Byeongchan Gwak, Student ID : 501026

Washington University in St.Louis

McKelvey School of Engineering

Fall Semester 2021

CSE467M: Embedded Computing Systems

Homework #4

Chapter 4 problems:

1) **Q5-2 (25 points)**

```
double getAverage() {
        double avg = 0;
        int sum = 0;
        int newValue = getNewValue(); // get new value from a function
        circ_update(newValue);          // put the new value into circular buffer
        for(i=0;i<CMAX;i++) {           // get summation from circular buffer
                sum += circ_get(i);
        }
        avg = (double)sum/(double)CMAX;     // get average value
        return avg;
}
```

Byeongchan Gwak, Student ID : 501026

2) Q5-4 (20 points)

a.

x = a + b;

y = c + d;

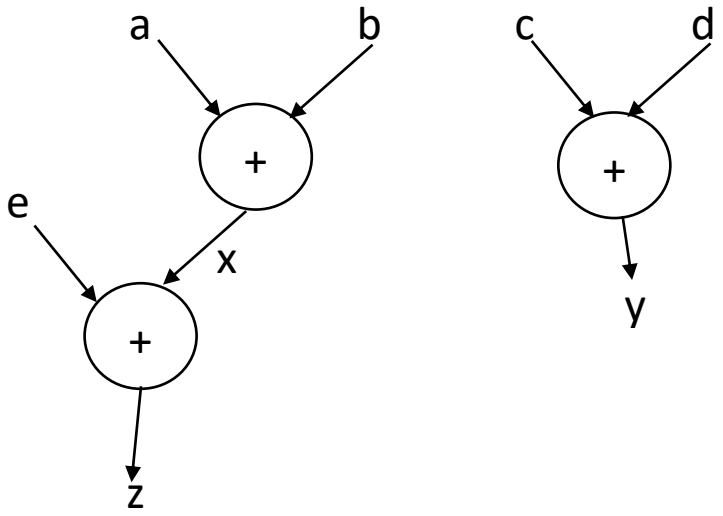
z = x + e;

single assigned form

x = a + b;

y = c + d;

z = x + e;



Byeongchan Gwak, Student ID : 501026

b.

r = a + b - c;

s = 2 * r;

t = b - d;

r = d + e;

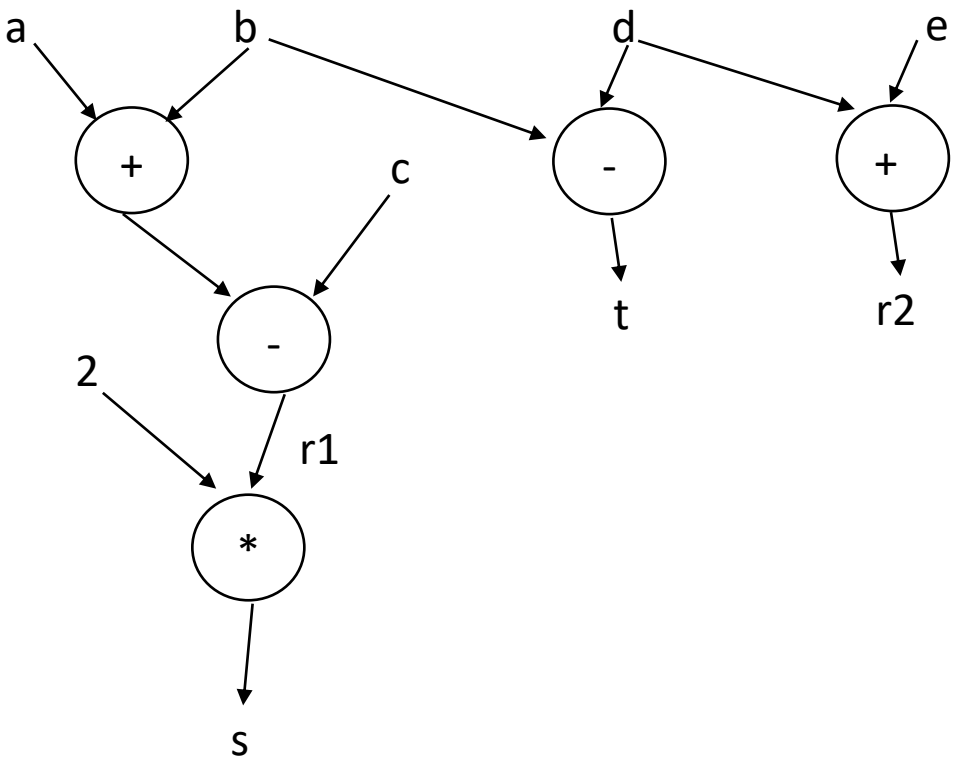
single assigned form

r1 = a + b - c;

s = 2 * r1;

t = b - d;

r2 = d + e;



Byeongchan Gwak, Student ID : 501026

c.

a = q - r;

b = a + t;

a = r + s;

c = t - u;

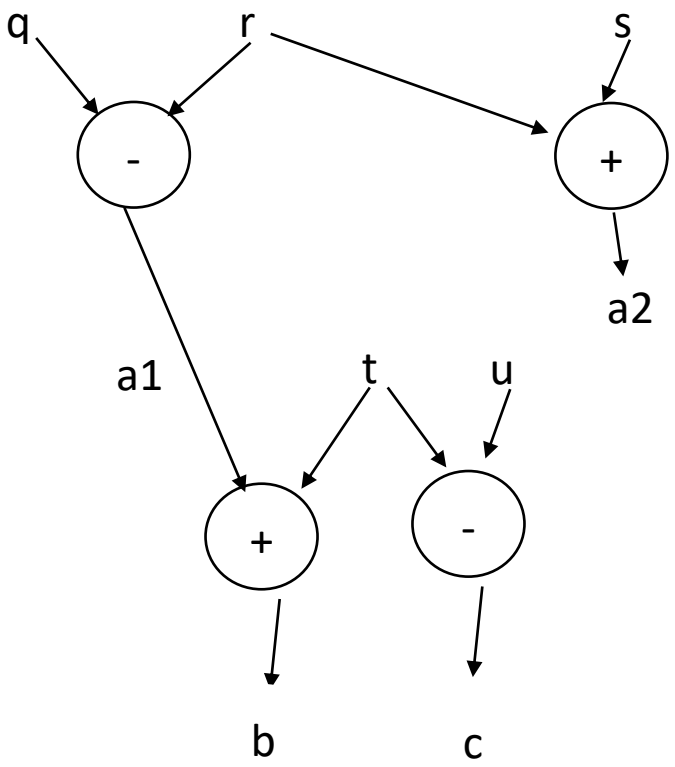
single assigned form

a1 = q - r;

b = a1 + t;

a2 = r + s;

c = t - u;



Byeongchan Gwak, Student ID : 501026

d.

