w ICS 2020 Problem Sheet #10:

Problem 1:

a), b)

#	HEX	Binary	Assembly Code	Description
0	2e	001 0 1110	LOAD 14	load the value in the address 14 into the accumulator
1	b0	101 1 0000	EQUAL #0	0 is not equal to the value stored in adresse 14 (06) so no instruction skipped
2	d4	110 1 0100	JUMP #4	Jump to instruction in 4
3	e0	111 0 0000	HALT 0	Should stop execution but skipped
4	2f	001 0 1111	LOAD 15	load the value in the address 15 into the accumulator
5	6f	011 0 1111	ADD 15	Add the value in the address 15 to the accumulator
6	4f	010 0 1111	STORE 15	Storing the value in the accumulator in the address 15
7	2e	001 0 1110	LOAD 14	load the value in the address 14 into the accumulator
8	91	100 1 0001	SUB #1	Subtract 1 from the value in the acumulator
9	4e	010 0 1110	STORE 14	Storing the value in the accumulator in the address 14
10	cb	110 0 1011	JUMP 11	Jump to instruction in 11
11	00	000 0 0000		
12	00	000 0 0000		
13	00	000 0 0000		
14	06	000 0 0110		
15	01	000 0 0001		

c)

The program leaves a result in memory cell 15 which is equal to 64. At first the program loads the value in memory cell 14 (which is equal to 6) into the accumulator, then if this value is equal to 0 the program should stop otherwise, it continues its execution and jumps to the execution of the instruction stored in memory cell 4 because of the assembly code Jump. Then It loads the value in memory cell 15 (which is equal to 1) into the accumulator, after that it adds the same value stored in the memory cell 15 to the accumulator which results in 1+1= 2. Then It stores this value (2) in memory cell 15. After that the program loads again the value stored in memory cell 14 (which is equal to 6) in the accumulator then it subtracts 1 from this value which results in 6-1=5.Then It stores this value (5) in memory cell 14. The program starts over from instruction stored in 0 and the value stored in memory cell 14 gets loaded again into the accumulator to compare it to 0. When the program loops for the first time this value is equal to 5 so the program goes on and loops again until the value in memory cell 14 is equal to 0. The program abstracts 1 from this value each time so at the 7th execution the value will be equal to 0 the equal #0 instruction will result in skipping the jump 4 and in executing halt 0 which stops the program. At that time the value stored in memory cell 15 will be equal to 64 because it will add its value to itself 6 times as shown in the table below:

number of executions	Value stored in memory cell 14	Value stored in memory cell 15
1	5	2
2	4	4
3	3	8
4	2	16
5	1	32
6	0	64
7	0	64

-here is a simple program in C that shows how our program works and its results when it halts:

Input:

```
#include <stdio.h>

int main()
{

    int value_in14 = 6;
    int value_in15 = 1;

    while (1)
    {

        if (value_in14 == 0)
        {
            break;
        }
        value_in15 += value_in15;
        value_in14--;
        printf("The value stored in memory cell 14: %d        ", value_in14);
        printf("The value stored in memory cell 15: %d\n", value_in15);

    }
```

Output:

```
The value stored in memory cell 14: 5        The value stored in memory cell 15: 2
The value stored in memory cell 14: 4        The value stored in memory cell 15: 4
The value stored in memory cell 14: 3        The value stored in memory cell 15: 8
The value stored in memory cell 14: 2        The value stored in memory cell 15: 16
The value stored in memory cell 14: 1        The value stored in memory cell 15: 32
The value stored in memory cell 14: 0        The value stored in memory cell 15: 64
```

d)
If the value stored in memory cell 14 is equal to 10, the program will load 10 instead of 6 in the accumulator when it loads the value in memory cell14 for the first time. The program will not stop af instruction in 1 because the value in 14 is still different from 0, but it will result in a difference when subtracting 1

from it: it will store 10-1=9 in the memory cell 14. After that the program will loop and subtract 1 from this value each time so the value stored in 14 will be equal to 0 at the 11th execution. When it is the case the program will stop its execution because of the halt 0 instruction. At that time the value stored in memory cell 15 will be equal to 1024 because it will add its value to itself 10 times as shown in the table below:

number of executions	Value stored in memory cell 14	Value stored in memory cell 15
1	9	2
2	8	4
3	7	8
4	6	16
5	5	32
6	4	64
7	3	128
8	2	256
9	1	512
10	0	1024
	0	1024

```
The value stored in memory cell 14: 9     The value stored in memory cell 15: 2
The value stored in memory cell 14: 8     The value stored in memory cell 15: 4
The value stored in memory cell 14: 7     The value stored in memory cell 15: 8
The value stored in memory cell 14: 6     The value stored in memory cell 15: 16
The value stored in memory cell 14: 5     The value stored in memory cell 15: 32
The value stored in memory cell 14: 4     The value stored in memory cell 15: 64
The value stored in memory cell 14: 3     The value stored in memory cell 15: 128
The value stored in memory cell 14: 2     The value stored in memory cell 15: 256
The value stored in memory cell 14: 1     The value stored in memory cell 15: 512
The value stored in memory cell 14: 0     The value stored in memory cell 15: 1024
```