

static int maxXor(int left, int right, int k) {

int max = 0;

for (int i = left; i < right; i++) {

for (int j = i + 1; j <= right; j++) {

int xor = i ^ j;

if (xor > max)

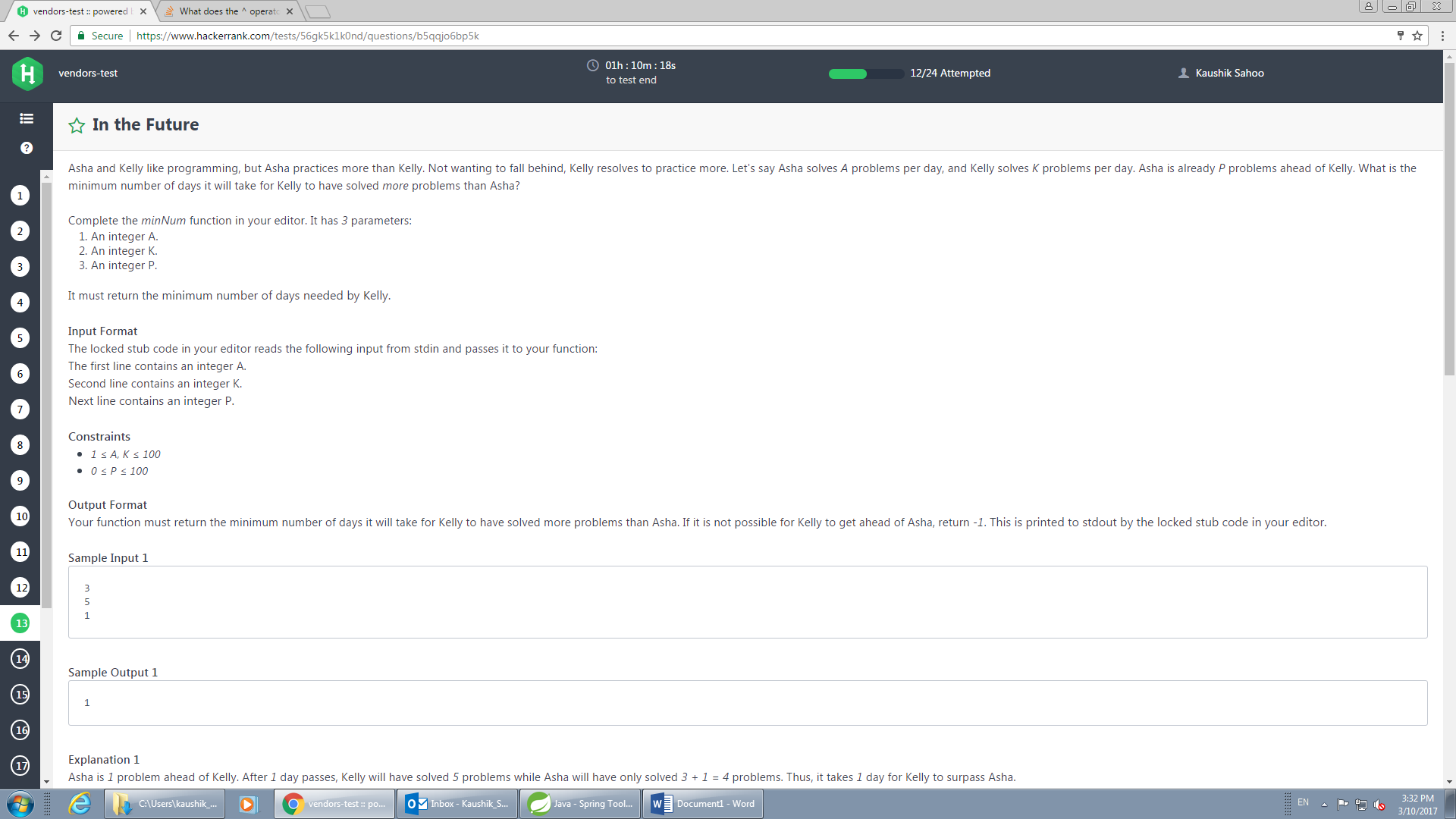
max = xor;