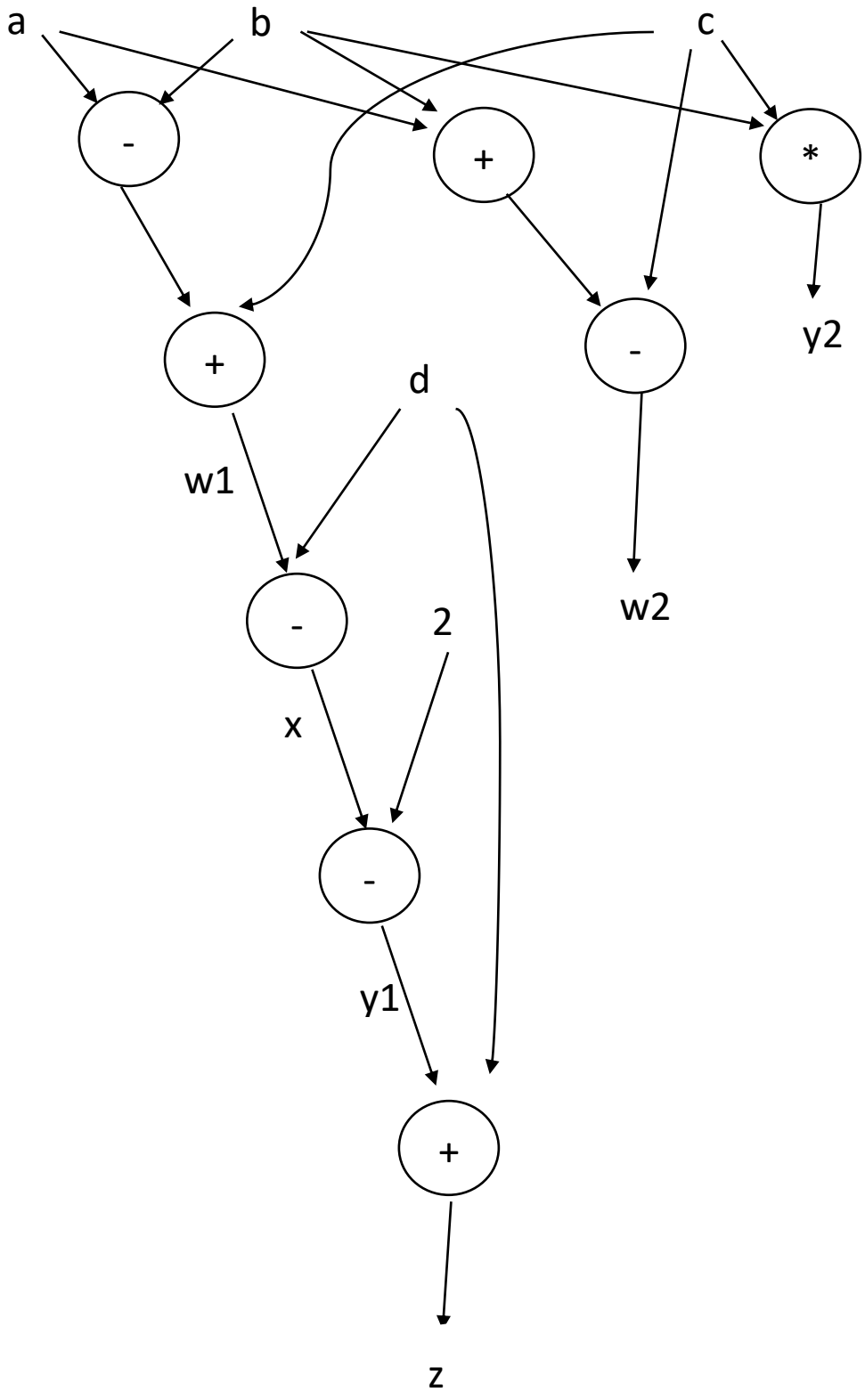
```
w = a - b + c;
x = w - d;
y = x - 2;
w = a + b - c;
z = y + d;
y = b * c;
```

single assigned form

```
w1 = a - b + c;
x = w1 - d;
y1 = x - 2;
w2 = a + b - c;
z = y1 + d;
y2 = b * c;
```

Please see the next page for the graph.

Byeongchan Gwak, Student ID : 501026



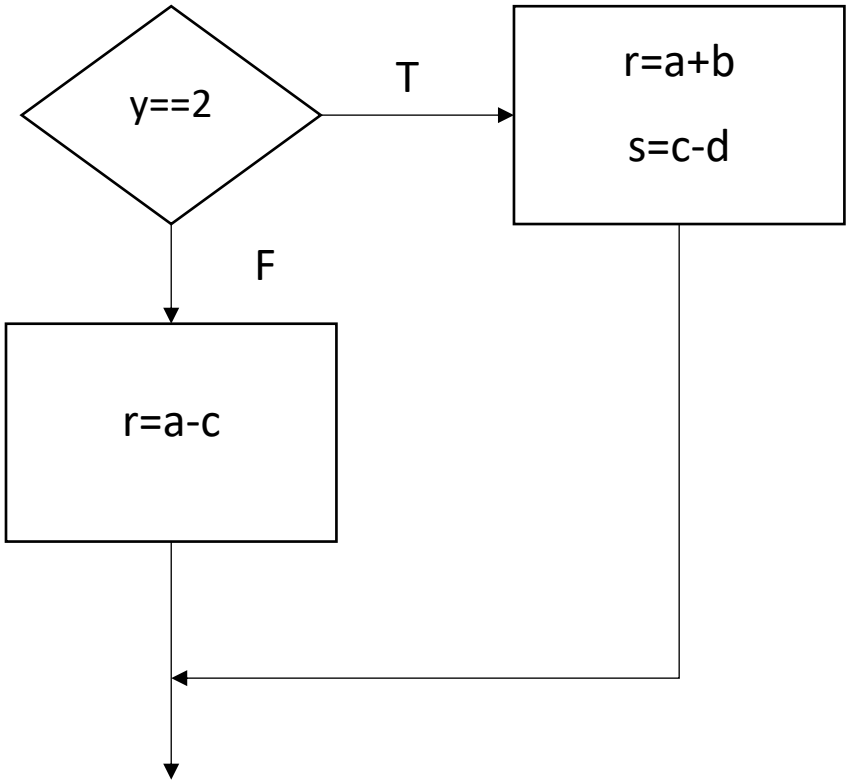
Byeongchan Gwak, Student ID : 501026

3) Q5-5 (25 points)

a.

if (y == 2) {r = a + b; s = c - d;}

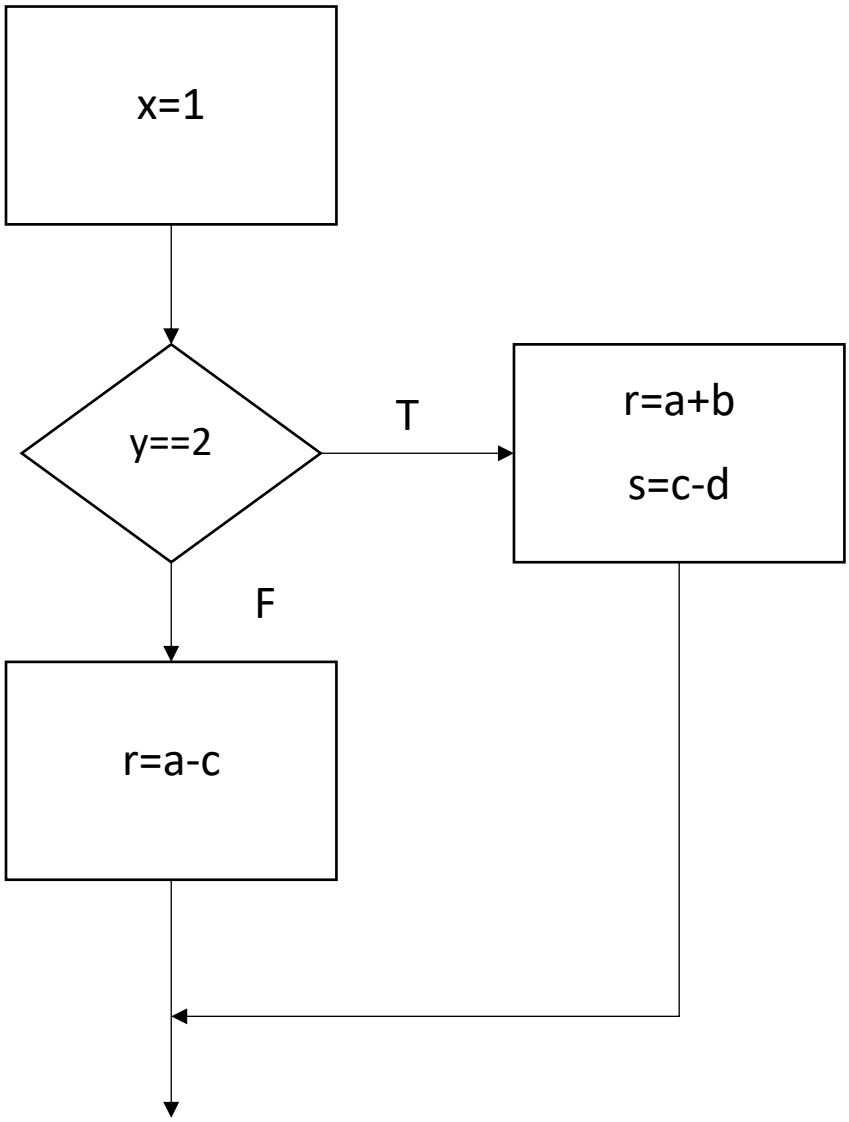
else r = a - c



Byeongchan Gwak, Student ID : 501026

b.

