1

```
.ORIG x3000
AND R2, R2, #0
LD R0, ★
AND R1, R1, #0
ADD R1, R1, #-1
★ ADD R2, R0, R2
ADD R0, R0, R1
BRnp ★
LD R0, ★
STR R2, R0, #0
TRAP x25
input ★ x0008
★ x3010
```

```
0010 0110
0010 0111 1111
0100 0000 0010
0000 0000 0001
1011 1111 1101
0000 0000 0011
0100 0000 0000
```

Computes $n + (n-1) + (n-2) + \cdots + 0$ where n is given @ input and stores the result at input + 1

(2)

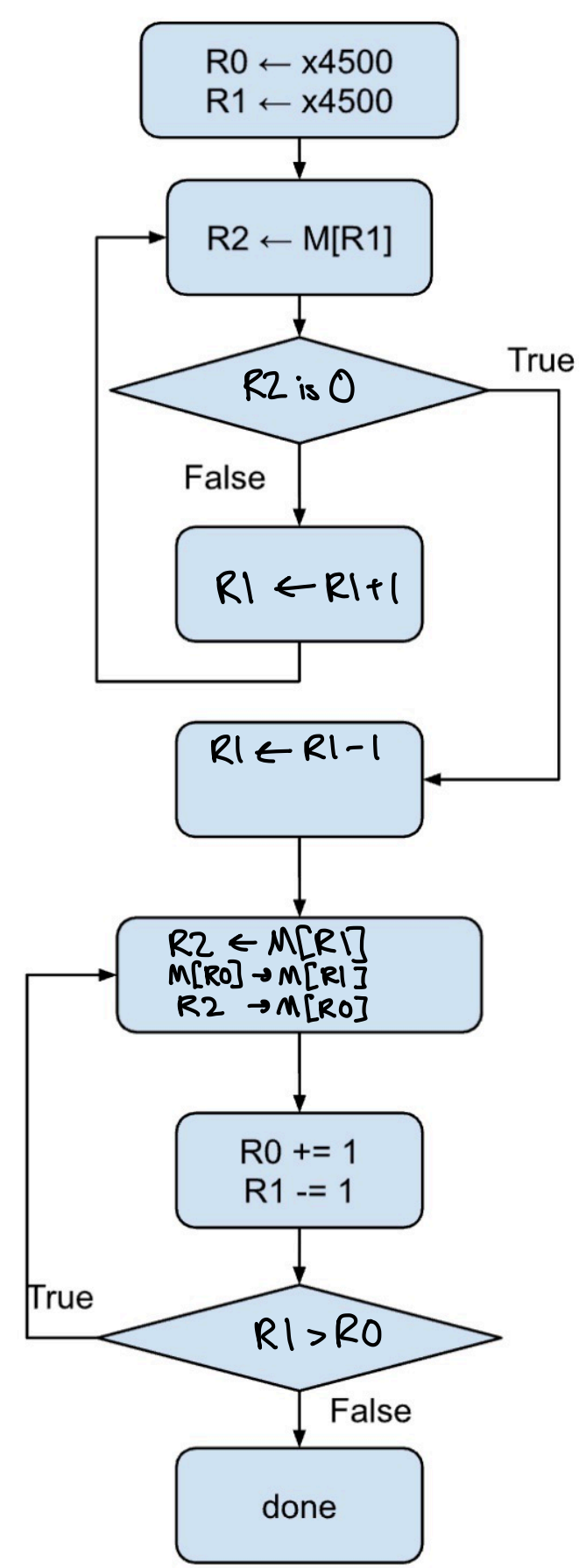
a.

Symbol	Address
Loop	x4002
End	x4009
Arr	x400A
Ct	x4014

b.

	Before	After
· x400A	0	9
· x400B	0	8
· x400C	0	7
· x400D	0	6
· x400E	0	5
· x400F	0	4
· x4010	0	3
· x4011	0	2
· x4012	0	1
· x4013	0	0
· x4014	9	-1

3


R0 ← x4500
R1 ← x4500
R2 ← M[R1]
R2 is 0
True
False
R1 ← R1+1
R1 ← R1-1
R2 ← M[R1]
M[R0] → M[R1]
R2 → M[R0]
R0 += 1
R1 -= 1
R1 > R0
True
False
done

4

```
.ORIG x3000

LD R0, ArrayPtr
LD R2, Converter
AND R3, R3, #0

Loop1S
LDR R1, R0, #0
BRz Loop1E

ADD R1, R1, R2

ADD R3, R3, R3
ADD R3, R3, R3
ADD R3, R3, R3

ADD R3, R3, R1

ADD R0, R0, #1
BRnzp Loop1S
Loop1E

STI R3, Result

HALT
ArrayPtr    .FILL x4000
Converter   .FILL #-48
Result      .FILL x5000
.END
```

5

Blanks:

```
1   NOT  R1, R1
2   LDR  R2, R0, #0
3   BRz  Misc 2
4   ADD  R5, R5, #1
5   .BLKW #1
```

6

4 1	4 2	4 3
3 2	3 3	3 5
2 1	2 2	2 3

Calculates the nth fibonacci number where n is stored in x4500 and stores the result at Result label

7

Array Interpretation	Linked-list Interpretation
DRK	ABE
BRK	ELL
CAG	CAG
ABE	FLS
GFY	BRK
ELL	GFY
FLS	DRK
HRY	HRY

8

x0002 BRnzp #2

Stuck in infinite loop