Computer Science

COMPSCI 210 S1

C Programming Assignment

Due:	**11:59 pm Tuesday 6 June 2023**
Worth:	**6 marks (6% of the final mark)**
Late Submission	**30% penalty**

Introduction

1. Convolution

Convolution is the most fundamental concept in signal processing and analysis. By using convolution, we can construct the output of the system for any arbitrary input signal, if we know th[illegible]

2. Convolution in 2[illegible]

2D convolution is j[illegible] convolving both horizontal and vert[illegible]. Convolution is frequently used fo[illegible]ning, and edge detection of images[illegible]

The impulse (delta) [illegible]e in 2D is usually called "kernel" or "fi[illegible]

The second image i[illegible]tion. The shaded centre point is the o[illegible]decomposed into a sum of scaled an[illegible]

x [illegible]

Notice that the ker[illegible] in most cases, which means the centre point of a kernel is h[0, 0]. For example, if the kernel size is 5, then the array index of 5 elements will be -2, -1, 0, 1, and 2. The origin is located in the middle of the kernel.

n \ m	-1	0	1
-1	a	b	c
0	d	e	f
1	g	h	i

Examine an example to clarify how to convolve in 2D space. Let's say that the size of the impulse response (kernel) is 3x3, and its values are a, b, c, d,..., i. Notice the origin (0,0) is located in the centre of the kernel. Let's pick the simplest sample and compute convolution, for instance, the output at (1, 1) will be:

$$
\begin{aligned}
y[1,1] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i,j] \cdot h[1-i, 1-j] \\
= \quad & x[0,0] \cdot h[1,1] \quad + x[1,0] \cdot h[0,1] \quad + x[2,0] \cdot h[-1,1] \\
& + x[0,1] \cdot h[1,0] \quad + x[1,1] \cdot h[0,0] \quad + x[2,1] \cdot h[-1,0] \\
& + x[0,2] \cdot h[1,-1] + x[1,2] \cdot h[0,-1] + x[2,2] \cdot h[-1,-1]
\end{aligned}
$$

It results in a sum of 9 elements of scaled and shifted impulse responses. The following image shows the graphical representation of 2D convolution.

Notice that the kern directions before
multiplying the over the last sample
of the impulse resp mple, h[-1,-1].

Exercise a little mo uppose we have
3x3 input and 3x3 k

1	2	3
4	5	6
7	8	9

Input

-1	-2	-1
0	0	0
1	2	1

Kernel

-13	-20	-17
-18	-24	-18
13	20	17

Output

The output at (1, 1) for this example will be:

$$
\begin{aligned}
y[1,1] = \sum_j \sum_i x[i,j] \cdot h[1-i,1-j] \\
= \quad & x[0,0]\cdot h[1,1] \quad + x[1,0]\cdot h[0,1] \quad + x[2,0]\cdot h[-1,1] \\
& + x[0,1]\cdot h[1,0] \quad + x[1,1]\cdot h[0,0] \quad + x[2,1]\cdot h[-1,0] \\
& + x[0,2]\cdot h[1,-1] + x[1,2]\cdot h[0,-1] + x[2,2]\cdot h[-1,-1] \\
= \quad & 1\cdot 1 + 2\cdot 2 + 3\cdot 1 \\
& + 4\cdot 0 + 5\cdot 0 + 6\cdot 0 \\
& + 7\cdot(-1) + 8\cdot(-2) + 9\cdot(-1) \\
= & -24
\end{aligned}
$$

C Programming

In this assignment, you are asked to write a C program to implement the 2D convolution. In this … ad from the files based on the comm… calculation, the results matrix (1024… d line argument.