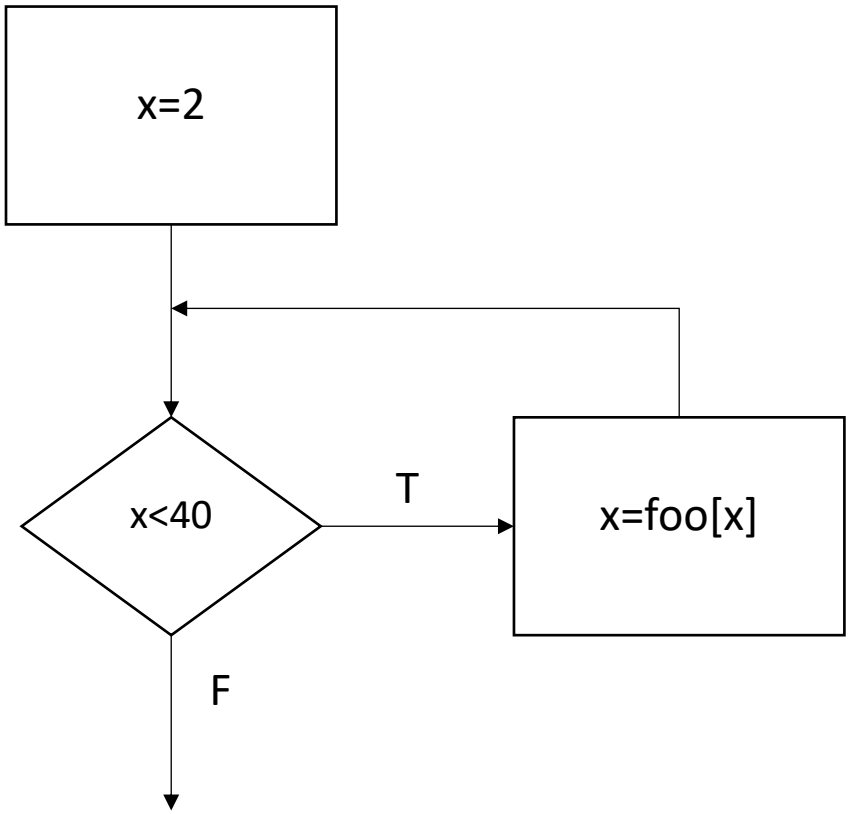
```
x = 1;
if (y == 2) { r = a + b; s = c - d; }
else { r = a - c; }
```



Byeongchan Gwak, Student ID : 501026

c.

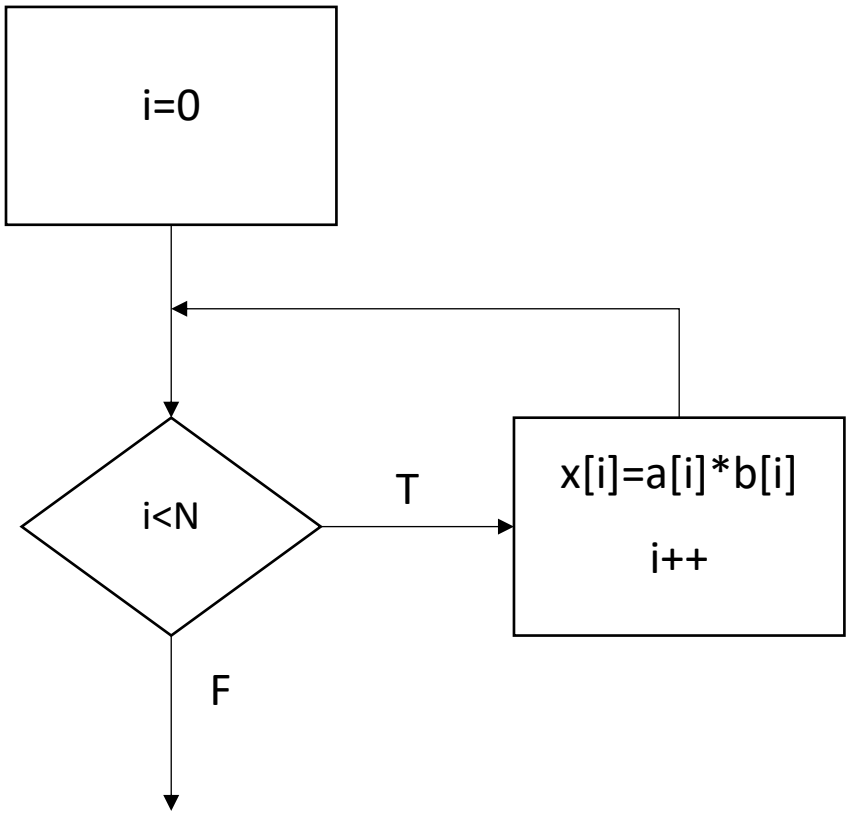
```
x = 2;
while (x <40) {
        x = foo[x];
}
```



Byeongchan Gwak, Student ID : 501026

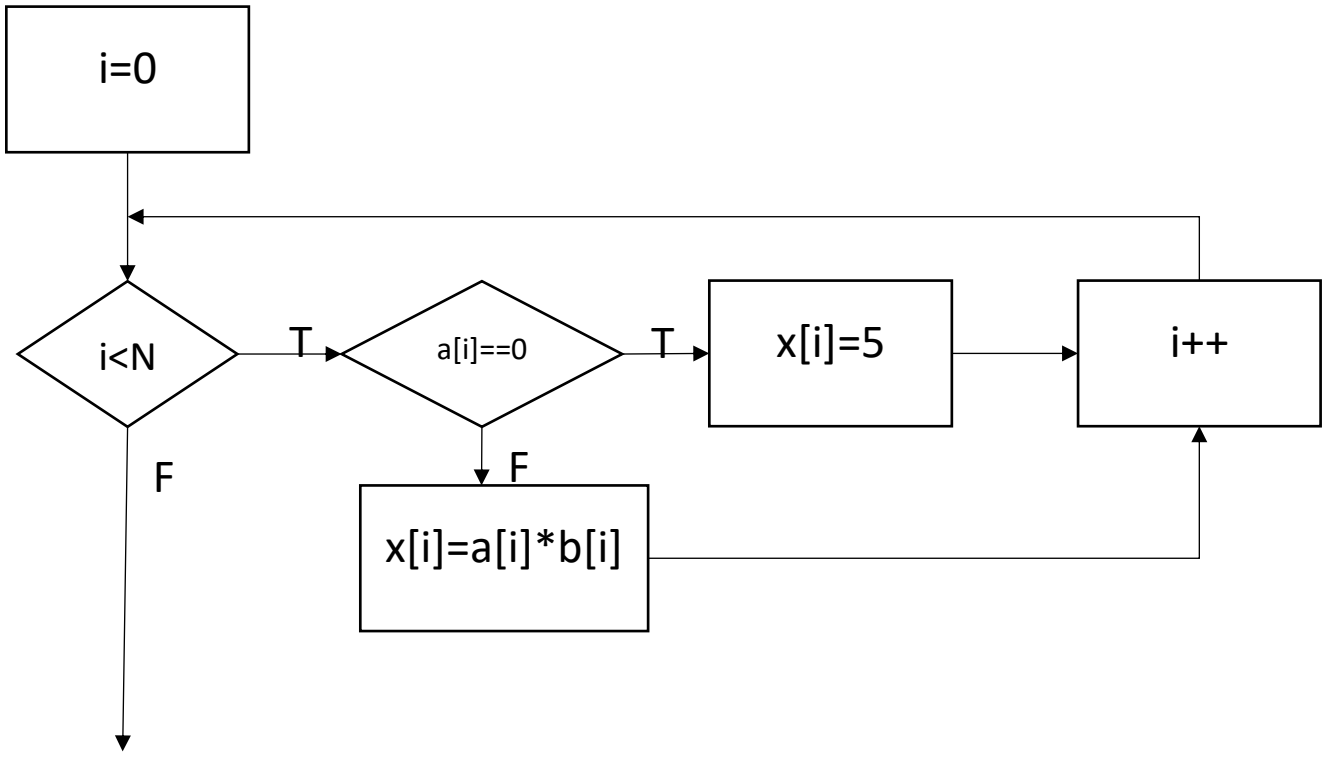
d.

```
for(i=0;i<N;i++) {
        x[i]=a[i]*b[i]
}
```



e.

```
for(i=0;i<N;i++) {
        if(a[i]==0)
           x[i] = 5;
        else
           x[i] = a[i]*b[i];
}
```



Byeongchan Gwak, Student ID : 501026

4) Q5-6 (15 points)

a.

```
    ORG 200
p1: ADR r4,a
    LDR r0,[r4]
    ADR r4,e
    LDR r1,[r4]
    ADD r0,r0,r1
    CMP r0,r1
    BNE q1
p2: ADR r4,e
```

I'll answer to this question in base 16.

