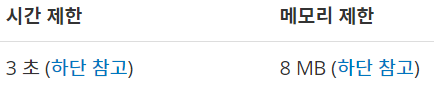
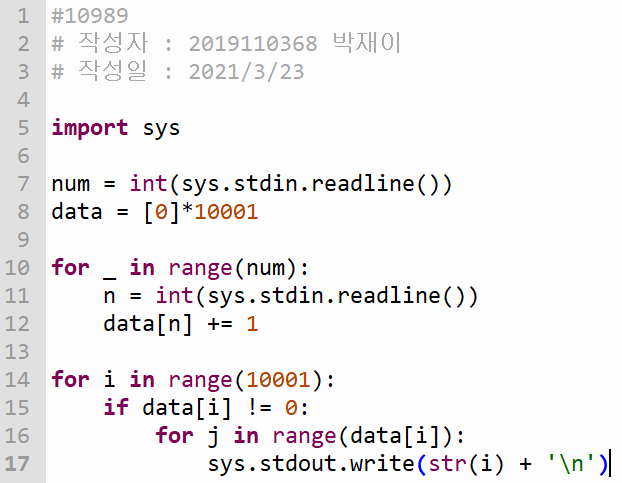
#10989 – 수 정렬하기3

[문제]





[소스코드]



[출력 결과]



<입력> <출력>

[풀이]

문제가 쉬워 보이지만 시간, 메모리 제한이 있기 때문에 input()을 sys.stdin.readline()으로 바꾸고 print()를 sys.stdout.write()으로 바꾼다. 이 문제는 메모리가 낮아서 정렬을 사용하지 않고 배열을 사용한다. N의 최댓값인 10000에 + 1 한만큼 배열을 먼저 생성해 주고 date에 입력을 받은 후 각 숫자의 등장 횟수를 카운트한다. 리스트에 각 요소마다 0을 할당해주고 입력 값을 받을 때마다 그 입력값과 같은 인덱스에 +1을 해준다. 만약 한 번도 등장하지 않아 초반에 선언했던 0이 그대로 남아있는 원소는 생략하고, 카운트되어 있는 숫자들만 출력한다.

는 로 선언해도 된다.

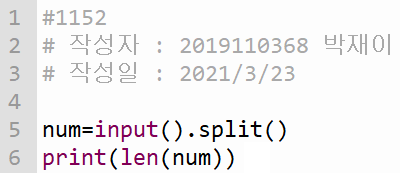
도 로 선언해도 된다.

#1152 – 단어의 개수

[문제]



[소스코드]



[출력 결과]





[풀이]

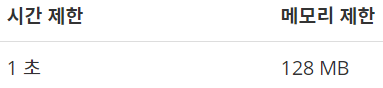
split(): 주어진 문자열을 띄어쓰기를 기준으로 나누어 준다.

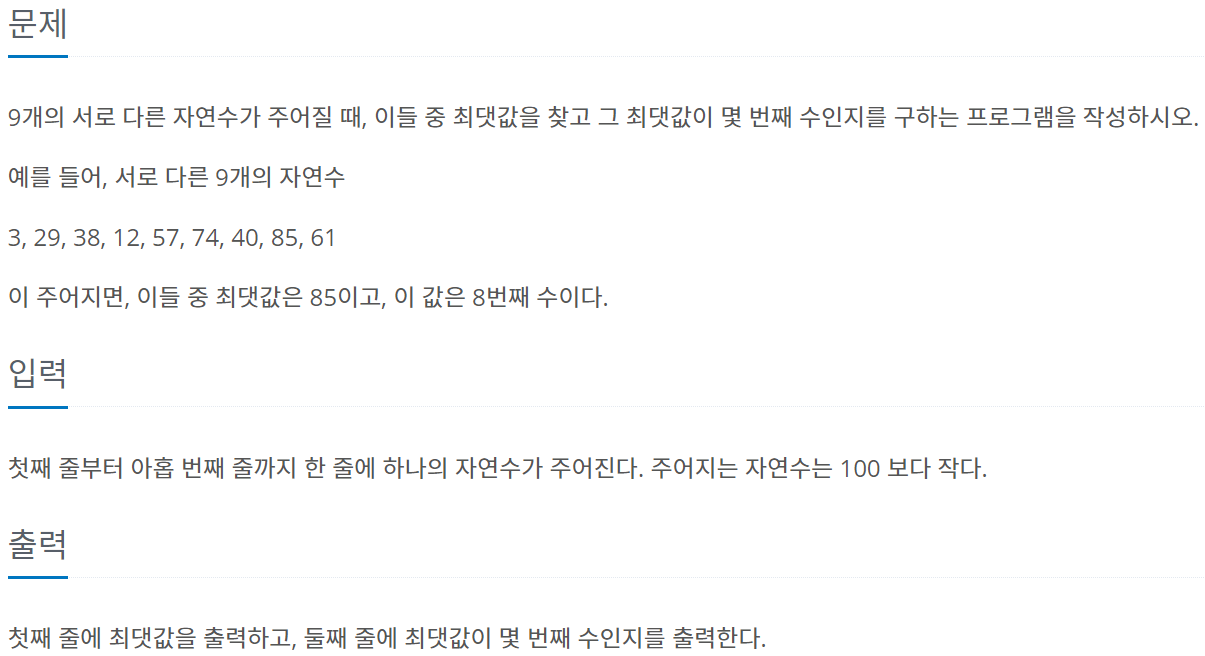
input 함수를 이용하여 문자열을 입력 받고, split 함수를 이용하여 공백을 제거한 단어들을 변수num에 저장한다. num의 원소의 개수를 샌 뒤 출력시킨다.

로 표현해도 같은 결과가 나온다.

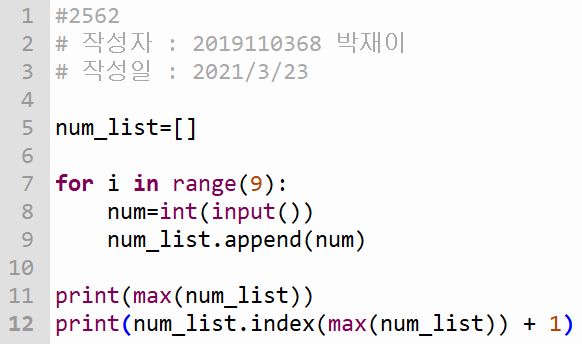
#2562 – 최댓값

[문제]





[소스코드]



[출력 결과]



<입력> <출력>

[풀이]

num\_list 라는 리스트를 생성하고 숫자를 입력 받기 위해 for문을 써서 9번 반복한다. 입력 받은 숫자를 num\_list 리스트에 추가한다. Max 함수를 사용하여 num\_list 리스트에서 가장 큰 숫자를 출력하고 index 함수로 최댓값의 위치를 찾는다. 인덱싱은 숫자를 셀 때 0부터 시작하기 때문에 1을 더해서 1부터 시작하게 만들어 준다.