

```
import java.util.*;

public class Ex {
    public static void main(String[] args) {
        // 주어진 배열 시계방향으로 90도 회전시켜서 출력하기
        char[][] star = {
            {'*', '*', ' ', ' ', ' '},
            {'*', '*', ' ', ' ', ' '},
            {'*', '*', '*', '*', '*'},
            {'*', '*', '*', '*', '*'}
        };

        char[][] result = new char[star[0].length][star.length];

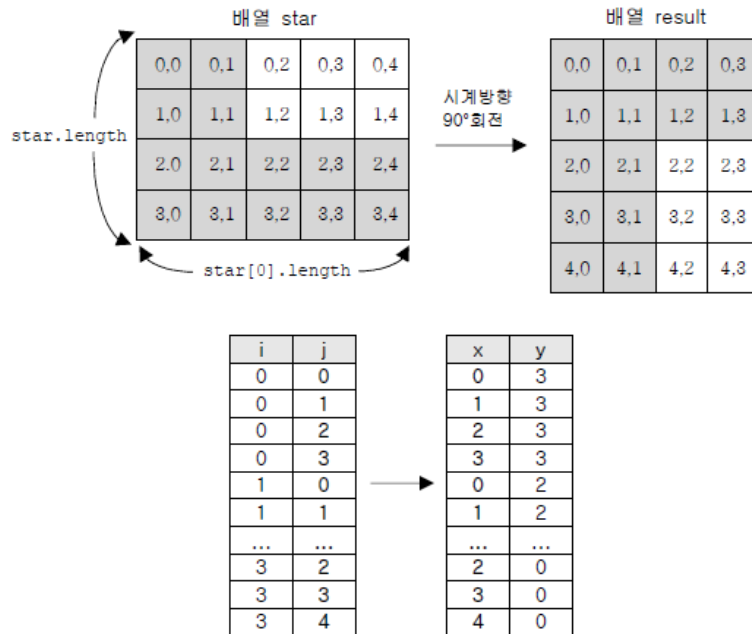
        for (int i = 0; i < star.length; i++) {
            for (int j = 0; j < star[i].length; j++) {
                System.out.print(star[i][j]);
            }
            System.out.println();
        }

        System.out.println();

        for (int i = 0; i < star.length; i++) {
            for (int j = 0; j < star[i].length; j++) {
                int x = j;
                int y = star.length - 1 - i;
                result[x][y] = star[i][j];
            }
        }

        for (int i = 0; i < result.length; i++) {
            for (int j = 0; j < result[i].length; j++) {
                System.out.print(result[i][j]);
            }

            System.out.println();
        }
    }
}
```



```
import java.util.*;

public class Ex {
    public static void main(String[] args) {
        // 알파벳과 숫자를 암호표로 암호화하기
        char[] abcCode =
            { '\'', '~', '!', '@', '#', '$', '%', '^', '&', '*',
              '(', ')', '-', '_', '+', '=', '|', '[', ']', '{',
              '}', ';', ':', ',', '.', '/' };

        //           0  1  2  3  4  5  6  7  8  9
        char[] numCode = { 'q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', 'p' };

        String src = "abc123";
        String result = "";
        // 문자열 src의 문자를 charAt()으로 하나씩 읽어서 변환 후 result에 저장

        for(int i=0; i < src.length(); i++) {
            char ch = src.charAt(i);
            if (ch >= 'a' && ch <= 'z') {
                result += abcCode[ch - 97]; // 혹은 ch - 'a'
            } else {
                result += numCode[ch - '0'];
            }
        }

        System.out.println("src:"+src);
        System.out.println("result:"+result);
    }
}
```

```
import java.util.*;

public class Ex {
    public static void main(String[] args) {
        // 주어진 2차원 배열의 데이터보다 가로와 세로로 10이 더 큰 배열을 생성해서
        // 배열의 행과 열의 마지막 요소에 각 열과 행의 총합을 저장하고 출력하는 프로그램

        int[][] score = {
            {100, 100, 100}
            , {20, 20, 20}
            , {30, 30, 30}
            , {40, 40, 40}
            , {50, 50, 50}
        };

        int[][] result = new int[score.length + 1][score[0].length + 1];

        for (int i = 0; i < score.length; i++) {
            for (int j = 0; j < score[i].length; j++) {
                result[i][j] = score[i][j];
                result[i][score[i].length] += result[i][j];
                result[score.length][j] += result[i][j];
                result[score.length][score[i].length] += result[i][j];
            }

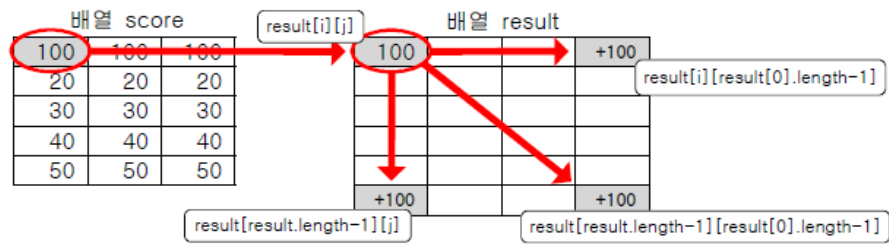
        }

        for (int i = 0; i < result.length; i++) {
            for (int j = 0; j < result[i].length; j++) {
                System.out.printf("%4d", result[i][j]);
            }

            System.out.println();
        }
    }
}
```