Symbol table:

p1 = 0x200

p2 = 0x21C

b.

```
        ORG 100
p1:  CMP r0,r1
        BEQ x1
p2:  CMP r0,r2
        BEQ x2
p3:  CMP r0,r3
        BEQ x3
```

I'll answer to this question in base 16.

Symbol table:

p1 = 0x100

p2 = 0x108

p3 = 0x110

c.

```
        ORG 200
S1:   ADR r2,a
        LDR r0,[r2]
S2:   ADR r2,b
        LDR r2,a
        ADD r1,r1,r2
```

I'll answer to this question in base 16.

Symbol table:

S1 = 0x200

S2 = 0x208

5) Q5-8 (15 points)

a.

```
int p1(int a, int b) {
        return(a + b);
}
```

It's reentrant. p1 function does not change global variables.

b.

```
int x, y;
int p2(int a) {
        return a + x;
}
```

It's reentrant. p2 function does not change global variables.

c.

```
int x, y;
int p3(int a, int b) {
        if (a > 0)
                x = b;
        return a + b;
}
```

It's NOT reentrant. p3 function changes global variable x.

Byeongchan Gwak, Student ID : 501026

6) Q5-10 (15 points)

a.

{a+b,c+d}

(a+b) – (c+d)

((a+b) – (c+d)) + e

b.

{a+b, d+e}

{(a+b) - c, (d+e) + f}

c.

{a+b, c+d}

(a+b) + c

((a+b) + c) + f

g = (((a+b) + c) + f)+ (c+d)

Byeongchan Gwak, Student ID : 501026

7) Q5-12 (10 points)

```
for (i = 0; i < 32; i++)
x[i] = a[i] * c[i];
```

a. Unroll two times :

```
for(i=0; i<16; i++)
{
    x[(2*i)]   = a[(2*i)]   * c[(2*i)];
    x[(2*i)+1] = a[(2*i)+1] * c[(2*i)+1];
}
```

b. Unroll three times :

```
for(i=0; i<10; i++)
{
    x[(3*i)]   = a[(3*i)]   * c[(3*i)];
    x[(3*i)+1] = a[(3*i)+1] * c[(3*i)+1];
    x[(3*i)+2] = a[(3*i)+2] * c[(3*i)+2];
}
x[30] = a[30] * c[30];
x[31] = a[31] * c[31];
```

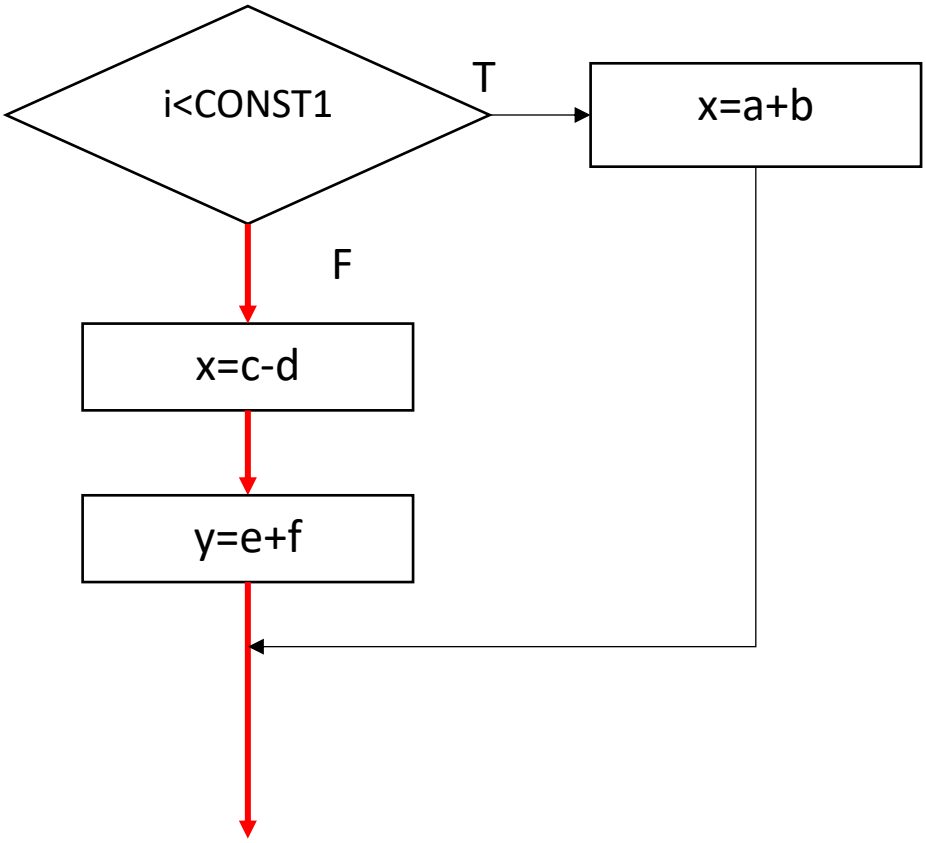
Byeongchan Gwak, Student ID : 501026

8) Q5-18 (15 points)

a. The longest path is the FALSE case of if statement.

