

<배열 실습 문제>

□ 사용자로부터 일차원 배열의 원소(성분) 개수를 입력 받아 그 만큼의 double 타입 배열을 생성하시오. 또한, 아래와 같은 작업을 하는 메소드를 만들어 테스트하시오.

- `initArr(double [] arr)`: 배열의 성분을 0이상 10미만의 랜덤 넘버로 초기화
- `printArr(double [] arr)`: 배열의 성분을 출력
- `sumArr(double [] arr)`: 배열의 모든 성분의 합을 반환
- `maxArr(double [] arr)`: 배열의 최댓값을 반환
- `indexMaxArr(double [] arr)`: 배열의 최댓값의 인덱스를 반환
- `shuffleArr(double [] arr)`: 배열의 성분의 순서를 랜덤으로 재배치
- `rotateRlArr(double [] arr)/rotateLlArr(double [] arr)`: 배열의 성분을 오른쪽(왼쪽)으로 한 칸 로테이션
- `rotateRnArr(double [] arr, int n)/rotateLnArr(double [] arr, int n)`: 배열의 성분을 오른쪽(왼쪽)으로 n칸 로테이션

실행 예시)

배열의 원소 개수: 8

배열 성분:

0.71 3.43 7.09 7.88 3.30 9.61 7.55 1.61

합: 41.19

배열의 최댓값: 9.61

배열의 최댓값의 인덱스: 6

셔플 후 배열 성분:

1.61 9.61 7.88 7.09 3.30 3.43 7.55 0.71

오른쪽으로 1칸 로테이션:

0.71 1.61 9.61 7.88 7.09 3.30 3.43 7.55

왼쪽으로 1칸 로테이션:

1.61 9.61 7.88 7.09 3.30 3.43 7.55 0.71

오른쪽 시프트 횟수: 22

오른쪽으로 22칸 로테이션:

7.88 7.09 3.30 3.43 7.55 0.71 1.61 9.61

왼쪽 시프트 횟수: 14

왼쪽으로 14칸 로테이션:

1.61 9.61 7.88 7.09 3.30 3.43 7.55 0.71

□ 3행 2열의 정수형 이차원 배열을 생성하고, 아래와 같은 메소드를 만들어 테스트하시오.

- `initArr(int [][] arr)`: 배열의 성분을 사용자가 입력한 값으로 초기화
- `printArr(int [][] arr)`: 배열의 성분을 출력
- `sumArr(int [][] arr)`: 배열의 모든 성분의 합을 반환
- `colSumArr(int [][] arr)`: 배열의 열 별 합을 반환
- `rowSumMaxArr(int [][] arr)`: 배열의 행 별 합의 최댓값을 반환
- `indexMaxArr(int [][] arr)`: 배열의 최댓값의 인덱스를 반환
- `shuffleArr(int [][] arr)`: 배열의 성분의 순서를 랜덤으로 재배치

실행 예시)

```
1행 1열 성분: 5
1행 2열 성분: 4
2행 1열 성분: 1
2행 2열 성분: 9
3행 1열 성분: 15
3행 2열 성분: 2

배열 성분:
5 4
1 9
15 2

모든 성분의 합: 36
배열의 열 별 합: 21 15 행 별 합의 최댓값: 17
배열의 최댓값의 인덱스: (3, 1)

셔플 후 배열 성분:
1 15
9 4
5 2
```