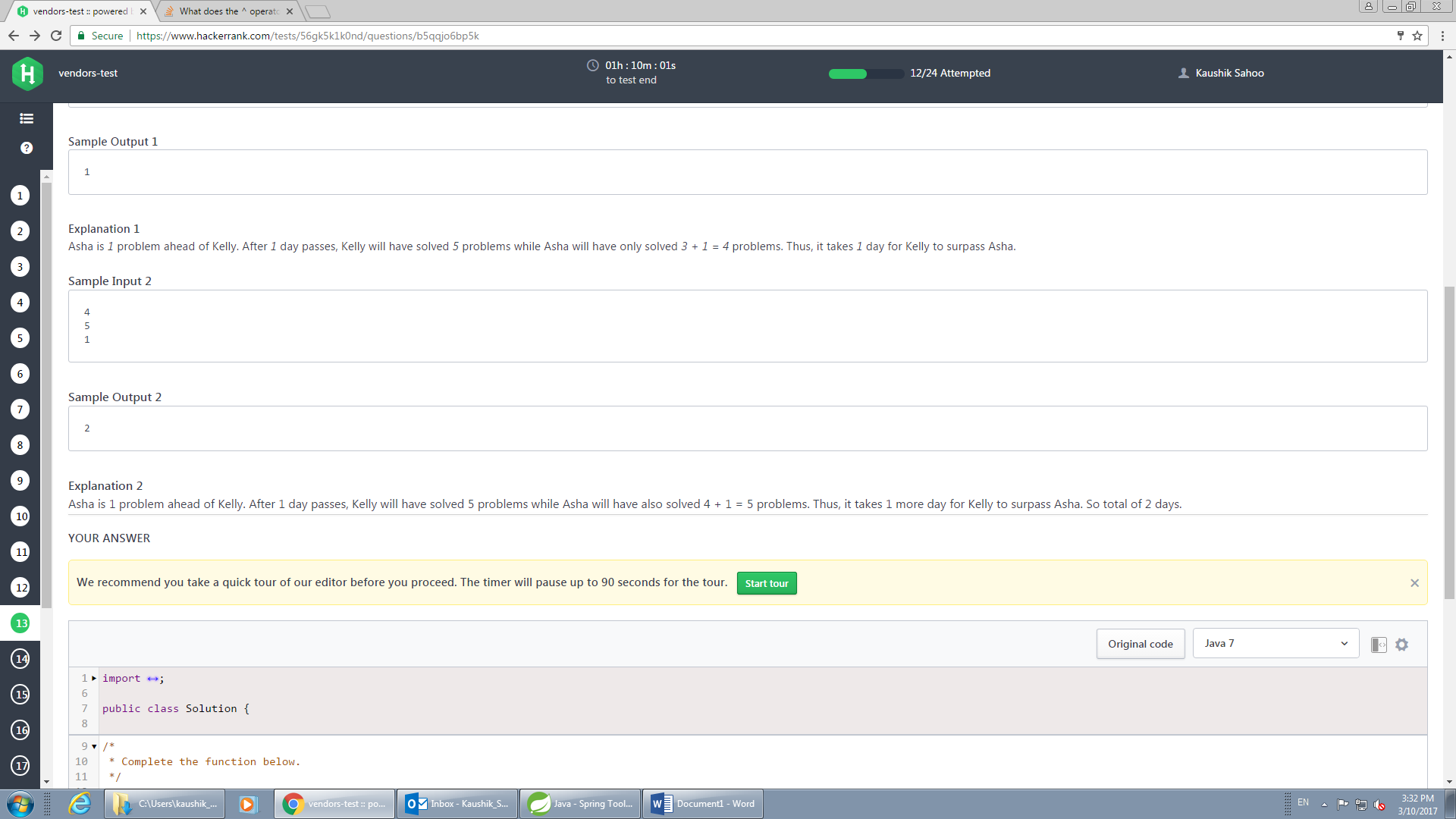
}

