이번 과제는 프로세스간 통신이 아닌 쓰레드간 통신이었기에 공유하는 자원을 넘겨주는 방식은 훨씬 편하게 구현할 수 있었다. 하지만 자원을 넘겨주더라도 여러가지 자원을 한 번에 넘기려고 하면 구조체를 이용했어야 했기에 적정한 크기로 구조체 배열을 동적 할당하는 것에 신경을 썼다.

함수는 총 4개를 사용했으며 main에서 콜하는 함수 2개 쓰레드를 만들고 쓰레드가 실행하는 함수 2개이다. 함수를 main에서 2번만 콜해서 이 프로그램이 어떤 흐름을 가지는지 대충 알 수 있게끔 코드를 짰다.

그렇게 잘 짠 코드는 아니지만 그래도 가독성은 좋게끔 최대한 노력하면서 짰다.

|  |  |  |
| --- | --- | --- |
| Function Name | Arguments | Description |
| Void Cal\_Con | Int\*\* IM | 동적 할당을 받는 2차원 배열, input.txt로부터 값을 받아와 만들어진 배열 |
| Int conNum | CM 배열의 길이 |
| Int \*\*CM | 동적 할당을 받는 2차원 배열, Cal\_Con의 실질적 return 값으로 보면 됨 |
| Return Value |  |

|  |  |  |
| --- | --- | --- |
| Function Name | Arguments | Description |
| Void \*make\_con\_thread | Void \*arg | 3\*3으로 나누어진 배열과 배열이 나누어진 인덱스 값 2개를 인자로서 받는다. |
|  |  |
|  |  |
| Return Value | con\_ret이라는 형태의 구조체의 형식으로 리턴한다. 구조체 con\_ret는 인덱스 값 2개와 배열을 통해 나온 결과값을 저장한다. |

|  |  |  |
| --- | --- | --- |
| Function Name | Arguments | Description |
| Void Cal\_Pool | Int \*\*CM | 동적 할당을 받는 2차원 배열, Cal\_Con으로부터 값을 계산해 만들어진 배열 |
| Int conNum | CM 배열의 길이 |
| Int \*\*PM | 동적 할당을 받는 2차원 배열, Cal\_Pool의 실질적 return 값으로 보면 됨 |
| Return Value |  |

|  |  |  |
| --- | --- | --- |
| Function Name | Arguments | Description |
| Void \*make\_pool\_thread | Void \*arg | 2\*2으로 나누어진 배열과 배열이 나누어진 인덱스 값 2개를 인자로서 받는다. |
|  |  |
|  |  |
| Return Value | pool\_ret이라는 형태의 구조체의 형식으로 리턴한다. 구조체 pool\_ret는 인덱스 값 2개와 배열을 통해 나온 결과값을 저장한다. |