마지막 행에는 각 열의 총합이, 마지막으로 맨 오른쪽 행의 마지막 열에는 전체 총합이 저장되어야 한다. 언뜻 복잡해 보이지만 그림을 그려보면 간단명료해 진다.

먼저 `score[i][j]`를 `result[i][j]`에 저장하고, `result[i][j]`는 아래 그림에 표시한 세 곳에 누적해서 더해주면 된다.



`result.length-1`과 `result[0].length-1`은 `score.length`와 `score[0].length`와 각각 동일하므로 보다 식을 간단히 하기 위해 대체해서 사용했는데, 반드시 그렇게 해야 하는 것은 아니다.

```
result.length-1 → score.length
result[0].length-1 → score[0].length
```

`score[i][j]`의 값을 `result[i][j]`에 저장하기 때문에, `result[i][j]` 대신 `score[i][j]`를 사용해도 같은 결과를 얻는다.

```
result[i][score[0].length] += score[i][j];
result[score.length][j] += score[i][j];
result[score.length][score[0].length] += score[i][j];
```

```
import java.util.*;

public class Ex {
    public static void main(String[] args) {
        // 예제 5-23을 변경하여 아래와 같은 결과가 나오도록 하기.

        int count = 0;

        String[][] words = {
            {"chair", "의자"},           // words[0][0], words[0][1]
            {"computer", "컴퓨터"},      // words[1][0], words[1][1]
            {"integer", "정수"}          // words[2][0], words[2][1]
        };

        Scanner scanner = new Scanner(System.in);

        for (int i = 0; i < words.length; i++) {
            System.out.printf("Q%d. %s의 뜻은? ", i + 1, words[i][0]);

            String tmp = scanner.nextLine();

            if (tmp.equals(words[i][1])) {
                System.out.printf("정답입니다.%n%n");
                count++;
            } else {
                System.out.printf("틀렸습니다. 정답은 %s입니다.%n%n", words[i][1]);
            }
        }

        System.out.printf(" 전체 %d문제 중 %d문제 맞추셨습니다.", words.length, count);
    }
}
```

---

```

import java.util.*;

public class Ex {
    public static void main(String[] args) {
        // 단어의 글자 위치를 섞어서 보여주고 원래의 단어를 맞추는 예제.
        String[] words = {"television", "computer", "mouse", "phone"};

        Scanner scanner = new Scanner(System.in);

        for (int i = 0; i < words.length; i++) {
            char[] question = words[i].toCharArray(); // String을 char[]로 변환

            // char 배열 question에 담긴 문자의 위치를 임의로 바꾼다.
            for (int j = 0; j < question.length; j++) {
                int random = (int) (Math.random() * question.length);
                char temp;
                temp = question[i]; // question[j]도 무관. i는 i번째 글자(0~3), 같은 글자를
                // j는 0부터 question.length-1글자를 한번씩, 총 10번 섞음.
                question[i] = question[random];
                question[random] = temp;
            }

            System.out.printf("Q%d. %s의 정답을 입력하세요.>", i + 1, new String(question));
            String answer = scanner.nextLine();

            // trim()으로 answer의 좌우 공백을 제거한 후, equals로 word[i]와 비교
            if (words[i].equals(answer.trim())) {
                System.out.printf("맞았습니다.%n%n");
            } else {
                System.out.printf("틀렸습니다.%n%n");
            }
        }
    }
}

```

Resource: 자바의 정석