1. Command line a…

The command line… s in the correct sequence:

Executable (convol… t file” “number of convolutions”

Example: ./convolut…

- “./convolutio…
- “./data1.txt” i…
- “./filter1.txt” i…
- “temp2” is th…
- “12” is the nu…

Timing the execution time of the programme:

Example: time ./convolution2 ./data1.txt ./filter1.txt temp2 12

2. Two data structures for the data and filter

You are asked to use two different data structures to implement the convolution. For the first one (named as convolution1.c), you should use “struct” to store the data (o_val) and the output (n_val) matrices such as below:

```
struct matrix {
    int o_val;
    int n_val;
};
typedef struct matrix Matrix;

int main(int argc, char *argv[]) {
    FILE *file1, *file2, *file3;
    int i = 0;
    int filter[5][5];
    Matrix** data;
    int j, k, l, m;
    int val;
    int iter;

    data = (
    for (i =
        data
    }
    file1 =
    file2 =
    file3 =
    iter = a
```

In the second one parate arrays to store the data (data

```
int main(int
    FILE *fi
    int i =
    int filt
    int** da
    int** rl
    int j, k
    int val;
    int iter;

    data = (int**) malloc(sizeof(int*)*1024);
    rlt = (int**) malloc(sizeof(int*)*1024);
    for (i = 0; i < 1024; i++) {
        data[i] = (int*) malloc(sizeof(int)*1024);
        rlt[i] = (int*) malloc(sizeof(int)*1024);
    }
    file1 = fopen(argv[1], "r");
    file2 = fopen(argv[2], "r");
    file3 = fopen(argv[3], "w");
    iter = atoi(argv[4]);
```

3. Implementation details

In the implementation, you need to do saturation and scaling in addition to convolution. You can follow the steps below.

1. $y[p,q] = \sum_j \sum_i x[p+i, q+j] \times h[0-i, 0-j]$
2. $y'[p,q] = \frac{y[p,q]}{16}$
3. $y''[p,q] = \begin{cases} 16, if\ y'[p,q] > 16 \\ y'[p,q], if -16 \le y'[p,q] \le 16 \\ -16, if\ y'[p,q] < -16 \end{cases}$

In the first step, you can do the convolution between the data matrix (x[p, q]) and the filter matrix (h[i,j]). The results of each convolution should be stored in the output matrix (y[p, q]). This is used as the data matrix in the next convolution. Notice that the size of the data matrix is not the same as the filter matrix. The array index (i and j) of 5 elements in the above equations will be -2, -1, 0, 1, and 2. You can see an example in the appendix for the details of convolution.

In the second step, t values becoming too large or too sm eep the range of values between -16

4. Report Writing

Write a report (maxi ose two different data structures. Yo uch as merging array and loop inte influenced the cache operations re

5. Submission

You may electror Web Dropbox (https://adb.aucklar up until the final date. You can make mission that you make replaces your ery submission. Only your very lates ck that you have included all the files

No marks will be awarded if your program does not compile and run. You are to electronically submit all the following files:

- convolution1.c
- convolution2.c
- report.pdf

6. Grading

- Report (3 marks)
  - Clear description about the memory management on convolution1.c (1 mark)
  - Clear description about the memory management on convolution2.c (1 mark)
  - Clear analysis about the memory management between convolution1.c and convolution2.c (1 mark)
  - You are suggested to use more convolutions to show the time difference. Example: “time ./convolution2 ./data1.txt ./filter1.txt temp 100”
- Programme correctness (3 marks)
  - 20 test cases will be tested. 12 of the results files will be available on Canvas.
  - ./convolution1 ./data1.txt ./filter1.txt temp111 1
  - ./convolution1 ./data1.txt ./filter2.txt temp121 1
  - ./convolution1 ./data1.txt ./filter3.txt temp131 1
  - ./conv
  - ./conv
  - ./conv
  - ./conv
  - ./conv
  - ./conv
  - ./conv
  - ./conv
  - ./conv
  - ./conv
  - ./conv
  - ./conv
  - ./conv
  - ./conv
  - ./conv
  - ./conv
  - ./conv