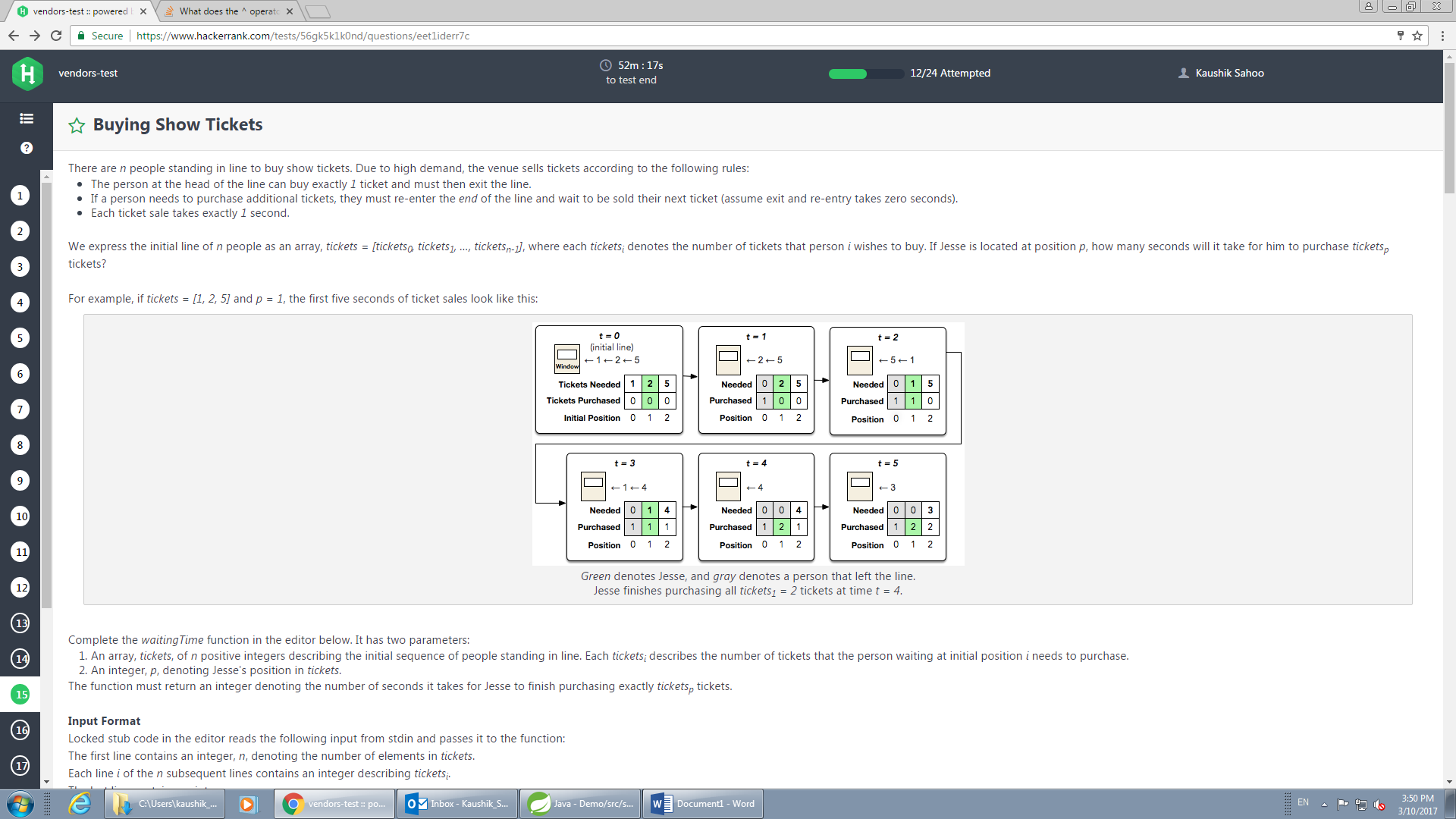
}

