

**22AIE113**  
**ECS-2**  
**ASSIGNMENT-1**

**Guruprasath M R**  
**CH.EN.U4AIE22015**

1. Write and execute a hack assembly program for the following C statement  
 $k = (a+b) - (c+d)$

CODE:

```
// this code is to evaluate k = (a+b)-(c+d)

//input
//@0 == a
//@1 == b
//@2 == c
//@3 == d

//step1
//@4 == sum1 = a+b
//@5 == sum2 = c+d

//step1
//@6 == k = sum1 - sum2

@0
D = M
@1
D = D + M
@4
M = D

@2
D = M
@3
D = D + M
@5
M = D

@4
D = M
@5
D = D - M
@6
M = D
```

# Screen Shot

## INPUT

CPU Emulator (2.5) - D:\Sem2\Notes\ecs-2\Lab\assignment\_1.asm

File View Run Help

Slow Fast Animate: Program flow View: Screen Format: Decimal

**ROM** Asm

0	@0
1	D=M
2	@1
3	D=D+M
4	@4
5	M=D
6	@2
7	D=M
8	@3
9	D=D+M
10	@5
11	M=D
12	@4
13	D=M
14	@5
15	D=D-M
16	@6
17	M=D
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	

**RAM**

0	0
1	5
2	4
3	1
4	2
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0

PC: 0 A: 0

D: 0

ALU

D Input: 0 M/A Input: 0 ALU output: 0

## OUTPUT

CPU Emulator (2.5) - D:\Sem2\Notes\ecs-2\Lab\assignment\_1.asm

File View Run Help

Slow Fast Animate: Program flow View: Screen Format: Decimal

**ROM** Asm

0	@0
1	D=M
2	@1
3	D=D+M
4	@4
5	M=D
6	@2
7	D=M
8	@3
9	D=D+M
10	@5
11	M=D
12	@4
13	D=M
14	@5
15	D=D-M
16	@6
17	M=D
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	

**RAM**

0	5
1	5
2	3
3	3
4	10
5	6
6	4
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0

PC: 21 A: 6

D: 4

ALU

D Input: 4 M/A Input: 6 ALU output: 4

2. Write and execute a hack assembly program to swap two values.

Code:

```
// this code will swap 2 variable

//input
//@1 == var1
//@2 == var2

//extra
//@3 == tmp

//steps
// tmp = va1
// var1 = var2
// var2 = tmp
// tmp = 0
//1
@1
D = M
@3
M = D

//2
@2
D = M
@1
M = D

//3
@3
D = M
@2
M = D

//4
@3
M = 0

@END
0;JMP

(END)
@END
0;JMP
```

# Screen Shot

## INPUT

CPU Emulator (2.5) - D:\Sem2\Notes\ecs-2\Lab\assignment\\_2.asm

File View Run Help

Animate: Program flow View: Screen Format: Decimal

ROM Asm

Address	Instruction
0	@1
1	D=M
2	@3
3	M=D
4	@2
5	D=M
6	@1
7	M=D
8	@3
9	D=M
10	@2
11	M=D
12	@3
13	M=0
14	@16
15	0; JMP
16	@16
17	0; JMP
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	

PC: 0

RAM

Address	Value
0	0
1	5
2	10
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0

A: 0

D: 0

ALU

D Input: 0

M/A Input: 0

ALU output: 0

## OUTPUT

CPU Emulator (2.5) - D:\Sem2\Notes\ecs-2\Lab\assignment\\_2.asm

File View Run Help

Animate: Program flow View: Screen Format: Decimal

ROM Asm

Address	Instruction
0	@1
1	D=M
2	@3
3	M=D
4	@2
5	D=M
6	@1
7	M=D
8	@3
9	D=M
10	@2
11	M=D
12	@3
13	M=0
14	@16
15	0; JMP
16	@16
17	0; JMP
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	

PC: 16

RAM

Address	Value
0	0
1	10
2	5
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0

A: 16

D: 5

ALU

D Input: 5

M/A Input: 16

ALU output: 0

3. Write a hack assembly program to perform the sum of “n” numbers.

code

```
// this code will sum of n
```

```
//input
```

```
// @1 = n
```

```
//output
```

```
// @2 = sum
```

```
//steps
```

```
// do
```

```
// sum = sum + n
```

```
// n = n - 1
```

```
// while(n>0)
```

```
(LOOP)
```

```
    // sum = sum + n
```

```
    @1
```

```
    D = M
```

```
    @2
```

```
    M = M + D
```

```
    // n = n - 1
```

```
    @1
```

```
    M = M - 1
```

```
    @1
```

```
    D = M
```

```
    @LOOP
```

```
    D;JGT
```

```
    @END
```

```
    0;JMP
```

```
(END)
```

```
    @END
```

```
    0;JMP
```

# Screen Shot

## INPUT

CPU Emulator (2.5) - D:\Sem2\Notes\ecs-2\Lab\assignment\_3.asm

File View Run Help

Animate: Program flow View: Screen Format: Decimal

ROM Asm

Address	Instruction
0	@1
1	D=M
2	@2
3	M=D+M
4	@1
5	M=M-1
6	@1
7	D=M
8	@0
9	D;JGT
10	@12
11	0;JMP
12	@12
13	0;JMP
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	

