

# Computation

1	2	3	4	5
6	7	8	9	0
3	2	1	2	4
3	7	1	9	2

Input: 4x5

1	1	0
0	3	1
2	1	1

Kernel: 3x3

# Computation

1	2	3	4	5
6	7	8	9	0
3	2	1	2	4
3	7	1	9	2

input

1	1	0
0	3	1
2	1	1

kernel

# Computation

output

41		

1	2	3	4	5
6	7	8	9	0
3	2	1	2	4
3	7	1	9	2

input

1	1	0
0	3	1
2	1	1

kernel

$$1*1 + 2*1 + 3*0 + 6*0 + 7*3 + 8*1 + 3*2 + 2*1 + 1*1 = 41$$

# Computation

output

41		

1	2	3	4	5
6	7	8	9	0
3	2	1	2	4
3	7	1	9	2

input

1	1	0
0	3	1
2	1	1

kernel

# Computation

output

41	45	

1	2	3	4	5
6	7	8	9	0
3	2	1	2	4
3	7	1	9	2

input

1	1	0
0	3	1
2	1	1

kernel

$$2*1 + 3*1 + 4*0 + 7*0 + 8*3 + 9*1 + 2*2 + 1*1 + 2*1 = 45$$

# Computation

output

41	45	

1	2	3	4	5
6	7	8	9	0
3	2	1	2	4
3	7	1	9	2

input

1	1	0
0	3	1
2	1	1

kernel

# Computation

output

41	45	36

1	2	3	4	5
6	7	8	9	0
3	2	1	2	4
3	7	1	9	2

input

1	1	0
0	3	1
2	1	1

kernel

$$3*1 + 4*1 + 5*0 + 8*0 + 9*3 + 0*1 + 1*2 + 2*1 + 4*1 = 36$$

# Computation

output

41	45	36

1	2	3	4	5
6	7	8	9	0
3	2	1	2	4
3	7	1	9	2

input

1	1	0
0	3	1
2	1	1

kernel



# Computation

output

41	45	36
34		

1	2	3	4	5
6	7	8	9	0
3	2	1	2	4
3	7	1	9	2

input

1	1	0
0	3	1
2	1	1

kernel

$$6*1 + 7*1 + 8*0 + 3*0 + 2*3 + 1*1 + 3*2 + 7*1 + 1*1 = 34$$

# Computation (Final)

output

41	45	36
34	44	40

1	2	3	4	5
6	7	8	9	0
3	2	1	2	4
3	7	1	9	2

input

1	1	0
0	3	1
2	1	1

kernel

# Pseudo Code

input: a[4][5],    kernel: b[3][3],    output: c[2][3]

```
for (n=0; n<(4-3+1); n++) {
    for (m=0; m<(5-3+1); m++) {

        //-----
        for (i=n; i<(n+3); i++) {
            for (j=m; j<(m+3); j++) {
                tmp = tmp + a[i][j]*b[i-n][j-m];
            }
        }
        //-----

        c[n][m] = tmp;
    }
}
```

## Homework #4 (1)

Write an ARM assembly program to do the computation.

- input is a 4x5 matrix and kernel is a 3x3 matrix
- output is a 2x3 matrix
- Figure 1 shows the layout of the output matrix
- Each element in input, kernel, and output is a word-sized integer
- The integer values of input and kernel are assigned by yourself

# Layout of output matrix

Register r1



(1,1)
(1,2)
(1,3)
(2,1)
(2,2)
(2,3)

## Homework #4 (2)

- The overflow/underflow problems are not considered during the computation
- After computation, register r1 will point to the address of output's first element
- 請勿繳交【利用編譯器所自動產生的組合語言程式】
- 請勿抄襲
- **Deadline: November 27 (Wednesday), 24:00, 2019.**