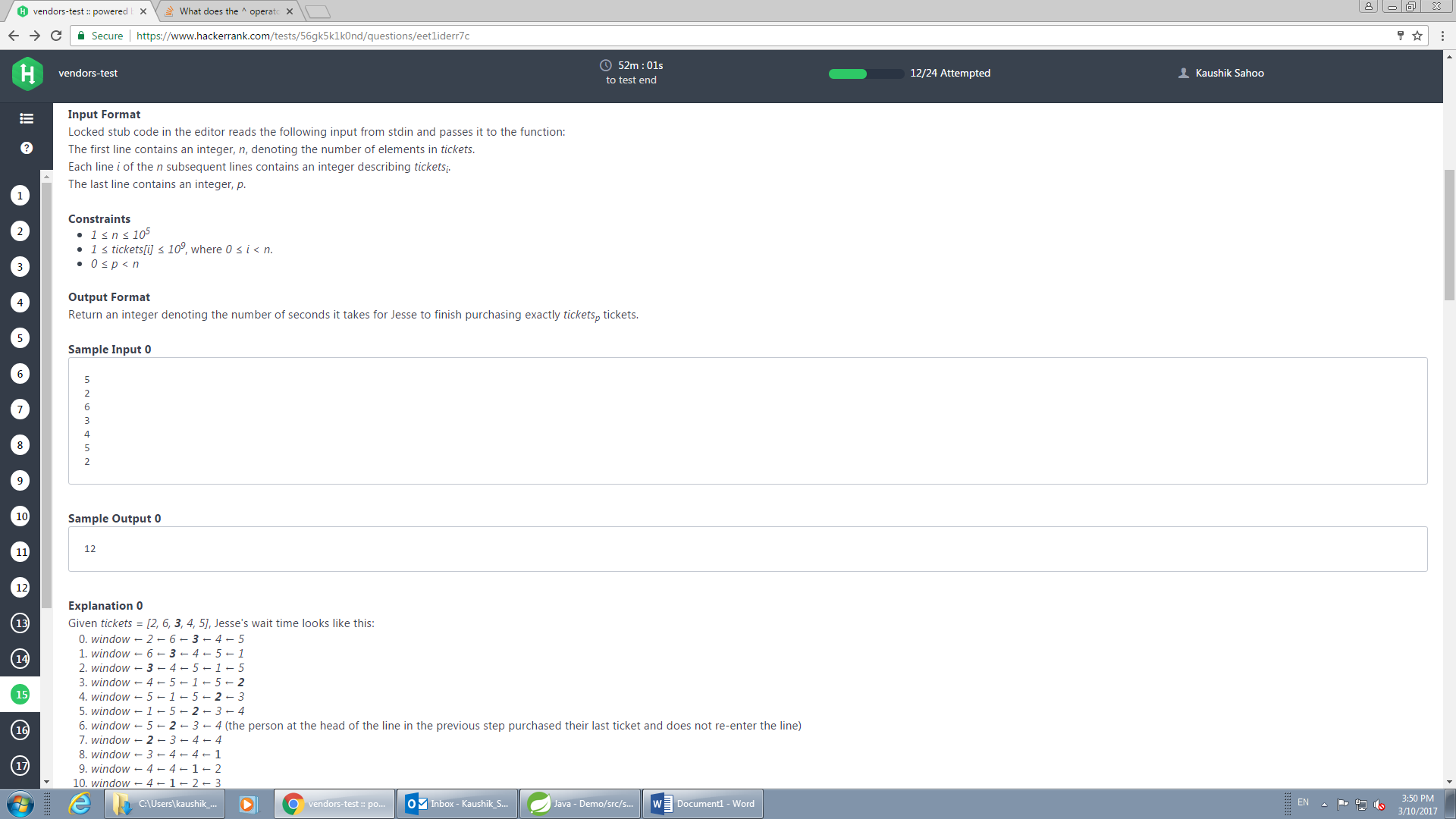
return max;

