

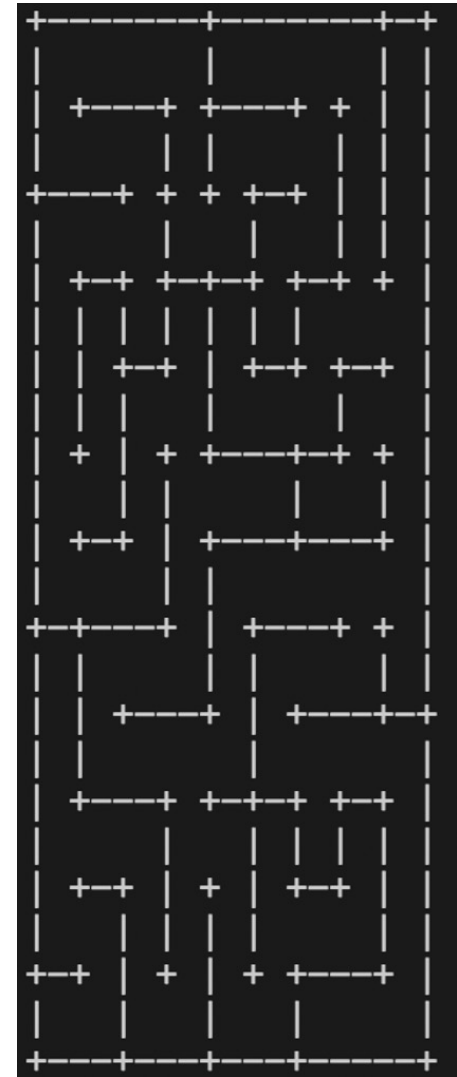
# 자료구조

과제 1. 미로 그리기

물리학과 20182326 이선민

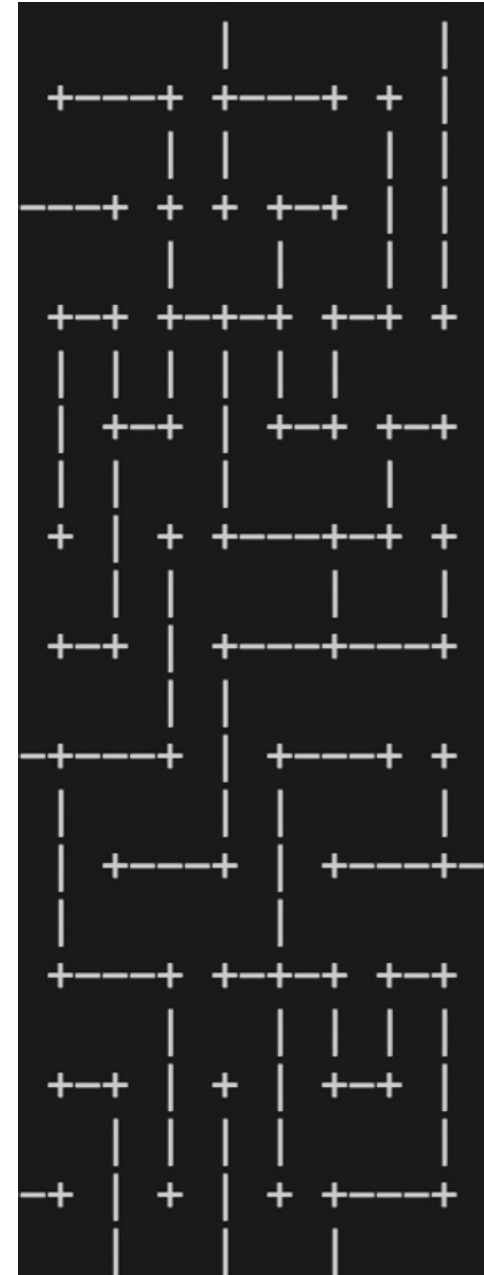
# 문제 내용

- 오른쪽 그림과 같은 미로를 텍스트 파일에서 읽어와 maze라는 이중배열에 저장한 후 그 이중배열을 읽어 다시 미로를 출력했다.



