	380
27	583
28	381
29	382
30	581
31	280
32	839
33	582
34	179
35	382
36	181
37	381
38	630
39	580
40	281

Figure 5 Proof of working menu code

Countdown program (1):

Table 2 Code for countdown program

Countdown code with mnemonics	Opt code	Address
IN	901	13
STO num	377	14
LOOP LDA num	577	15
OUT	902	16
SUB ONE	278	17
STO num	377	18
BRZ ENDLOOP	721	19
BR LOOP	615	20
ENDLOOP LDA num	577	21
SUB num	277	22
OUT	902	23
BR Menu	600	24

This code works and this can be seen in figure 5. In figure 5, the image shows that the user has inputted "15" after choosing to execute ("1") countdown. This program displays numbers from 15 to 0.

Triangle number sequence program (2):

The code for finding the nth term of this sequence is in table 3.

Table 3 Code for trinum program

Triangle number code with mnemonics	Opt code	address
IN	901	25
STO VALUE	380	26
LDA ZERO	583	27
STO TRINUM	381	28
STO NUM	382	29
LOOP2 LDA TRINUM	581	30
SUB VALUE	280	31
BRP ENDLOOP2	839	32
LDA NUM	582	33
ADD TWO	179	34
STO NUM	382	35
ADD TRINUM	181	36
STO TRINUM	381	37
BR LOOP2	630	38
ENDLOOP2 LDA VALUE	580	39
SUB TRINUM	281	40
BRZ EQUAL	745	41
LDA ZERO	583	42
OUT	902	43
BR DONE	647	44
EQUAL LDA NUM	582	45
OUT	902	46
DONE BR Menu	600	47

Figure 5 shows the working code when the user enters "2" in the input field and then enters an integer ("6" for this instance) the program will display the nth term for the integer that is inputted.

Fibonacci sequence program (3):

The table 4 shows the code for Fibonacci sequence program.

Table 4 code for Fibonacci sequence

Fibonacci sequence code with mnemonics	Opt code	Address
IN	901	48
STO N	386	49
LOOP3 LDA A	584	50
SUB N	286	51
BRP ENDLOOP3	863	52
LDA A	584	53
OUT	902	54
LDA B	585	55
ADD A	184	56
STO ACCUMULATOR	387	57
LDA B	585	58
STO A	384	59
LDA ACCUMULATOR	587	60
STO B	385	61
BR LOOP3	650	62
ENDLOOP3 BR LOOP3	664	63
	573	64
LOOP3 LDA ONE	384	65
STO A	385	67
STO B	387	68
STO ACCUMULATOR	583	69
LDA ZERO	600	70
STO N	000	71
BR Menu	004	72

This is a working code; this can be seen in figure 5. In figure 5 the user inputs "3" to choose the Fibonacci program to execute. The user enters "200" (200 is not part of Fibonacci sequence) and LMC displays numbers up to 144.

Stop the programs (4):

This code stops the program.

Table 5 Stop code for the program

Stop code with mnemonics	Opt code	Address
choice4 HLT	002	74

This is working code when this code is combined with other programs. This can be seen in figure 5 when the user enters choice "4" to quit the program. The LMC should display "Program halted normally", this means that user can close the Little Man Computer simulator.