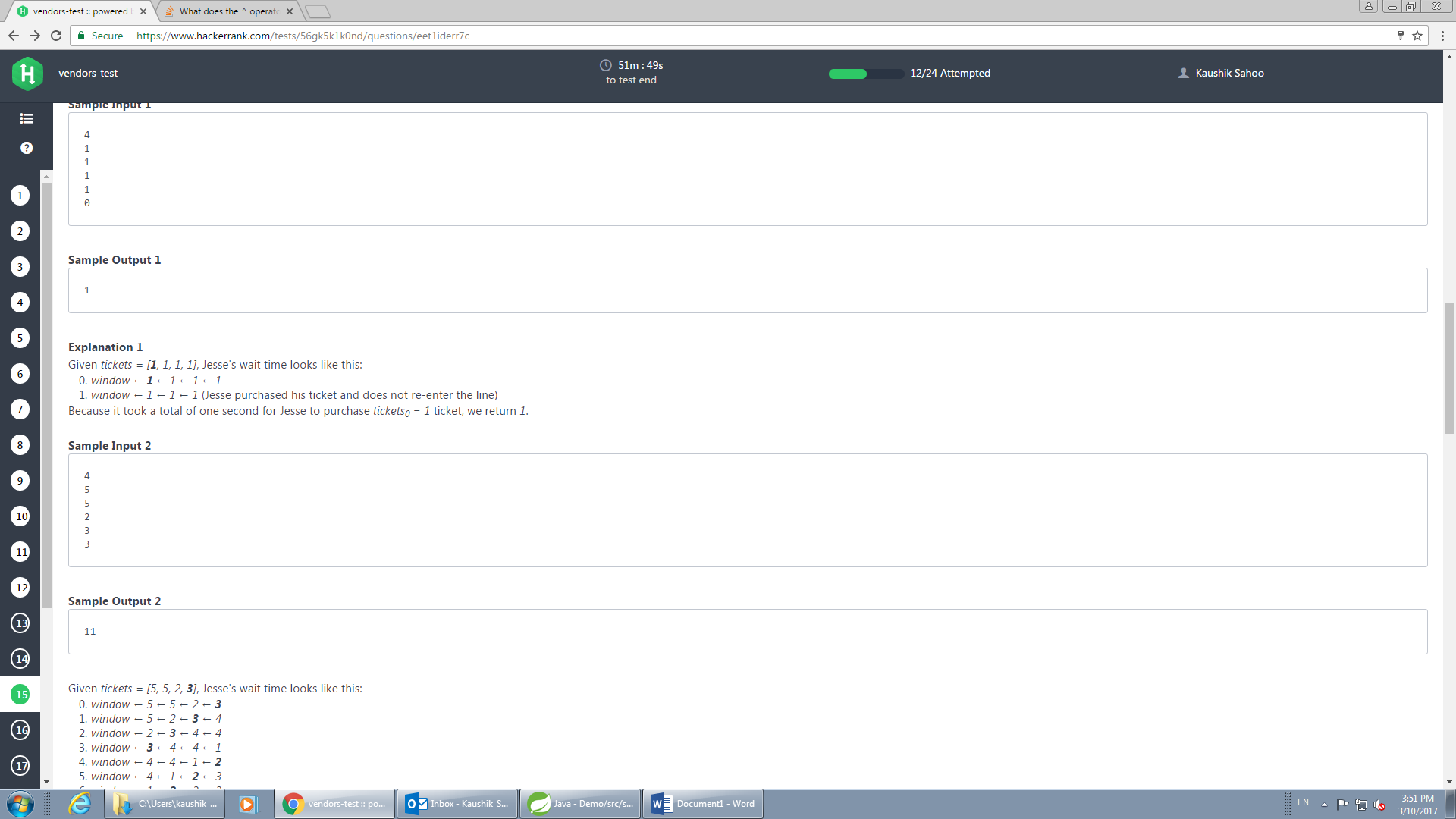
}











static long waitingTime(int[] tickets, int p) {

long time = 0;

int np = tickets[p];