Appendix

1	2	3
4	5	6
7	8	9

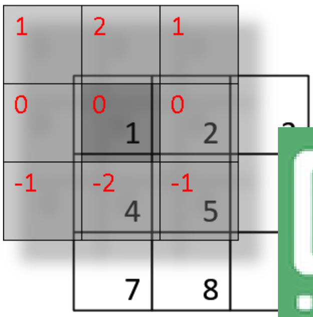
Input

-1	-2	-1
0	0	0
1	2	1

Kernel

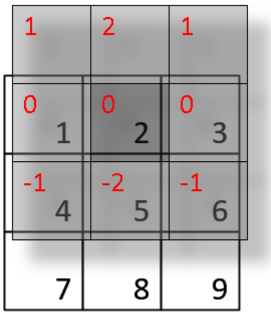
-13	-20	-17
-18	-24	-18
13	20	17

Output



$$y[0,0] = \sum_j \sum_i x[i,j] \cdot h[0-i,0-j]$$

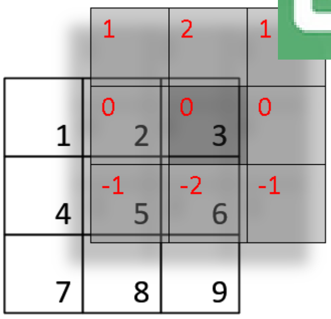
$$\begin{aligned} = \quad & x[-1,-1] \cdot h[1,1] + x[0,-1] \cdot h[0,1] + x[1,-1] \cdot h[-1,1] \\ & + x[-1,0] \cdot h[1,0] \quad + x[0,0] \cdot h[0,0] \quad + x[1,0] \cdot h[-1,0] \\ &] \cdot h[-1,-1] \end{aligned}$$



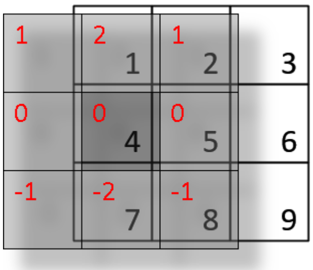
$$\cdot\, h[-1,1]$$

$$h[-1,0]$$

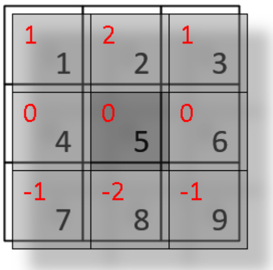
$$h[-1,-1]$$



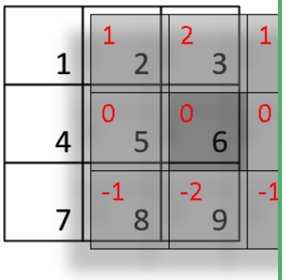
$$\begin{aligned} & 1] \cdot h[-1,1] \\ & + x[1,0] \cdot h[1,0] \quad + x[2,0] \cdot h[0,0] \quad + x[3,0] \cdot h[-1,0] \\ & + x[1,1] \cdot h[1,-1] + x[2,1] \cdot h[0,-1] + x[3,1] \cdot h[-1,-1] \\ = \quad & 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 1 \\ & + 2 \cdot 0 + 3 \cdot 0 + 0 \cdot 0 \\ & + 5 \cdot (-1) + 6 \cdot (-2) + 0 \cdot (-1) \\ = & -17 \end{aligned}$$



$$
\begin{aligned}
y[0,1] &= \sum_j \sum_i x[i,j] \cdot h[0-i,1-j] \\
&= \quad x[-1,0] \cdot h[1,1] \quad + x[0,0] \cdot h[0,1] \quad + x[1,0] \cdot h[-1,1] \\
&\quad + x[-1,1] \cdot h[1,0] \quad + x[0,1] \cdot h[0,0] \quad + x[1,1] \cdot h[-1,0] \\
&\quad + x[-1,2] \cdot h[1,-1] + x[0,2] \cdot h[0,-1] + x[1,2] \cdot h[-1,-1] \\
&= \quad 0 \cdot 1 + 1 \cdot 2 + 2 \cdot 1 \\
&\quad + 0 \cdot 0 + 4 \cdot 0 + 5 \cdot 0 \\
&\quad + 0 \cdot (-1) + 7 \cdot (-2) + 8 \cdot (-1) \\
&= -18
\end{aligned}
$$