if (i < CONST1) { x = a + b; }

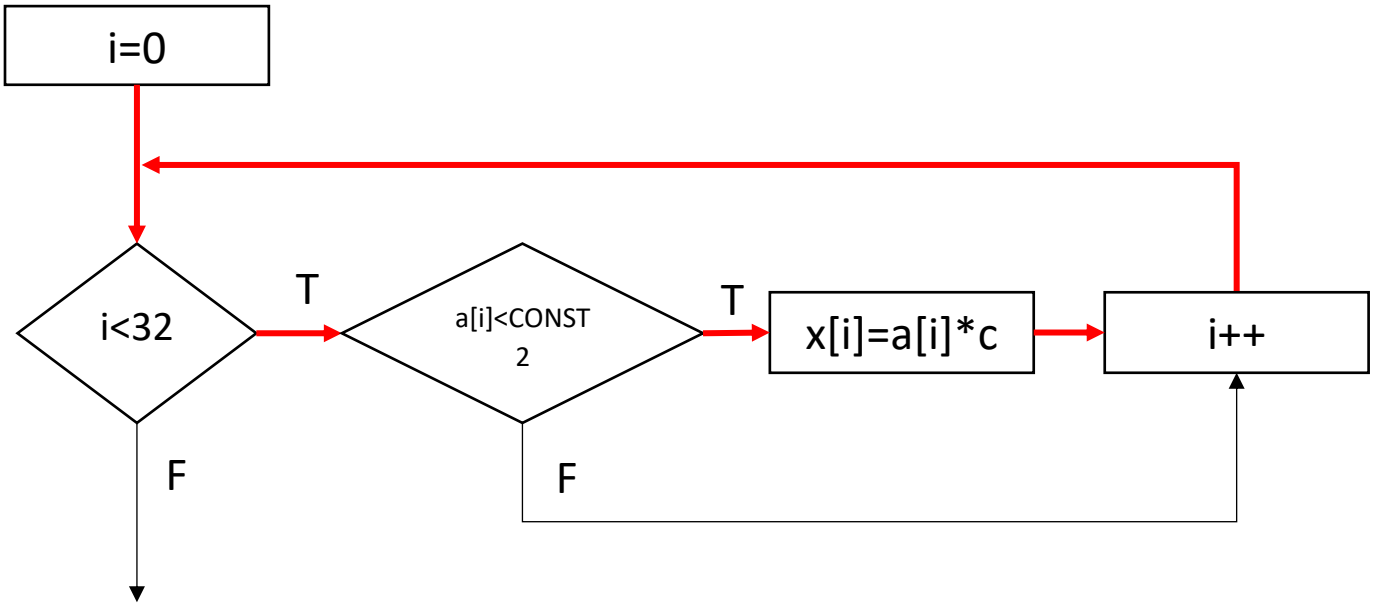
else { x = c - d; y = e + f; }



Byeongchan Gwak, Student ID : 501026

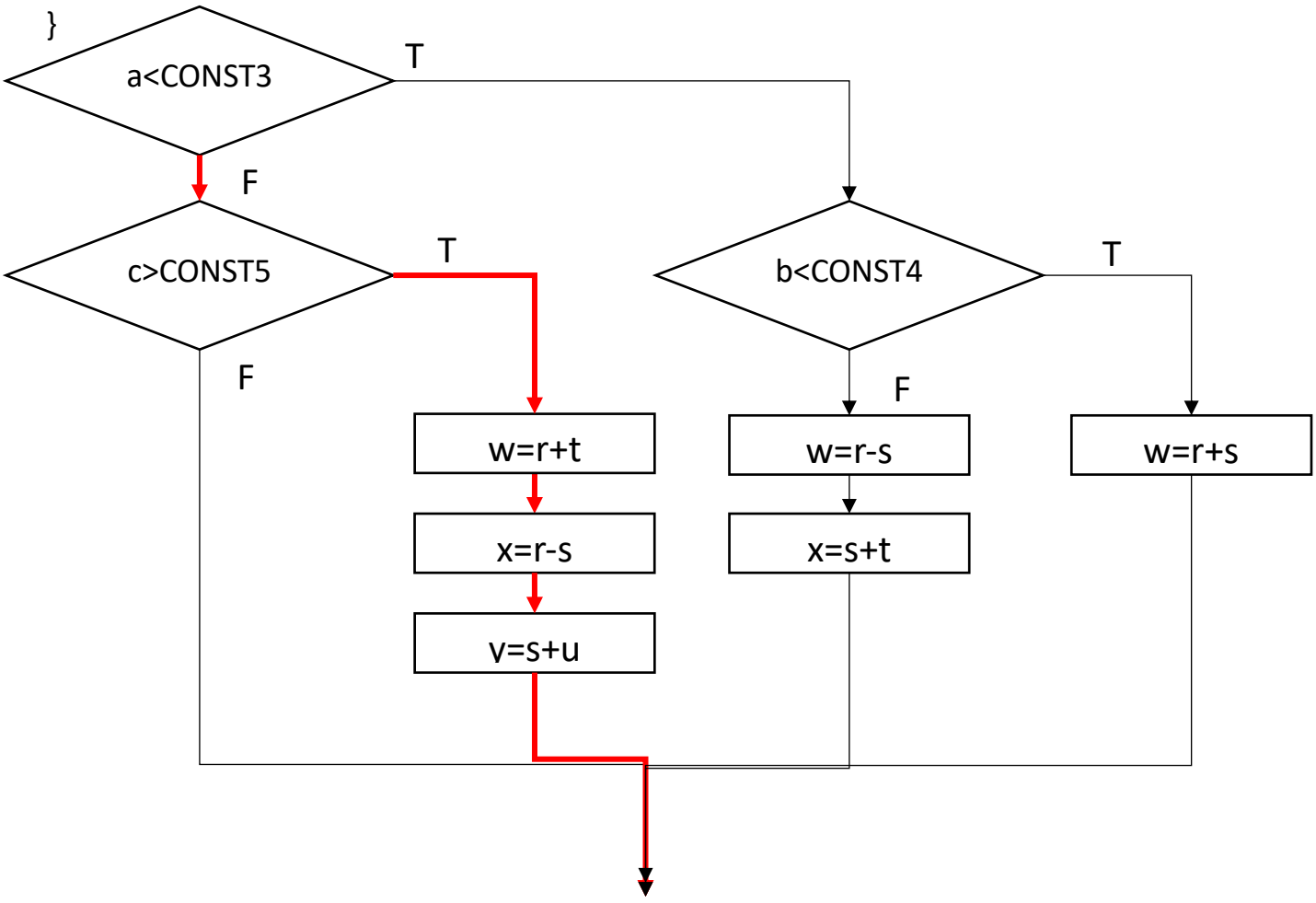
b. The longest path is the TRUE case of if statement inside of for statement.

```
for (i = 0; i < 32; i++)
        if (a[i] < CONST2)
                x[i] = a[i] * c[i];
```



c. The longest path is the first FALSE and the second TRUE statements

```
if (a < CONST3) {
        if (b < CONST4)
                w = r + s;
        else {
                w = r - s;
                x = s + t;
        }
} else {
        if (c > CONST5) {
                w = r + t;
                x = r – s;
                y = s + u;
        }
}
```



Byeongchan Gwak, Student ID : 501026

9) Q5-21 (40 points)

a. maximum number

B1: 1 Times(initialize)

B2: 17 Times(loop test)

B3: 16 Times(x[i] test)

B4: 16 Times(body)

B5: 16 Times(i++)

b. minimum number

B1: 1 Times(initialize)

B2: 17 Times(loop test)

B3: 16 Times(x[i] test)

B4: 0 Times(body), when x[i] > 2 is false

B5: 16 Times(i++)

c. maximum execution time

B1 : 1 * 6 = 6

B2 : 16 * 2 + 1 * 5 = 37

B3 : 16 * 3 = 48,

B4 : 16 * 7 = 112

B5 : 16 * 1 = 16

Total = 6 + 37 + 48 + 112 + 16 = 219 cycles

d. minimum execution time

B1 : 1 * 6 = 6

B2 : 16 * 2 + 1 * 5 = 37

B3 : 16 * 3 = 48,

B4 : 0

B5 : 16 * 1 = 16

Total = 6 + 37 + 48 + 0 + 16 = 107 cycles

Byeongchan Gwak, Student ID : 501026

10) Q5-24 (60 points)

Since the size of cache is 1K words= 1024 words with 4 words per line , hence number of cache lines = 1024/4 = 256 cache lines.

a. To produce a conflict miss every time, the cache has to be replaced whenever ‘x’ array and ‘a’ array are referenced. To do that address gap between the a and x should be a multiple of 1024. As a result, a[i][j] and x[i][j] are mapped to the same location in the cache for every referece.

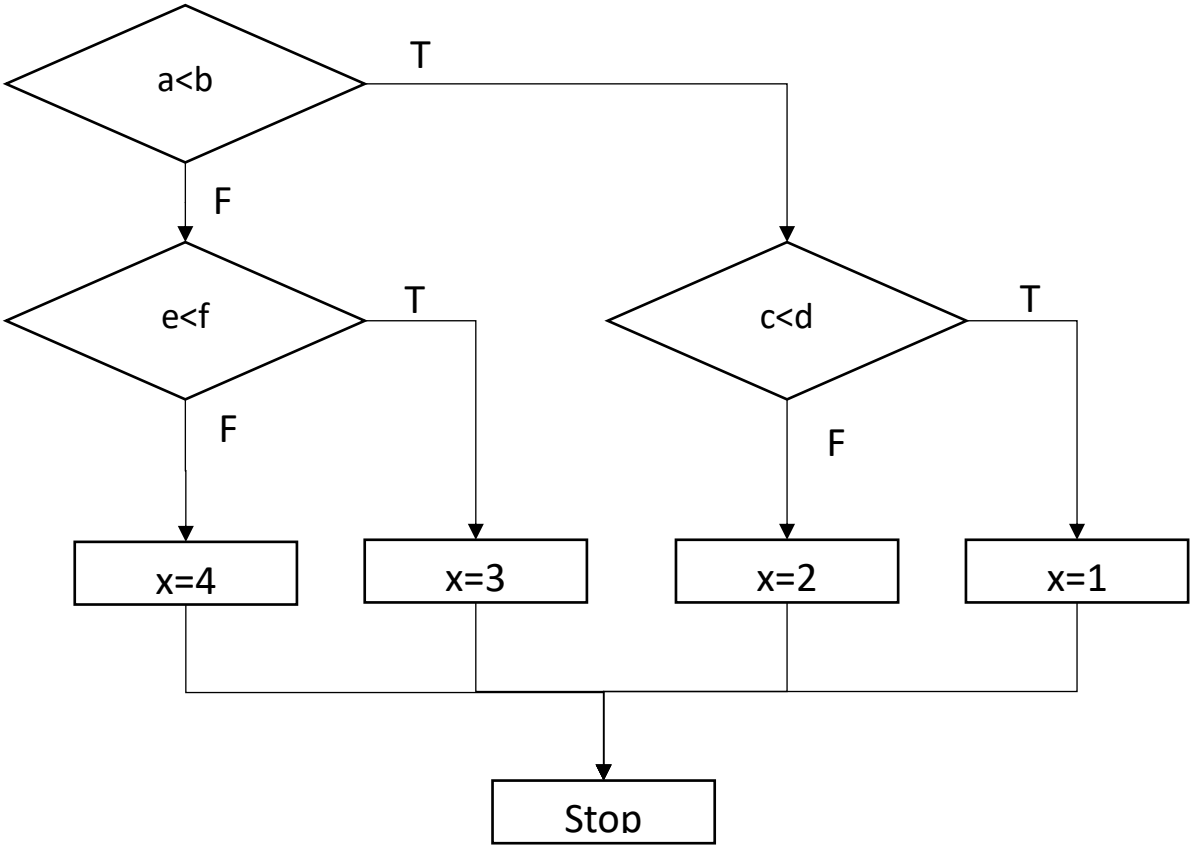
b. To produce a conflict miss one out of every four times, we need more address gap to hold the inner loop memory space. If the gap is 1024*k+3, the conflict miss will happen one out of every four times. Because j is referring from 0 to 3 so there must be memory space for 4 words to hold the one inner loop.