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

if (i < p) {

time += time(tickets[i], np);

} else if (i == p) {

time += np;

} else {

time += (time(tickets[i], np) - 1);

}

}

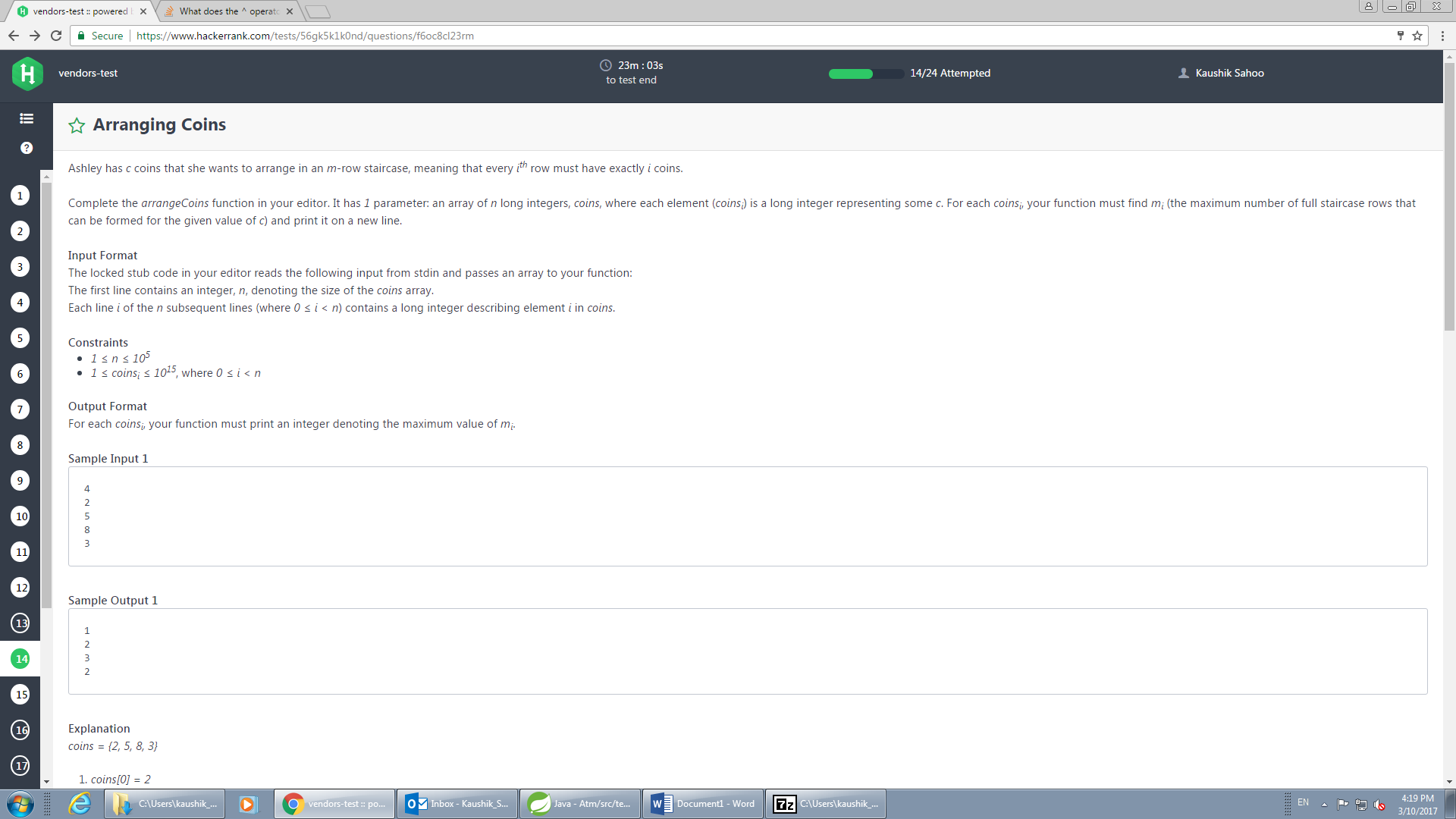
return time;