RAM

Address	Value
0	0
1	10
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0

PC: 0 A: 0

D: 0

ALU

D Input: 0 M/A Input: 0 ALU output: 0

## OUTPUT

CPU Emulator (2.5) - D:\Sem2\Notes\ecs-2\Lab\assignment\_3.asm

File View Run Help

Animate: Program flow View: Screen Format: Decimal

ROM Asm

Address	Instruction
0	@1
1	D=M
2	@2
3	M=D+M
4	@1
5	M=M-1
6	@1
7	D=M
8	@0
9	D;JGT
10	@12
11	0;JMP
12	@12
13	0;JMP
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	

RAM

Address	Value
0	0
1	0
2	55
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0

PC: 13 A: 12

D: 0

ALU

D Input: 0 M/A Input: 12 ALU output: 0

4. Write a hack assembly program for the following C statement.

```
if (a>=b)
    return a;
else
    return b;
```

Code

```
// implement if-else
// check cmp a and b and return largest

//input
// @1 == a
// @2 == b

//output
// @5 == result

//code
// if(a>b):
//   return a
// else:
//   return b

// @5 = i
// i = a-b
// i;JGT
// inside if
// @5 == @1
// else:
// @5 == @2

@1
D = M
@2
D = D - M
@3
M = D
@3
D = M

@IF
D;JGT
@ELSE
0;JMP
// if i>=0
(IF)
    @1
    D = M
    @5
    M = D
@END
0;JMP
```



```

// else
(ELSE)
    @2
    D = M
    @5
    M = D
    @END
    0; JMP

(END)
    @3
    M = 0
    @END
    0; JMP

```

# Screen Shot

## INPUT

CPU Emulator (2.5) - D:\Sem2\Notes\ecs-2\Lab\assignment\\_3.asm

File View Run Help

Animate: Program flow View: Screen Format: Decimal

Slow Fast

ROM	Asm
0	@1
1	D=M
2	@2
3	M=D+M
4	@1
5	M=M-1
6	@1
7	D=M
8	@0
9	D; JGT
10	@12
11	0; JMP
12	@12
13	0; JMP
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	