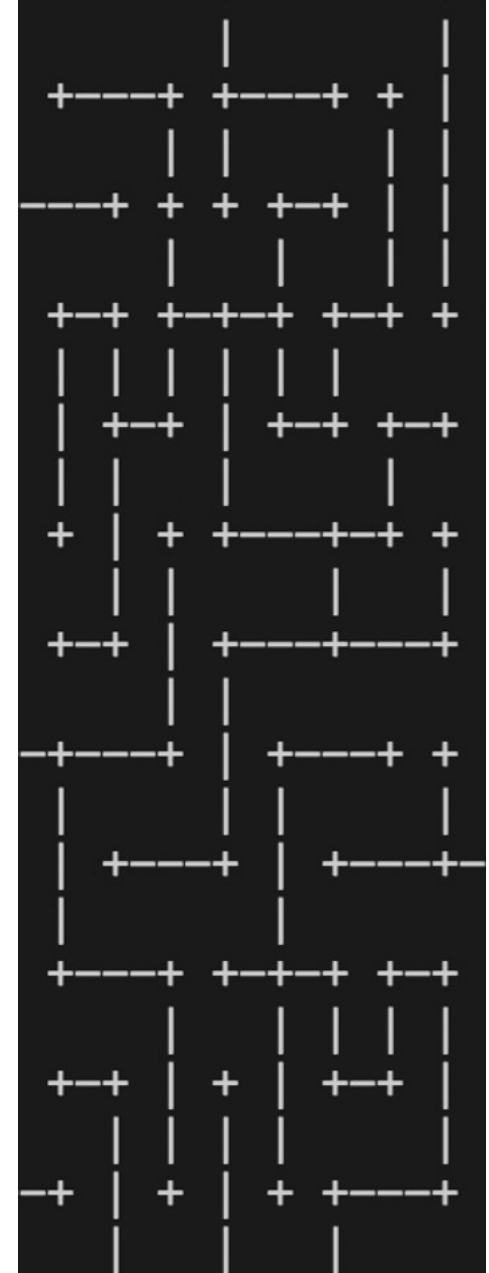
# 문제 해결 아이디어

- 우선 row에서 맨 윗줄과 맨 아랫줄은 두번째 줄과 밑에서 두번째 줄을 통해 유추해서 그릴 수 있으니 잘랐다.
- 왼쪽 끝 벽과 오른쪽 끝 벽도 옆에 있는 줄을 통해 유추하여 그릴 수 있으니 잘랐다.
- 오른쪽 그림처럼 저장되게 된다.
- 이 상태에서 각각의 row를 리스트에 각각 저장하고, row들을 저장한 리스트를 이중 리스트 maze에 최종적으로 담았다.



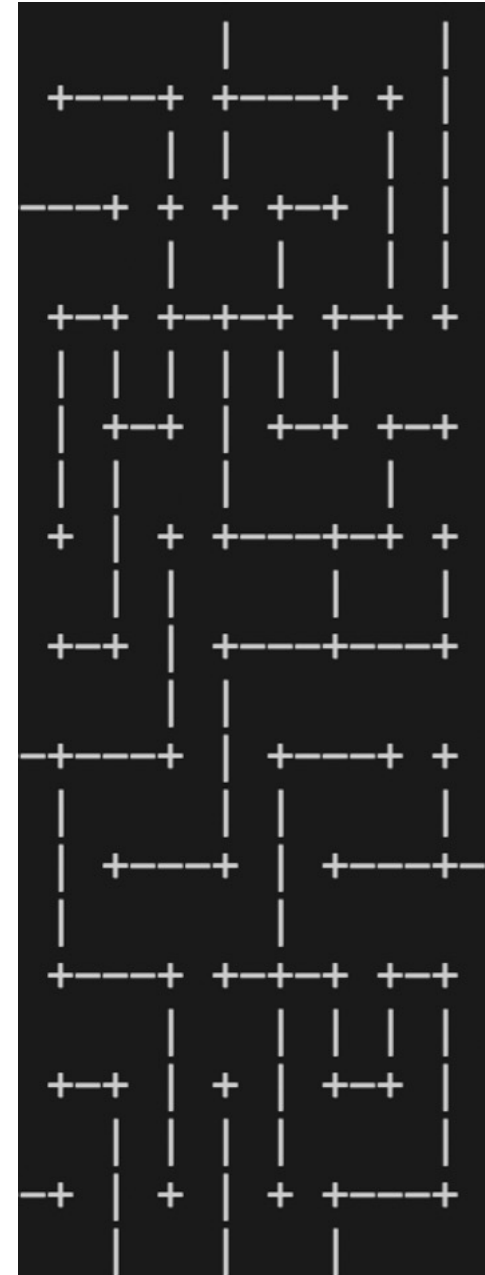
# 문제 해결 아이디어

- 각각의 row를 각각의 리스트에 저장할 때 짝수행과 홀수행의 규칙을 다르게 하여 저장했다.
- row중 맨 윗줄을 0번째 줄 이라고 할 때, 짝수 행들은 문자가 공백과 'I'밖에 없다. 따라서 짝수행은 'I'의 위치 인덱스를 숫자로 리스트에 저장했다.
- 홀수행들은 주로 '-' 문자가 반복되는 경우가 많아, '-' 문자가 두개 이상 반복되는 경우 반복되는 개수를 세어 숫자만 리스트에 담았다.
- 예를 들어 '+----+' 인 경우 리스트에는 ['+', '3', '+'] 이런식으로 담킨다.



