

### 과제 #3 - 평균 원소

[6~11]

배열을 입력받을 정수형 벡터 `vec`, 길이 `n`을 입력받을 정수형 `num`, 입력받은 원소를 임시 저장할 변수 `temp`, 연속된 인덱스를 가지는 부분 배열 `i~j`까지의 길이를 나타내주는 실수형 변수 `compare`, 평균 원소를 가지는 부분 배열의 개수를 나타내는 변수 `answer` 을 선언

[13~18]

사용자로부터 배열 `A`에 속한 원소의 개수를 나타내는 양의 정수 `num`을 매개변수로 받아 반복문을 통해서 벡터에 원소를 저장하는 함수.

[20~49]

부분배열의 길이를 나타내는 `n`을 매개변수로 받아 평균 원소를 갖는 부분 배열의 개수를 찾는 `check` 재귀 함수이다.

먼저 사용된 변수를 설명하자면 부분 배열의 원소들을 다 더한 값을 저장할 실수형 `add` 변수와 `add`를 부분 배열의 길이(`compare+1`) 만큼 나누어 나온 실수형 값을 저장할 `division` 변수와 이진 트리를 사용하기 위해 `sort`를 해야 하는데 `vec`를 사용하면 값이 변경되므로 `vec`의 값을 간접적으로 사용할 `vec2`를 선언하였다.(순환할 때마다 초기화해야 하기 때문에 `for`문 안에서 선언)

첫 번째 `for`문은 `i`를 시작 원소 인덱스로 사용하고, 부분 배열의 최대길이가 `num`이기 때문에 `i+compare < num` 까지의 길이를 갖는 모든 부분배열들의 각 합을 구하기 위해 이중 `for`문을 통해 `i, i+1, ... ,i+compare` 까지의 원소를 다 더하여 `add`에 저장한 후 `j==compare` 이면 그 `add`를 부분 배열의 길이인 `compare+1`으로 나누어 `division`에 입력한다. 그리고 해당 부분배열이 갖는 범위를 `sort`하여 이진 트리를 사용한 뒤 `division`이 있다면 `answer`을 1 증가시킨다.

`Compare`이 1인 경우, 2인 경우, 3인 경우 `num`인 경우를 다 구해야 하기 때문에 해당 `compare` 값에서 루프를 다 돌았다면 `compare`를 1 증가시켜 재귀한다.

[52~59]

`scanf`를 통해 길이를 입력받고, `makeVec`을 통해 배열을 생성 후 `check`함수를 호출