c. The size of whole loop array is 50 * 4 = 200 words. So, if the gap between the ‘x’ array and ‘a’ array is more than 200 words and then there will not be a conflict miss. Since we have 1024 cache memory, the upper bound should be lower than 1024. As a result the address gap between two array is 1024 * k + x, where $200 < x < 1024$.

Byeongchan Gwak, Student ID : 501026

11) Q5-26 (60 points)

a.



a.The cyclomatic complexity = e-n+2p

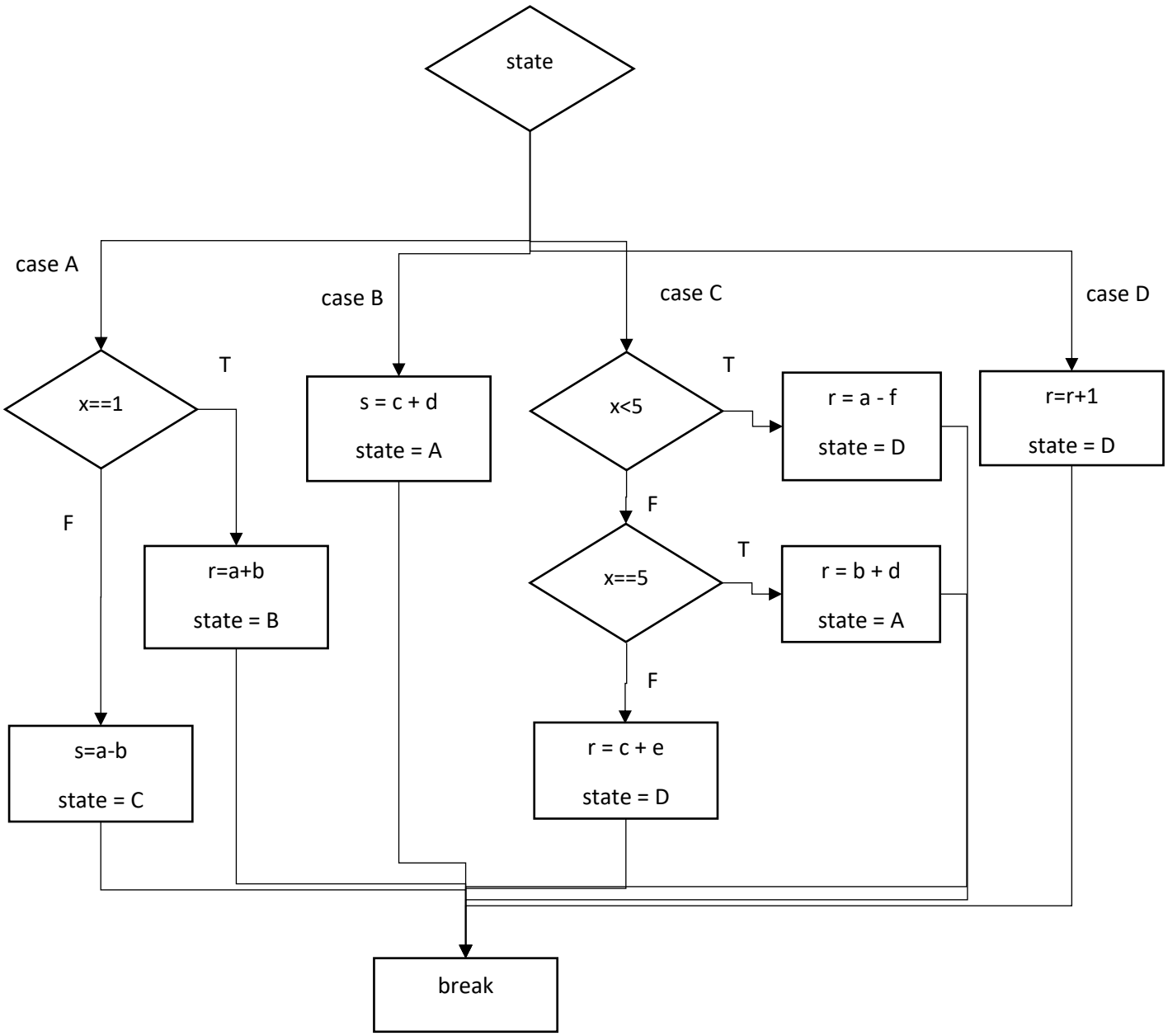
e: number of edges, n: number of nodes, p: number of component

e = 10, n = 8, p = 1

10 – 8 + 2 = 4

Byeongchan Gwak, Student ID : 501026

b.