# 문제 해결 아이디어

- 그리고 마지막으로 ‘+’ 와 ‘ ’(공백) 이 연속하여 나오는 경우 ‘+ ’ = ‘a’ 로 바꾸어줬고
- ‘+’ 와 ‘-’가 연속으로 나오는 경우 ‘+-’ = ‘b’ 로 바꾸어 줬고
- ‘|’ 와 ‘ ’(공백) 이 나오는 경우 ‘| ’ = ‘c’ 로 바꿔줬다.



# 이중배열에 저장한 내용

- 위의 과정을 거쳐  
저장하게 되면 최종  
다음과 같이  
저장되게 된다.
- 오른쪽의 내용들이  
Maze라는 이중  
배열에 저장된다.

바이트 수는 maze1  
일때, 총 143 바이트,  
maze2 일 때 총  
132바이트를 사용해  
저장했다.

```
[8, 16]
[' ', '+', '3', 'a', '+', '3', 'a', 'a', 'c']
[6, 8, 14, 16]
['3', 'a', 'a', 'a', 'b', 'a', 'c', 'c']
[6, 10, 14, 16]
[' ', 'b', 'a', 'b', 'b', 'a', 'b', 'a', 'a']
[2, 4, 6, 8, 10, 12]
[' ', 'c', 'b', 'a', 'c', 'b', 'a', 'b', 'a']
[2, 4, 8, 14]
[' ', 'a', 'c', 'a', '+', '3', 'b', 'a', 'a']
[4, 6, 12, 16]
[' ', 'b', 'a', 'c', '+', '3', '+', '3', 'a']
[6, 8]
['-', '+', '3', 'a', 'c', '+', '3', 'a', 'a']
[2, 8, 10, 16]
[' ', 'c', '+', '3', 'a', 'c', '+', '3', 'b']
[2, 10]
[' ', '+', '3', 'a', 'b', 'b', 'a', 'b', 'a']
[6, 10, 12, 14, 16]
[' ', 'b', 'a', 'c', 'a', 'c', 'b', 'a', 'c']
[4, 6, 8, 10, 16]
['-', 'a', 'c', 'a', 'c', 'a', '+', '3', 'a']
[4, 8, 12]
```

## 결과 캡처

```
[~/2022_1/Data_structure/Assignment/assignment_1]$  
9 12  
+-----+-----+  
|         |         |  
|  +---+  +---+  +  |  
|         |         |  
+---+  +  +  +---+  |  
|         |         |  
|  ++  ++---+  ++  +  
|  |  |  |  |  |  |  
|  ++  ++---+  ++  +  
|  |  |  |  |  |  |  
|  +  +  +---+  +  +  
|  |  |  |  |  |  |  
|  ++  ++---+  ++  +  
+---+  +  +---+  +  
|  |  |  |  |  |  |  
|  +---+  +---+  +  
|  |  |  |  |  |  |  
|  ++  +  +  ++  |  
|  |  |  |  |  |  |  
+---+  +  +  +---+  
|  |  |  |  |  |  |  
+-----+-----+  
+-----+-----+  
|         |         |  
|  +---+  +---+  +  |  
|         |         |  
+---+  +  +  +---+  |  
|         |         |  
|  ++  ++---+  ++  +  
|  |  |  |  |  |  |  
|  ++  ++---+  ++  +  
|  |  |  |  |  |  |  
|  +  +  +---+  +  +  
|  |  |  |  |  |  |  
|  ++  ++---+  ++  +  
+---+  +  +---+  +  
|  |  |  |  |  |  |  
|  +---+  +---+  +  
|  |  |  |  |  |  |  
|  ++  +  +  ++  |  
|  |  |  |  |  |  |  
+---+  +  +  +---+  
|  |  |  |  |  |  |  
+-----+-----+
```

## 결과 캡처

```
[~/2022_1/Data_structure/Assignment/assignment_1]$
```

10 10