Homework 5

PB20111686 Ruixuan Huang

T1

Indicates that all values in the array are equal.

T2

No recursive exit.

> *There is a bad loop between* `SUM` *and* `JSR SUM`*.*

T3

Calculates the number of odd numbers in the array.

T4

This is a depth first search strategy. Every time we encountered the break point, `R0` stored the neighbor node address, `R2` is the color of the original node, `R7` is the color of the neighbor node.

Listing the search route as follows.

```
(1)  x6100(x0042) -> x6200(x0052)
(2)  x6200(x0052) -> x6100(x0042)
(2.1)x6200(x0052) -> x6300(x0047)
(3)  x6300(x0047) -> x6200(x0052)
(3.1)x6300(x0047) -> x6400(x0052)
(4)  x6400(x0052) -> x6100(x0042)
(4.1)x6400(x0052) -> x6300(x0047)
(4.2)x6400(x0052) -> x6500(x0047)
(5)  x6500(x0047) -> x6100(x0042)
(5.1)x6500(x0047) -> x6200(x0052)
(5.2)x6500(x0047) -> x6400(x0052)
(2.2)x6200(x0052) -> x6500(x0047)
(1.1)x6100(x0042) -> x6400(x0052)
(1.2)x6100(x0042) -> x6500(x0047)
```

And we can build the data structure.

x6100	x8000	x6300	x8000	x6500	x8000
x6101	x0042	x6301	x0047	x6501	x0047

x6102	x6200	x6302	x6200	x6502	x6100
x6103	x6400	x6303	x6400	x6503	x6200
x6104	x6500	x6304	x0000	x6504	x6400
x6105	x0000	x6305	x0000	x6505	x0000
x6106	x0000	x6306	x0000	x6506	x0000
x6200	x8000	x6400	x8000		
x6201	x0052	x6401	x0052		
x6202	x6100	x6402	x6100		
x6203	x6300	x6403	x6300		
x6204	x6500	x6404	x6500		
x6205	x0000	x6405	x0000		
x6206	x0000	x6406	x0000		

T5

```
LDR R0, R2, #-1
BRp AGAIN
```

T6

```
LEA R0, PROMPT
ADD R2, R2, R1
NOT R0, R0
ADD R0, R0, #1
ADD R1, R0, #0
ADD R3, R3, #-1
BRz TRUE
```