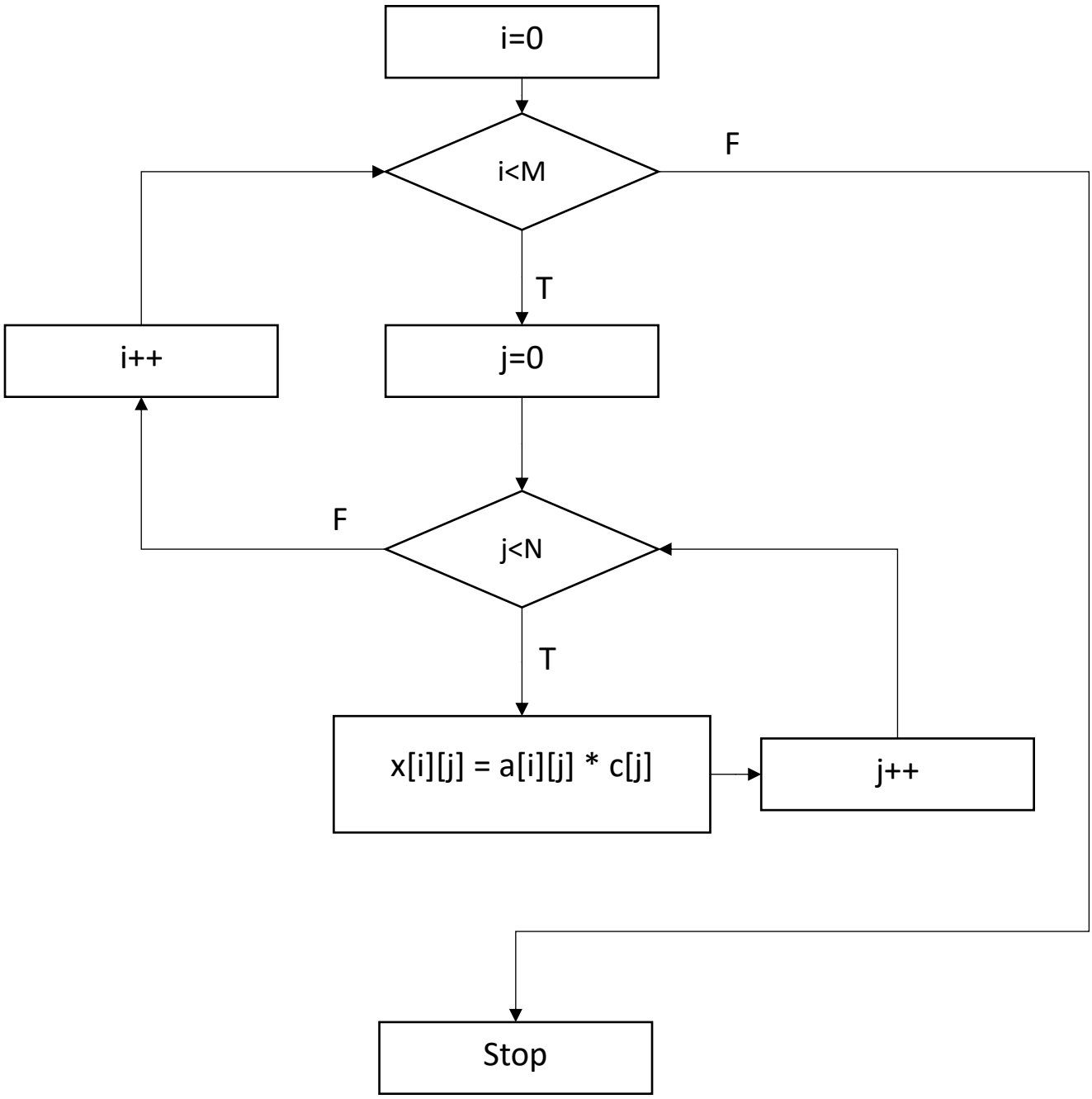
b.The cyclomatic complexity = e-n+2p

e: number of edges, n: number of nodes, p: number of component

e = 17, n = 12, p = 1

17 – 12 + 2 = 7

c.



b.The cyclomatic complexity = e-n+2p

e: number of edges, n: number of nodes, p: number of component

e = 9, n = 8, p = 1

9 – 8 + 2 = 3

12) Raspberry Pi Circuit Card Assembly (CCA) Laboratory #2– Measuring code execution time (500 points)

For this laboratory you need to do the following tasks:

12.1) Write C program (or Python program) that does the following mathematical functions:

```
for ( i = 0, f = 0, g = 0; i<20; i++) {
        f = f + a[i]*b[i] + c[i]*d[i] - e[i];
        g = g + (a[i]+d[i])*(b[i] - c[i])*e[i];
}

printf(f);
printf(g);
```

Arrays a, b, c, d and e are shown here:

a	b	c	d	e
1	3	2	5	10
3	7	5	10	25
5	11	8	15	40
7	15	11	20	55
9	19	14	25	70
11	23	17	30	85
13	27	20	35	100
15	31	23	40	115
17	35	26	45	130
19	39	29	50	145
21	43	32	55	160
23	47	35	60	175
25	51	38	65	190
27	55	41	70	205
29	59	44	75	220
31	63	47	80	235
33	67	50	85	250
35	71	53	90	265
37	75	56	95	280
39	79	59	100	295

Specify all arrays inside the main program.

Byeongchan Gwak, Student ID : 501026

Turn in copy of your code in your homework.

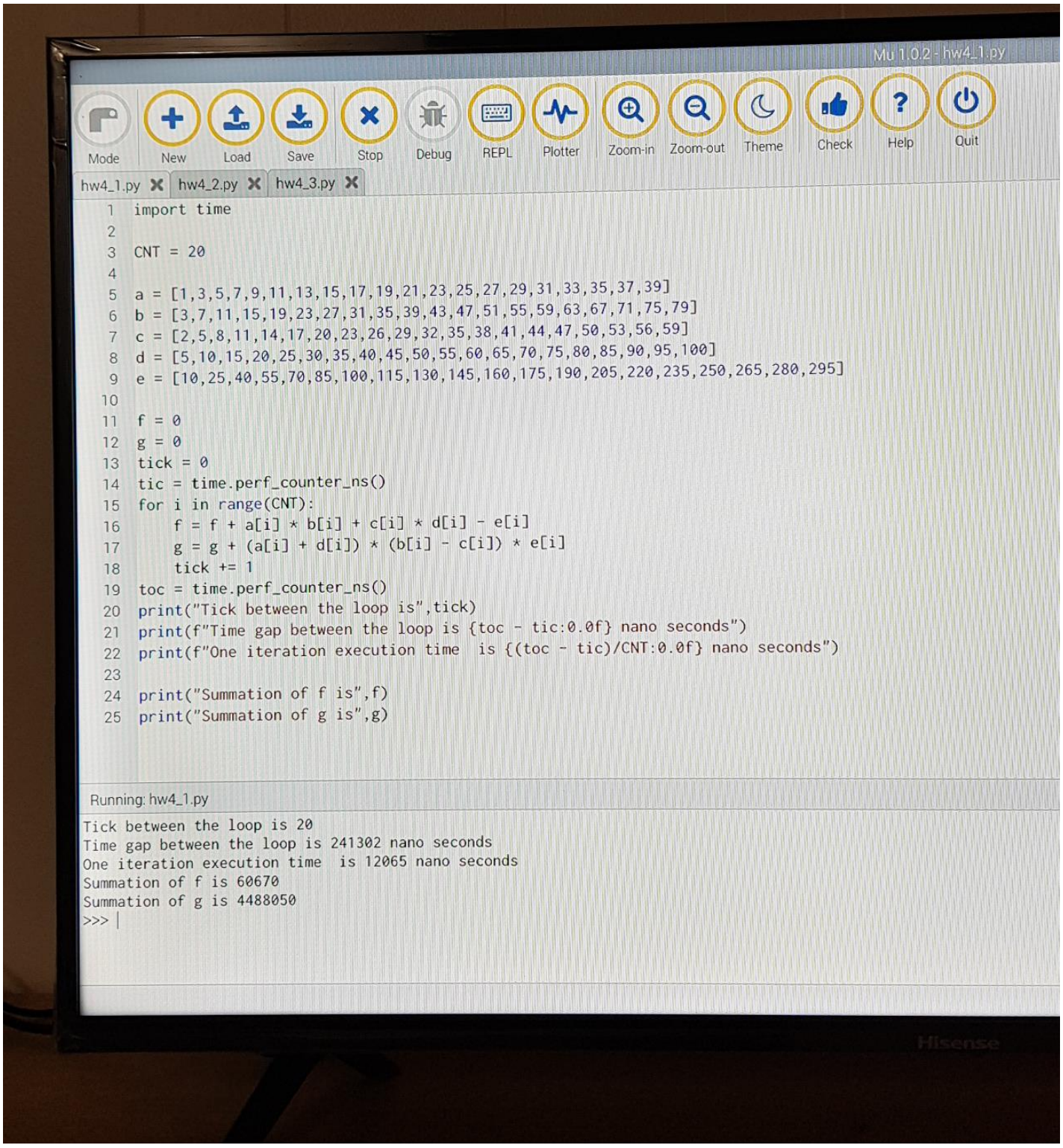
My code is shown below.

```
import time

a = [1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39]
b = [3,7,11,15,19,23,27,31,35,39,43,47,51,55,59,63,67,71,75,79]
c = [2,5,8,11,14,17,20,23,26,29,32,35,38,41,44,47,50,53,56,59]
d = [5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100]
e = [10,25,40,55,70,85,100,115,130,145,160,175,190,205,220,235,250,265,280,295]

f = 0
g = 0
tick = 0
tic = time.perf_counter_ns()
for i in range(20):
    f = f + a[i] * b[i] + c[i] * d[i] - e[i]
    g = g + (a[i] + d[i]) * (b[i] - c[i]) * e[i]
    tick += 1
toc = time.perf_counter_ns()
print("Tick between the loop is",tick)
print(f"Time gap between the loop is {toc - tic:0.0f} nano seconds")

print("Summation of f is",f)
print("Summation of g is",g)
```

12.2) Compile your C (python) program and run it. Make sure that your results for f and g are displayed. Print copy of your screen and turn it in your homework showing final results for f and g.

Summation of f is 60670

Summation of g is 4488050



12.3) Figure out how to turn in timer on your Raspberry Pi and measure how long it takes to calculate the entire for-loop. Turn in your timing analysis in your homework including code and timing results.

The test case as given above only covers 20 loops, so I made it thousands of times to get the exact execution time. I have tested for 3 cases.

First Case is 20, second is 1000, third is 5000 iterations.

