이번 과제를 하면서 중요하게 생각한 것 중 하나가 가독성이었다. 이러한 가독성을 위해서 함수들의 이름을 통해 무슨 작업을 하는 함수인지 직관적이게 알 수 있도록 노력했다.

구현하고자 하는 바를 함수로 잘게 나누어서 main()에 코드 줄 수를 최대한 줄이는 방법으로 구현했다. 즉 main()만 보더라도 어떤 일을 하는 코드인지 대략적으로 알 수 있는 코드이다.

|  |  |  |
| --- | --- | --- |
| Function Name | Arguments | Description |
| Void makeMatrix | Int\*\* matrix | 동적 할당을 받는 2차원 배열, 초기 상태 배열 |
| Int x | 2차원 배열에서의 행 |
| Int y | 2차원 배열에서의 열 |
| Return Value |  |

|  |  |  |
| --- | --- | --- |
| Function Name | Arguments | Description |
| Void setCM | Int\*\* CM | 동적 할당을 받는 2차원 배열, 처리 과정 한 번 거친 배열 |
| Int cx | 처리 과정 한 번 거친 배열의 행 |
| Int cy | 처리 과정 한 번 거친 배열의 열 |
| Return Value |  |

|  |  |  |
| --- | --- | --- |
| Function Name | Arguments | Description |
| Void CalCon | Int id | Convolutional 계산을 위한메시지 큐의 아이디 |
|  |  |
|  |  |
| Return Value |  |

|  |  |  |
| --- | --- | --- |
| Function Name | Arguments | Description |
| Void CalMax | Int id | Pooling 계산을 위한 메시지 큐의 아이디 |
|  |  |
|  |  |
| Return Value |  |

|  |  |  |
| --- | --- | --- |
| Function Name | Arguments | Description |
| Void devideIM | Int\*\* IM | makeMatrix로 만들어진 2차원 배열 |
| Int x | makeMatrix로 만들어진 2차원 배열의 행 |
| Int y | makeMatrix로 만들어진 2차원 배열의 열 |
| Return Value |  |

|  |  |  |
| --- | --- | --- |
| Function Name | Arguments | Description |
| Void devideCM | Int\*\* CM | setCM으로 만들어진 2차원 배열 |
| Int cx | setCM으로 만들어진 2차원 배열의 행 |
| Int cy | setCM으로 만들어진 2차원 배열의 열 |
| Return Value |  |

|  |  |  |
| --- | --- | --- |
| Function Name | Arguments | Description |
| Void makeCCP | Int x | makeMatrix로 만들어진 2차원 배열의 행의 수 |
| Int y | makeMatrix로 만들어진 2차원 배열의 열의 수 |
|  |  |
| Return Value |  |

|  |  |  |
| --- | --- | --- |
| Function Name | Arguments | Description |
| Void makeMCP | Int cx | setCM으로 만들어진 2차원 배열의 행의 수 |
| Int cy | setCM으로 만들어진 2차원 배열의 열의 수 |
|  |  |
| Return Value |  |

|  |  |  |
| --- | --- | --- |
| Function Name | Arguments | Description |
| Void printResult | Int cx | setCM으로 만들어진 2차원 배열의 행의 수 |
| Int cy | setCM으로 만들어진 2차원 배열의 열의 수 |
|  |  |
| Return Value |  |