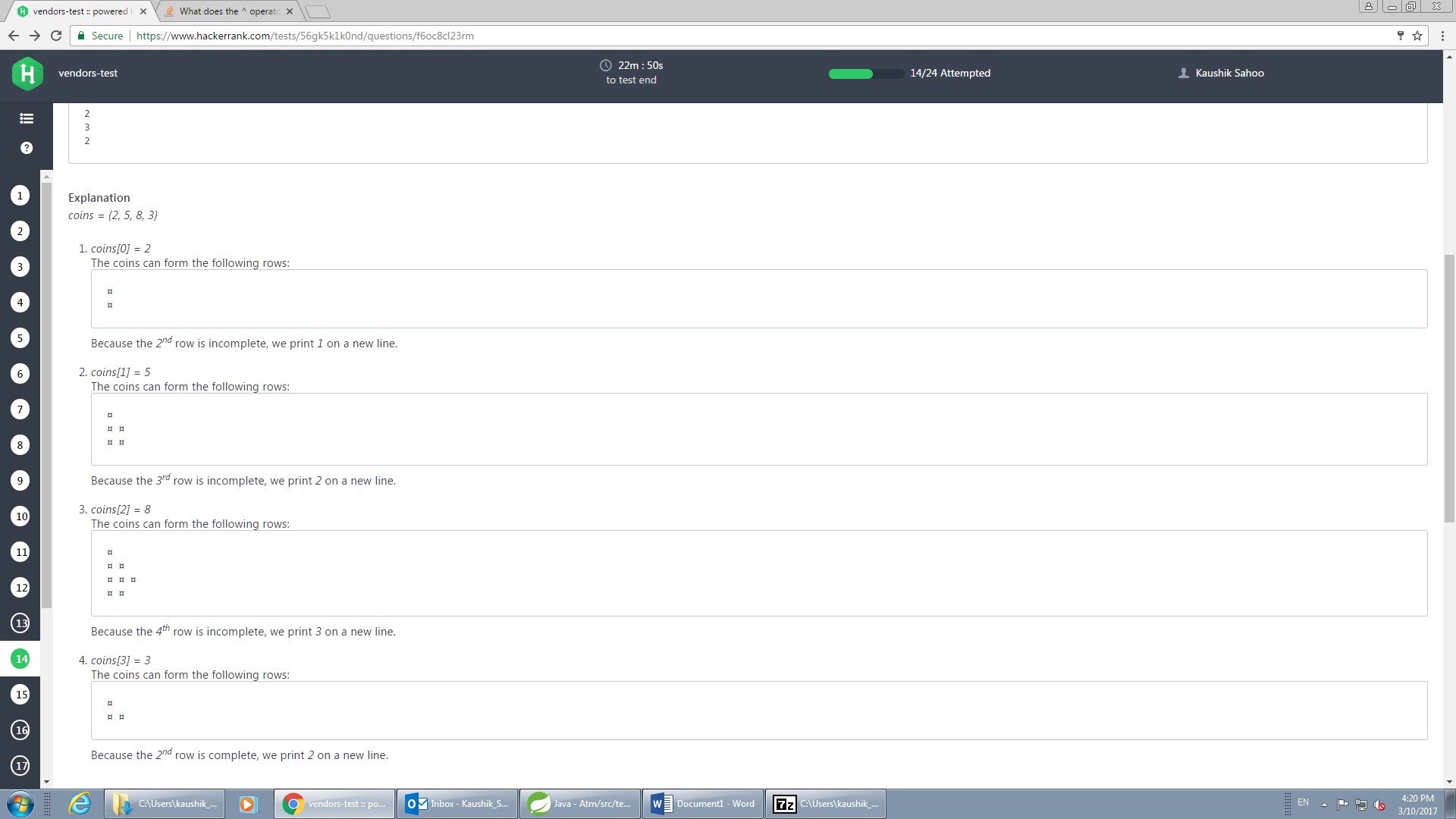
}