Variables that contains data

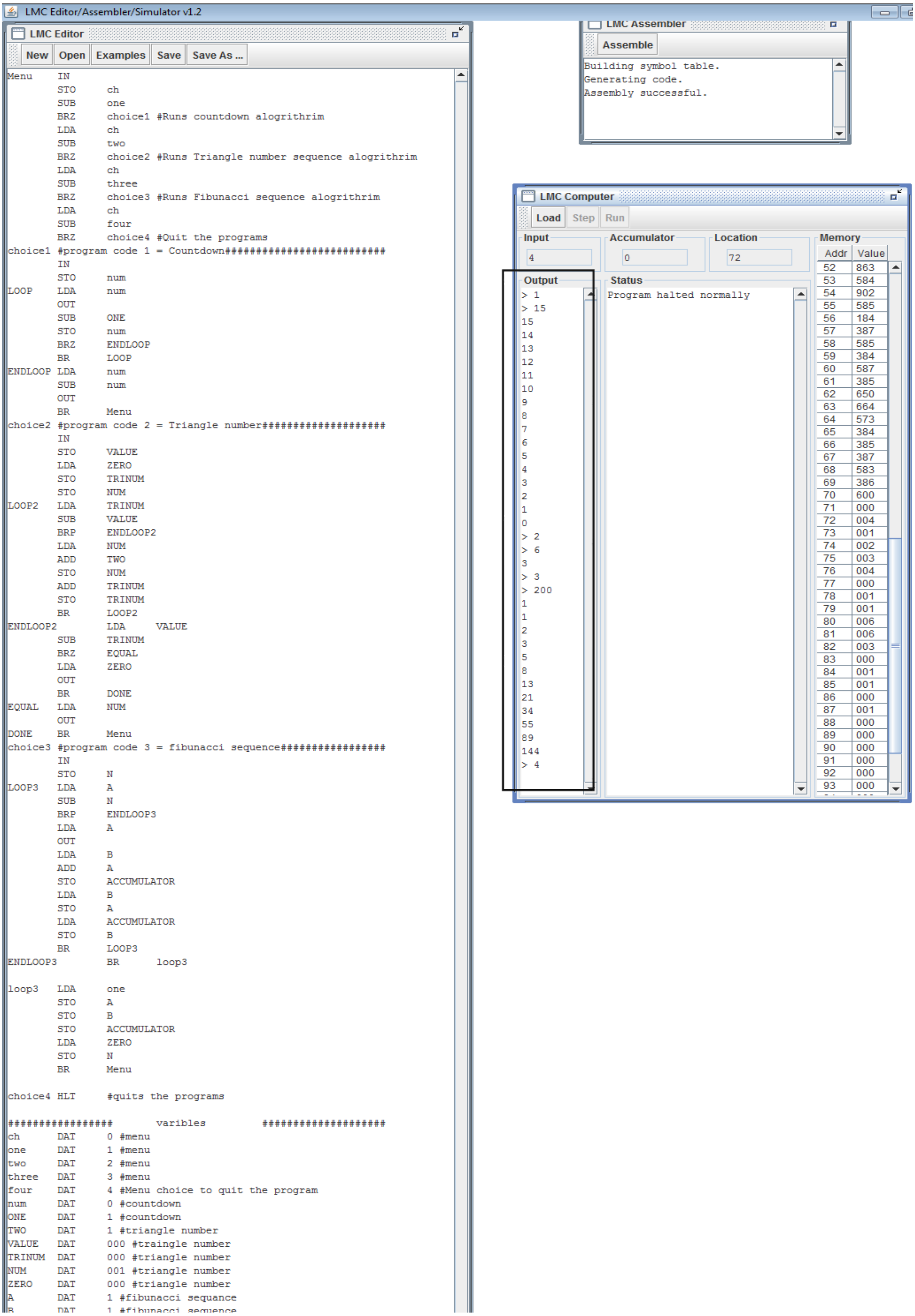
At the end of the algorithms for menu, countdown, tri-number and Fibonacci sequence, data variables are set.

Table 6 Data variable code

Variables code with mnemonics	Opt code	Address
ch DAT 0	000	77
one DAT 1	001	78
two DAT 2	001	79
three DAT 3	006	80
four DAT 4	006	81
num DAT 0	003	82
ONE DAT 1	000	83
TWO DAT 1	001	84
VALUE DAT 000	001	85
TRINUM DAT 000	000	86
NUM DAT 001	001	87
ZERO DAT 000	000	88
A DAT 1	000	89
B DAT 1	000	90
N DAT 0	000	91
ACCUMMULATOR DAT 1	000	92

This is the most important part of the code; without the data variables the program created cannot run.

Proof of algorithms execution



```
Menu    IN
        STO     ch
        SUB     one
        BRZ     choice1 #Runs countdown alogrithrim
        LDA     ch
        SUB     two
        BRZ     choice2 #Runs Triangle number sequence alogrithrim
        LDA     ch
        SUB     three
        BRZ     choice3 #Runs Fibunacci sequence alogrithrim
        LDA     ch
        SUB     four
        BRZ     choice4 #Quit the programs
choice1 #program code 1 = Countdown###########################
        IN
        STO     num
LOOP    LDA     num
        OUT
        SUB     ONE
        STO     num
        BRZ     ENDLOOP
        BR      LOOP
ENDLOOP LDA     num
        SUB     num
        OUT
        BR      Menu
choice2 #program code 2 = Triangle number#####################
        IN
        STO     VALUE
        LDA     ZERO
        STO     TRINUM
        STO     NUM
LOOP2   LDA     TRINUM
        SUB     VALUE
        BRP     ENDLOOP2
        LDA     NUM
        ADD     TWO
        STO     NUM
        ADD     TRINUM
        STO     TRINUM
        BR      LOOP2
ENDLOOP2        LDA     VALUE
        SUB     TRINUM
        BRZ     EQUAL
        LDA     ZERO
        OUT
        BR      DONE
EQUAL   LDA     NUM
        OUT
DONE    BR      Menu
choice3 #program code 3 = fibunacci sequence##################
        IN
        STO     N
LOOP3   LDA     A
        SUB     N
        BRP     ENDLOOP3
        LDA     A
        OUT
        LDA     B
        ADD     A
        STO     ACCUMULATOR
        LDA     B
        STO     A
        LDA     ACCUMULATOR
        STO     B
        BR      LOOP3
ENDLOOP3        BR      loop3

loop3   LDA     one
        STO     A
        STO     B
        STO     ACCUMULATOR
        LDA     ZERO
        STO     N
        BR      Menu

choice4 HLT     #quits the programs

#################       varibles        ####################
ch      DAT     0 #menu
one     DAT     1 #menu
two     DAT     2 #menu
three   DAT     3 #menu
four    DAT     4 #Menu choice to quit the program
num     DAT     0 #countdown
ONE     DAT     1 #countdown
TWO     DAT     1 #triangle number
VALUE   DAT     000 #traingle number
TRINUM  DAT     000 #triangle number
NUM     DAT     001 #triangle number
ZERO    DAT     000 #triangle number
A       DAT     1 #fibunacci sequance
B       DAT     1 #fibunacci sequence
```

```
Building symbol table.
Generating code.
Assembly successful.
```

