Foundations of Computer Science – Exercise 4

1.

1: Write contents of register D to bus 2 (DC2), read contents of bus 3 (DC3) to MAR, read contents of MM address given by MAR to MDR, write 1 to bus 1 (DC1), write contents of MPC to bus 2 (DC2).

2: Write contents of MDR to bus 1 (DC1), read contents of bus 3 (DC3) to register A, write 1 to bus 1 (DC1), write contents of MPC to bus 2 (DC2).

3: Write 1 to bus 1 (DC1), result of addition in bus 3 (DC3) is multiplied by 2, read contents of bus 3 (DC3) to register C, write 1 to bus 1 (DC1), write contents of MPC to bus 2 (DC2).

4: Write contents of register C to bus 2 (DC2), write 1 to bus 1 (DC1), read contents of bus 3 (DC3) to MAR, read contents of MM address given by MAR to MDR, write 1 to bus 1 (DC1), write contents of MPC to bus 2 (DC2).

5: Write contents of register A to bus 2 (DC2), write contents of MDR to bus 1 (DC1), c10 = read contents of bus 3 (DC3) to register B, write 1 to bus 1 (DC1), write contents of MPC to bus 2 (DC2).

1: 0+D -> MAR; MAR->MDR; 1+MPC -> MPC
2: MAR+0 -> A; ; 1+MPC -> MPC
3: (1+0)*2->C; ; 1+MPC -> MPC
4: 1+C->MAR; MAR->MDR; 1+MPC -> MPC
5: A+MDR -> B; ; 1+MPC ->MPC

It reads data of D to A, write 2 to C. Then add 1 to C and write to MDR. Write the sum of MDR and A to B. Therefore, B = D+3

2.

0: 0+D-> A; ; 1+MPC -> MPC
1: 0+0-> B; ; 1+MPC -> MPC
2: ; ; (A = 0)+MPC -> MPC
3: ; ; 1000 -> MPC

4: 0+A -> MDR; ; 1+MPC -> MPC
5: MDR+B -> B; ; 1+MPC -> MPC
6: -1+A -> A; ; 1+MPC -> MPC
7: ; ; 10 -> MPC
8: 0+B -> A

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1				1					1								1					1
2										1							1					1
3																			1			1
4					1													1				
5	1												1									1
6		1				1				1							1					1
7					1		1		1								1					1
8		1							1													

3.

```
class FP:
    global b, p
    b = 10
    p = 8

    def __init__(self, e, f):
        self.e = e
        self.f = f

    @staticmethod
    def sum(u, v):
        w = FP(0, 0)
        if u.e > v.e:
            w.e = u.e
            w.f = u.f + v.f / (b**(u.e - v.e))
        else:
            w.e = v.e
            w.f = v.f + u.f / (b**(v.e - u.e))
        w.f = w.f.__round__(8)
        return w
```

The workload will be more intense as floating points require more multiplication, division and comparison.

4.

a) D = 0, MDR = 11, A=109
First While loop:
C = 11, 22, 44, 88
D = 1, 2, 3, 4
MDR = 22, 44, 88, 176
MDR = 88, C = 0
Second While loop:
D = 3, 2, 1, 0
C = (0, 1), 2, 4, (8,9)
A = (21, 42), 84, 168, 80
Final results: C = 9

b) D , which is how many times on the base of 2 dividend is bigger than divisor. The number of iterations would be the minimum correct integer answer of the in the following statement:
MDR >= A * 2^D