RAM	
0	0
1	5
2	4
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0

PC 0 A 0

D 0

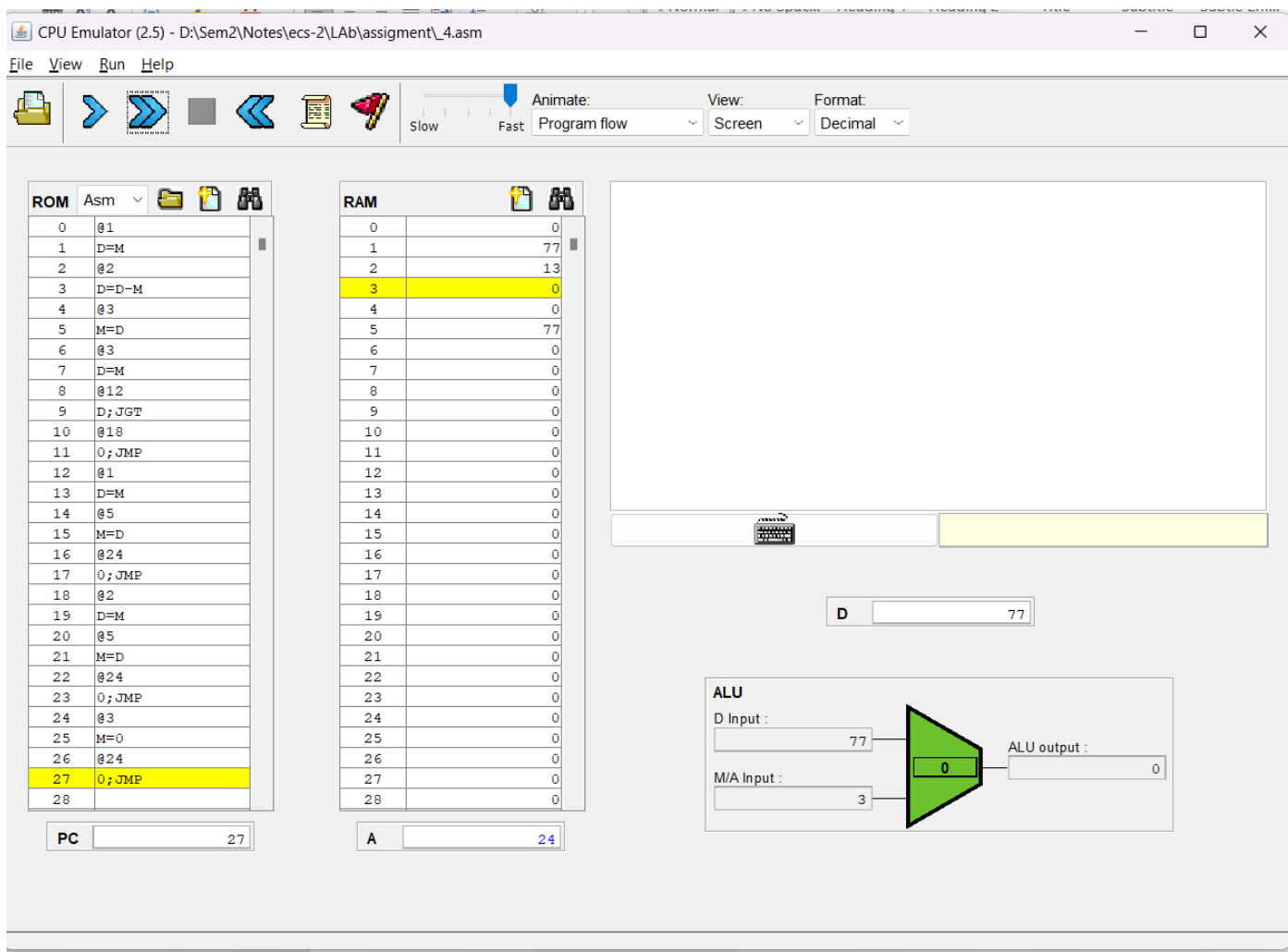
ALU

D Input : 0

M/A Input : 0

ALU output : 0

## OUTPUT



5. Write and execute a hack assembly program to perform multiplication of two operands.

## CODE

```
// this one to multiplication of a*b

// input
// @1 == a
// @2 == b

// output
// @3 == sum = a*b

// do
// @3 = @3 + a
// @2 = @2 - 1
// @2
// D = M
// @LOOP
// D;JEQ
// @END
// 0;JMP

(LOOP)
    @1
    D = M
    @3
    M = M + D
    @2
    M = M - 1
    @2
    D = M
    @END
    D;JEQ
    @LOOP
    0;JMP
(END)
    @END
    0;JMP
```

# Screen Shot

## INPUT

CPU Emulator (2.5) - D:\Sem2\Notes\ecs-2\Lab\assignment\_5.asm

File View Run Help

Slow Fast Animate: Program flow View: Screen Format: Decimal

ROM Asm

0	@1
1	D=M
2	@3
3	M=D+M
4	@2
5	M=M-1
6	@2
7	D=M
8	@12
9	D;JNEQ
10	@0
11	0;JMP
12	@12
13	0;JMP
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	

PC 0

RAM

0	0
1	5
2	4
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0

A 0

D 0

ALU

D Input : 0

M/A Input : 0

ALU output : 0

## OUTPUT

CPU Emulator (2.5) - D:\Sem2\Notes\ecs-2\Lab\assignment\_5.asm

File View Run Help

Slow Fast Animate: Program flow View: Screen Format: Decimal

ROM Asm

0	@1
1	D=M
2	@3
3	M=D+M
4	@2
5	M=M-1
6	@2
7	D=M
8	@12
9	D;JNEQ
10	@0
11	0;JMP
12	@12
13	0;JMP
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	

PC 13

RAM

0	0
1	5
2	0
3	20
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0

A 12

D 0

ALU

D Input : 0

M/A Input : 12

ALU output : 0

6. Write and execute a hack assembly program to perform division of two operands

CODE

```
// this one to division of a/b
```

```
// input
```

```
// @1 ==> a
```

```
// @2 ==> b
```

```
// output
```

```
// @3 == c = a/b
```

```
//
```

```
//steps
```

```
//while(a-b>0)
```

```
//c++;
```

```
//a = a - b
```

```
(LOOP)
```

```
@3
```

```
M = M + 1
```

```
@2
```

```
D = M
```

```
@1
```

```
M = M - D
```

```
@1
```

```
D = M
```

```
@LOOP
```

```
D; JGT
```

```
@END
```

```
0; JMP
```

```
(END)
```

```
@END
```

```
0; JMP
```

# Screen Shot

## INPUT

CPU Emulator (2.5) - D:\Sem2\Notes\ecs-2\Lab\assignment\_6.asm

File View Run Help

slow Fast Animate: Program flow View: Screen Format: Decimal

ROM Asm

0	@3
1	M=M+1
2	@2
3	D=M
4	@1
5	M=M-D
6	@1
7	D=M
8	@0
9	D; JGT
10	@12
11	0; JMP
12	@12
13	0; JMP
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	

PC 0

RAM

0	0
1	8
2	2
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0

A 0

D 0

ALU

D Input : 0

M/A Input : 0

ALU output : 0

## OUTPUT

CPU Emulator (2.5) - D:\Sem2\Notes\ecs-2\Lab\assignment\_6.asm

File View Run Help

slow Fast Animate: Program flow View: Screen Format: Decimal

ROM Asm

0	@3
1	M=M+1
2	@2
3	D=M
4	@1
5	M=M-D
6	@1
7	D=M
8	@0
9	D; JGT
10	@12
11	0; JMP
12	@12
13	0; JMP
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	

PC 13

RAM

0	0
1	0
2	2
3	4
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0

A 12

D 0

ALU

D Input : 0

M/A Input : 12

ALU output : 0