Input: 4

Accumulator: 0

Location: 72

Output

```
> 1
> 15
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
> 2
> 6
3
> 3
> 200
1
1
2
3
5
8
13
21
34
55
89
144
> 4
```

Status

```
Program halted normally
```

Memory

Addr	Value
52	863
53	584
54	902
55	585
56	184
57	387
58	585
59	384
60	587
61	385
62	650
63	664
64	573
65	384
66	385
67	387
68	583
69	386
70	600
71	000
72	004
73	001
74	002
75	003
76	004
77	000
78	001
79	001
80	006
81	006
82	003
83	000
84	001
85	001
86	000
87	001
88	000
89	000
90	000
91	000
92	000
93	000

Figure 6 Proof of execution

Appendix

```
#click assemble first then click load in LMC computer. click run to test the program

#menu loop

Menu    IN

        STO     ch

        SUB     one #input 1

        BRZ     choice1 #Runs countdown alogrithrim

        LDA     ch

        SUB     two #input 2

        BRZ     choice2 #Runs Triangle number sequence alogrithrim

        LDA     ch

        SUB     three #input 3

        BRZ     choice3 #Runs Fibunacci sequence alogrithrim

        LDA     ch

        SUB     four #input 4

        BRZ     choice4 #Quit the programs


choice1 #program code 1 = Countdown#########################

        IN #input a number

        STO     num #store number

LOOP    LDA     num #load number into the loop

        OUT #emit number

        SUB     ONE #subtract ONE

        STO     num #Store number

        BRZ     ENDLOOP #branch if zero, new loop

        BR      LOOP #break

ENDLOOP         LDA     num #load number

        SUB     num #subtract number

        OUT     #display the result

        BR      Menu #back to menu loop
```

```
choice2 #program code 2 = Triangle number####################

        IN

        STO     VALUE #stores a number for value

        LDA     ZERO #loads zero

        STO     TRINUM #store number as trinum

        STO     NUM # stores another number as num

LOOP2   LDA     TRINUM #new loop and loads the trinum

        SUB     VALUE #subtracts the value

        BRP     ENDLOOP2 #breaks only if the number is positive

        LDA     NUM #loads num

        ADD     TWO #adds the number stored as two

        STO     NUM #stores the number

        ADD     TRINUM #adds trinum

        STO     TRINUM #stores trinum

        BR      LOOP2 #break the loop

ENDLOOP2        LDA     VALUE #new loop, load value

        SUB     TRINUM #subtract trinum

        BRZ     EQUAL #break if zero eqaul

        LDA     ZERO #load zero

        OUT

        BR      DONE

EQUAL   LDA     NUM

        OUT #displays the result

DONE    BR      Menu


choice3 #program code 3 = fibunacci sequence#################

        IN

    STO         N #stores number for n

LOOP3   LDA     A #loads number for a in the loop

    SUB         N #subtract number n

    BRP         ENDLOOP3
    LDA       A #load number a
    OUT
    LDA       B #load number b
    ADD       A #add a
    STO       ACCUMULATOR #store accumulator
    LDA       B #load b
    STO       A #store a
        LDA   ACCUMULATOR #load accumulator
    STO       B #store b
    BR  LOOP3 #break loop
ENDLOOP3      BR        loop3    #new loop
#####reset inputs
loop3   LDA   one
        STO   A
        STO   B
        STO   ACCUMULATOR
        LDA   ZERO
        STO   N
        BR    Menu

choice4 HLT   #quits the programs

#################     varibles     ####################
ch      DAT   0 #menu
one     DAT   1 #menu
two     DAT   2 #menu
three   DAT   3 #menu
four    DAT   4 #Menu choice to quit the program
num     DAT   0 #countdown
ONE     DAT   1 #countdown
TWO     DAT   1 #triangle number
```

```
VALUE   DAT     000 #traingle number

TRINUM          DAT     000 #triangle number

NUM     DAT     001 #triangle number

ZERO    DAT     000 #triangle number

A       DAT         1 #fibunacci sequance

B       DAT         1 #fibunacci sequence

N       DAT         0 #fibunacci sequence

ACCUMULATOR DAT     1 #fibunacci sequence
```

Reference list

1. Englander, I. (2003).*" The architecture of computer hardware and systems software: an information technology approach 3rd edition"* Bentley College.
2. Kheirkhahzadeh, A. (2016) "Week 9 to Week 12" [PowerPoint] Lecture slides related to Little Man Computer. Available at: https://online.uwl.ac.uk/webapps/blackboard/content/listContent.jsp?course_id=_82939_1&content_id=_1730544_1 (Accessed: 16 December 2016)
3.