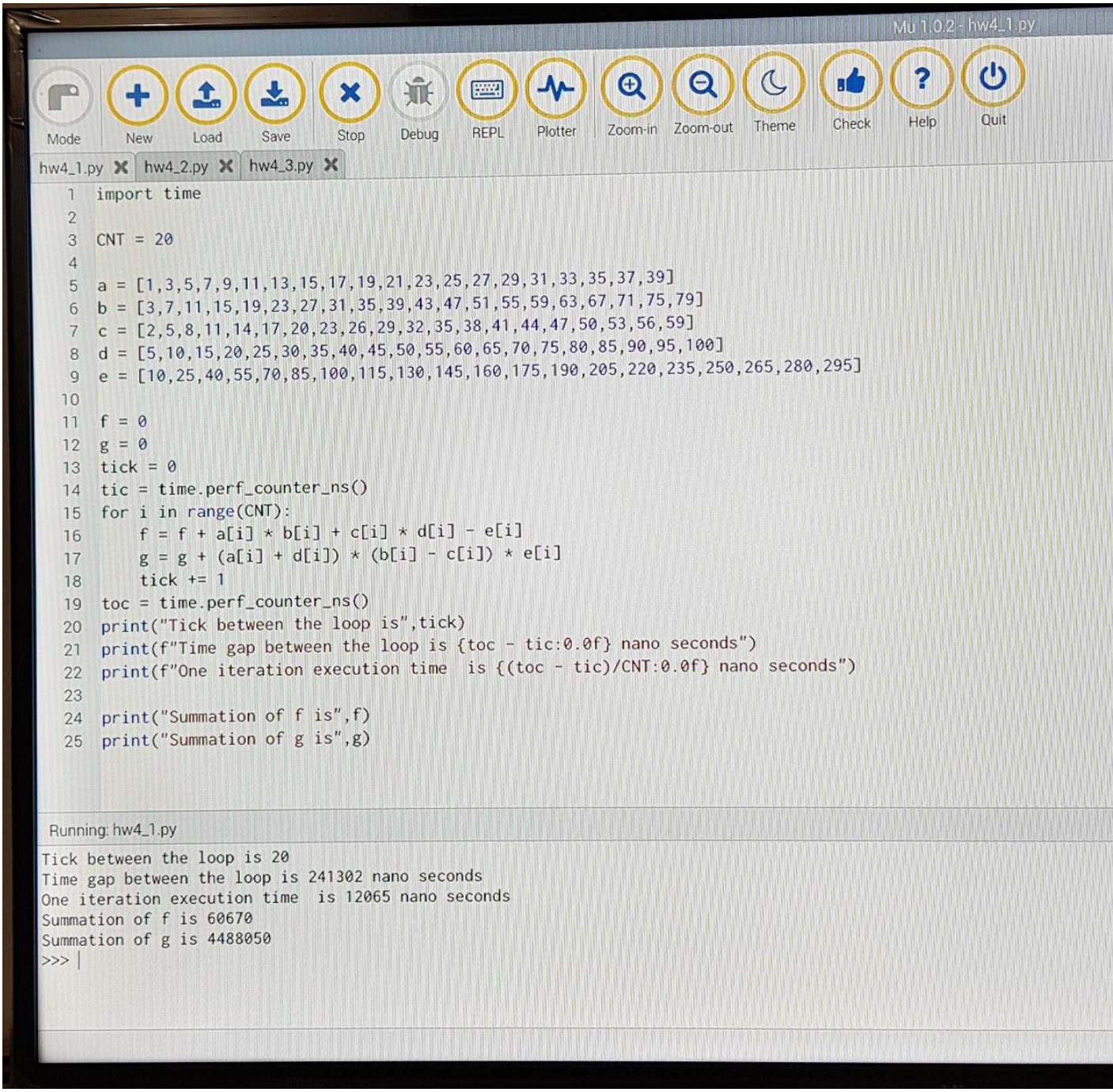
You can see that the first case requires more execution time for one iteration because the overhead of the loop has more impact when the loop only executes 20 times. However, in the third case, you can see that the time is shortened because the overhead of the loop does not significantly affect the total execution time.

As a result, when investigating execution time, we need to test a certain number of times that is not affected by the overhead.

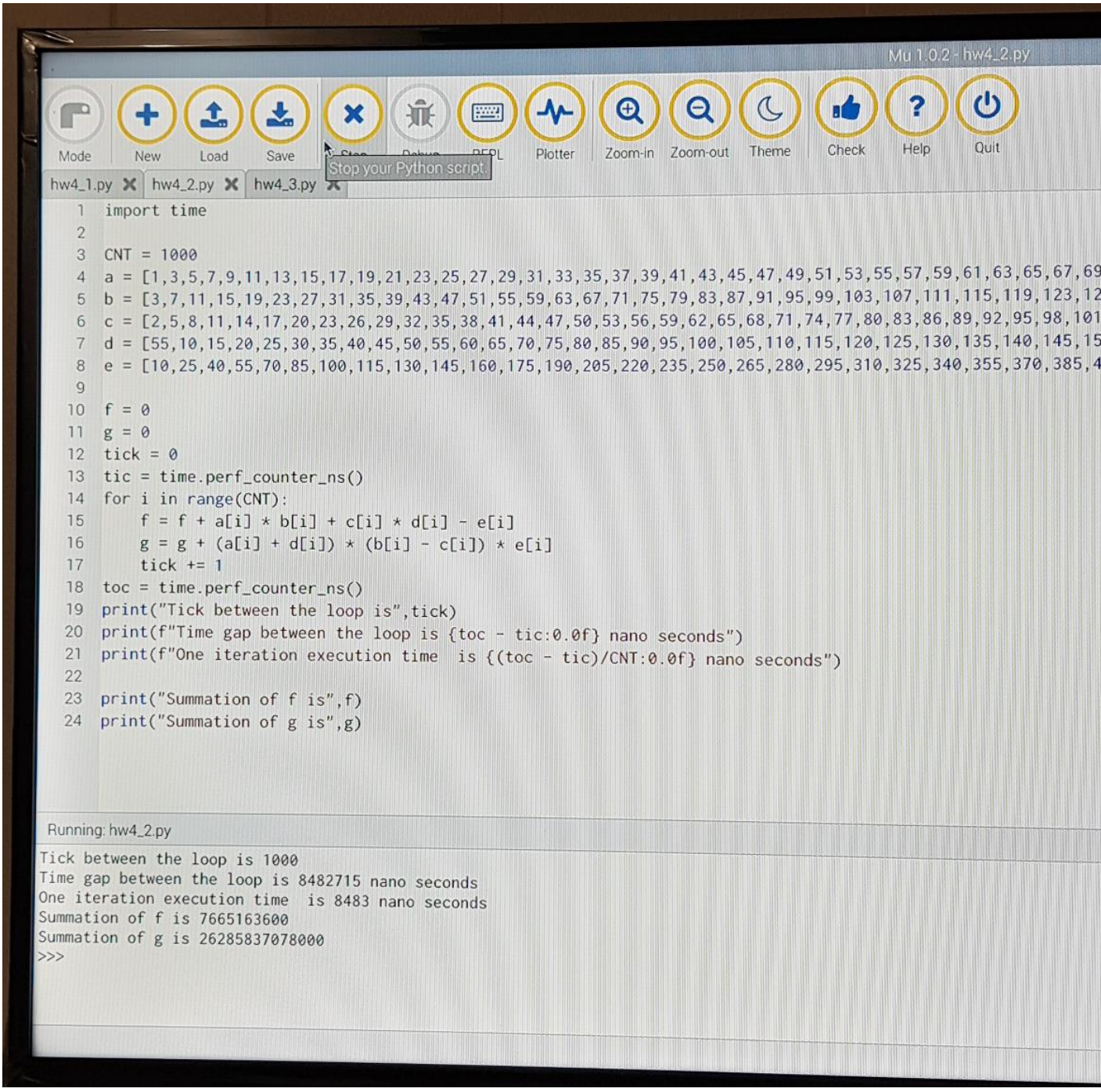
CNT	Total execution time (nano second)	Simulation time for one iteration(nano second)
20	241302	12065
1000	8482715	8483
5000	38480655	7696

Please see the screenshot below.

Byeongchan Gwak, Student ID : 501026



Byeongchan Gwak, Student ID : 501026



Byeongchan Gwak, Student ID : 501026