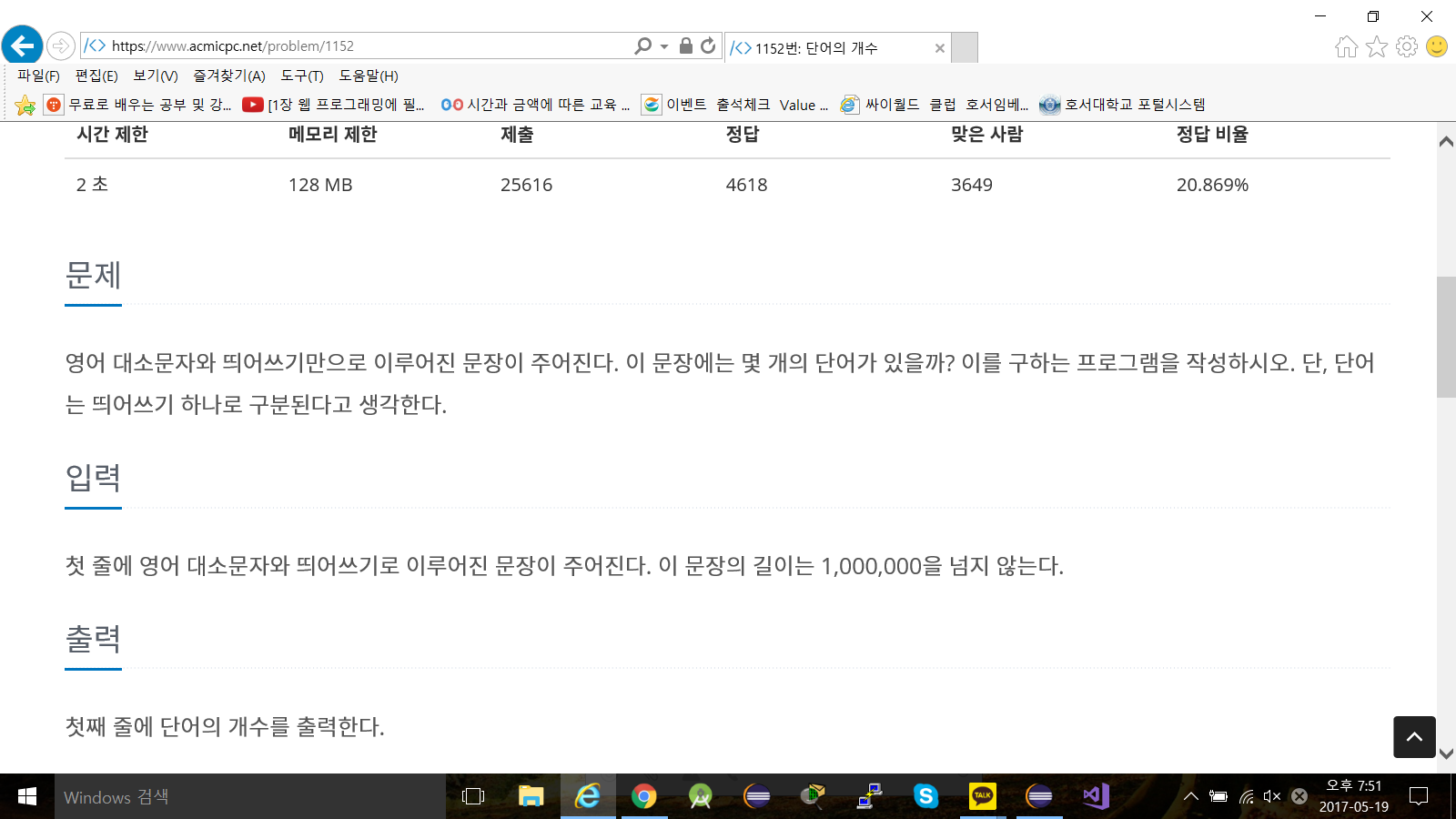
question

answer1 -wrong // i don’t know why it’s wrong ;;

import java.util.Scanner;

public class T\_1152 {

public static void main(String[] args) {

Scanner scan=new Scanner(System.in);

String word=scan.nextLine();

//StringTokenizer token = new StringTokenizer(word);

String[] dvide=word.split(" ");

System.out.println(dvide.length);

}

}

answer2 -

import java.util.Scanner;

public class T\_1152 {

public static void main(String[] args) {

Scanner scan=new Scanner(System.in);

String input=scan.nextLine();

int check=0;

//StringTokenizer token = new StringTokenizer(word);

String[] words=input.split(" ");

for(int i=0;i<words.length;i++){

if(words[i].equals("")){

check++;

}

}

System.out.println(words.length-check);

}

}

question2

셀프 넘버는 1949년 인도 수학자 D.R. Kaprekar가 이름 붙였다. 양의 정수 n에 대해서 d(n)을 n과 n의 각 자리수를 더하는 함수라고 정의하자. 예를 들어, d(75) = 75+7+5 = 87이다.

양의 정수 n이 주어졌을 때, 이 수를 시작해서 n, d(n), d(d(n)), d(d(d(n))), ...과 같은 무한 수열을 만들 수 있다.

예를 들어, 33으로 시작한다면 다음 수는 33 + 3 + 3 = 39이고, 그 다음 수는 39 + 3 + 9 = 51, 다음 수는 51 + 5 + 1 = 57이다. 이런식으로 다음과 같은 수열을 만들 수 있다.

33, 39, 51, 57, 69, 84, 96, 111, 114, 120, 123, 129, 141, ...

n을 d(n)의 생성자라고 한다. 위의 수열에서 33은 39의 생성자이고, 39는 51의 생성자, 51은 57의 생성자이다. 생성자가 한 개보다 많은 경우도 있다. 예를 들어, 101은 생성자가 2개(91과 100) 있다.

생성자가 없는 숫자를 셀프 넘버라고 한다. 100보다 작은 셀프 넘버는 총 13개가 있다. 1, 3, 5, 7, 9, 20, 31, 42, 53, 64, 75, 86, 97

10000보다 작거나 같은 셀프 넘버를 한 줄에 하나씩 출력하는 프로그램을 작성하시오.

answer - good

public class T\_4673 {

static boolean [] number= new boolean[9999];

public static void main(String[] args) {

for(int i=0;i<9999;i++){

calculation(i);

}

}

public static void calculation(int a){

if(!number[a]){

System.out.println(a+1);

}

a++;

int sum=a;

while(a!=0){

sum+=a%10;

a/=10;

}

if(sum<10000){

number[sum-1]=true;

}

}

}