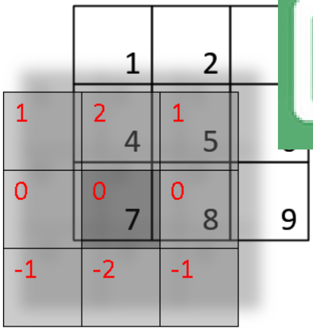
$$
\begin{aligned}
y[1,1] &= \sum_j \sum_i x[i,j] \cdot h[1-i,1-j] \\
&= \quad x[0,0] \cdot h[1,1] \quad + x[1,0] \cdot h[0,1] \quad + x[2,0] \cdot h[-1,1] \\
&\quad + x[0,1] \cdot h[1,0] \quad + x[1,1] \cdot h[0,0] \quad + x[2,1] \cdot h[-1,0] \\
&\quad + x[0,2] \cdot h[1,-1] + x[1,2] \cdot h[0,-1] + x[2,2] \cdot h[-1,-1] \\
&= \quad 1 \cdot 1 + 2 \cdot 2 + 3 \cdot 1
\end{aligned}
$$



$0] \cdot h[-1,1]$

$1] \cdot h[-1,0]$

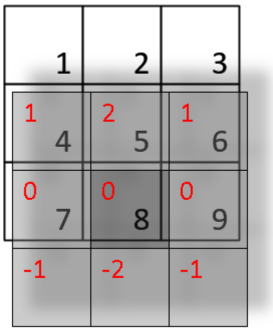
$2] \cdot h[-1,-1]$



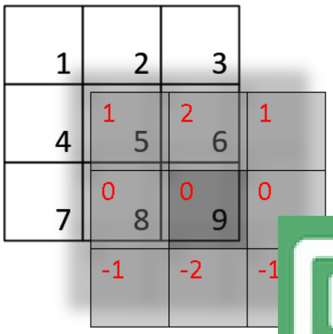
$] \cdot h[-1,1]$

$] \cdot h[-1,0]$

$$
\begin{aligned}
&+ x[-1,3] \cdot h[1,-1] + x[0,3] \cdot h[0,-1] + x[1,3] \cdot h[-1,-1] \\
&= \quad 0 \cdot 1 + 4 \cdot 2 + 5 \cdot 1 \\
&\quad + 0 \cdot 0 + 7 \cdot 0 + 8 \cdot 0 \\
&\quad + 0 \cdot (-1) + 0 \cdot (-2) + 0 \cdot (-1) \\
&= 13
\end{aligned}
$$



$$\begin{aligned}
y[1,2] &= \sum_j \sum_i x[i,j] \cdot h[1-i, 2-j] \\
&= \quad x[0,1]\cdot h[1,1] \quad + x[1,1]\cdot h[0,1] \quad + x[2,1]\cdot h[-1,1] \\
&\quad + x[0,2]\cdot h[1,0] \quad + x[1,2]\cdot h[0,0] \quad + x[2,2]\cdot h[-1,0] \\
&\quad + x[0,3]\cdot h[1,-1] + x[1,3]\cdot h[0,-1] + x[2,3]\cdot h[-1,-1] \\
&= \quad 4\cdot 1 + 5\cdot 2 + 6\cdot 1 \\
&\quad + 7\cdot 0 + 8\cdot 0 + 9\cdot 0 \\
&\quad + 0\cdot(-1) + 0\cdot(-2) + 0\cdot(-1) \\
&= 20
\end{aligned}$$



$$\begin{aligned}
y[2,2] &= \sum_j \sum_i x[i,j] \cdot h[2-i, 2-j] \\
&= \quad x[1,1]\cdot h[1,1] \quad + x[2,1]\cdot h[0,1] \quad + x[3,1]\cdot h[-1,1] \\
&\quad + x[1,2]\cdot h[1,0] \quad + x[2,2]\cdot h[0,0] \quad + x[3,2]\cdot h[-1,0] \\
&\quad + x[1,3]\cdot h[1,-1] + x[2,3]\cdot h[0,-1] + x[3,3]\cdot h[-1,-1] \\
&= \quad 5\cdot 1 + 6\cdot 2 + 0\cdot 1
\end{aligned}$$