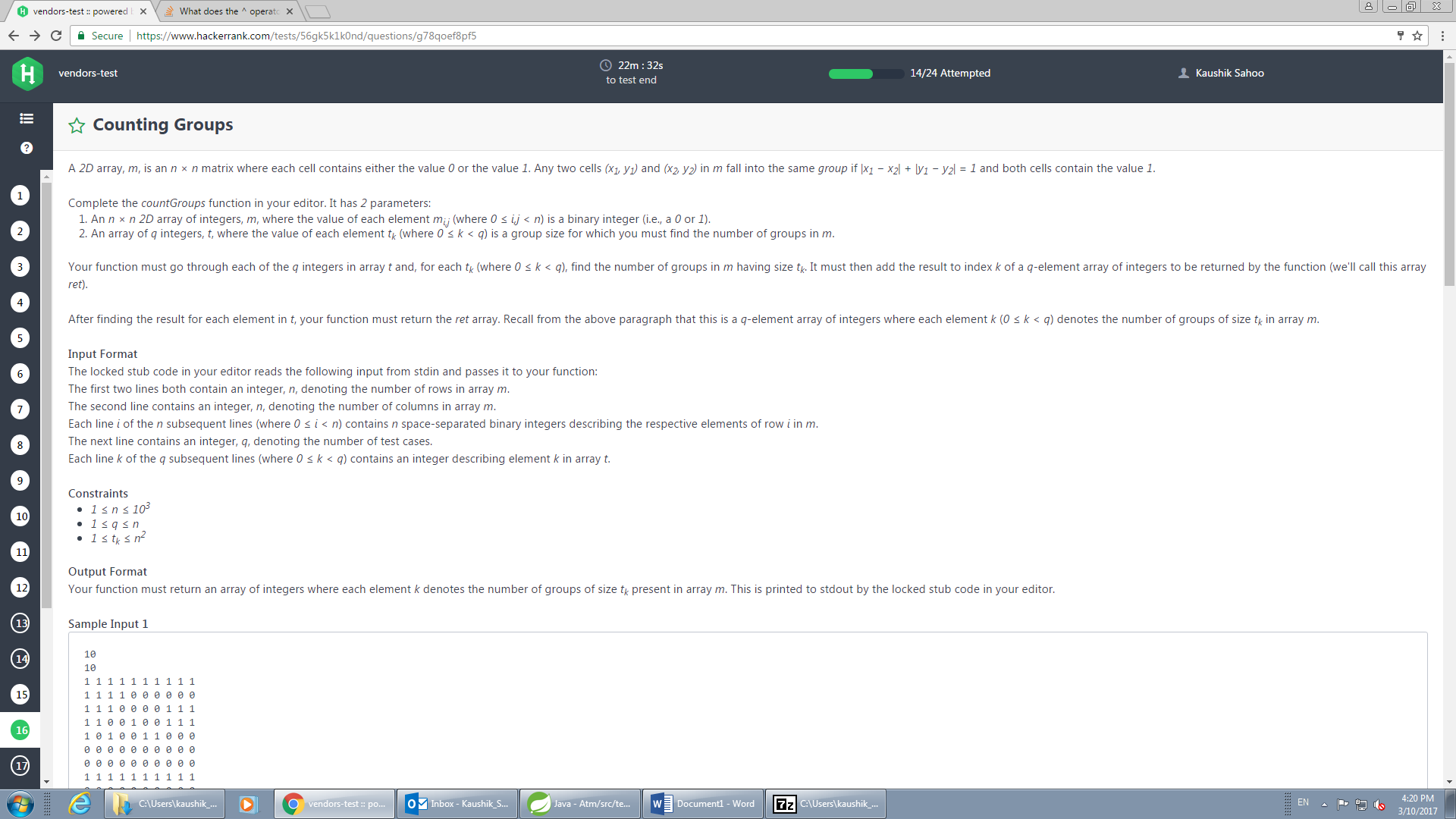
static int time(int i, int np) {

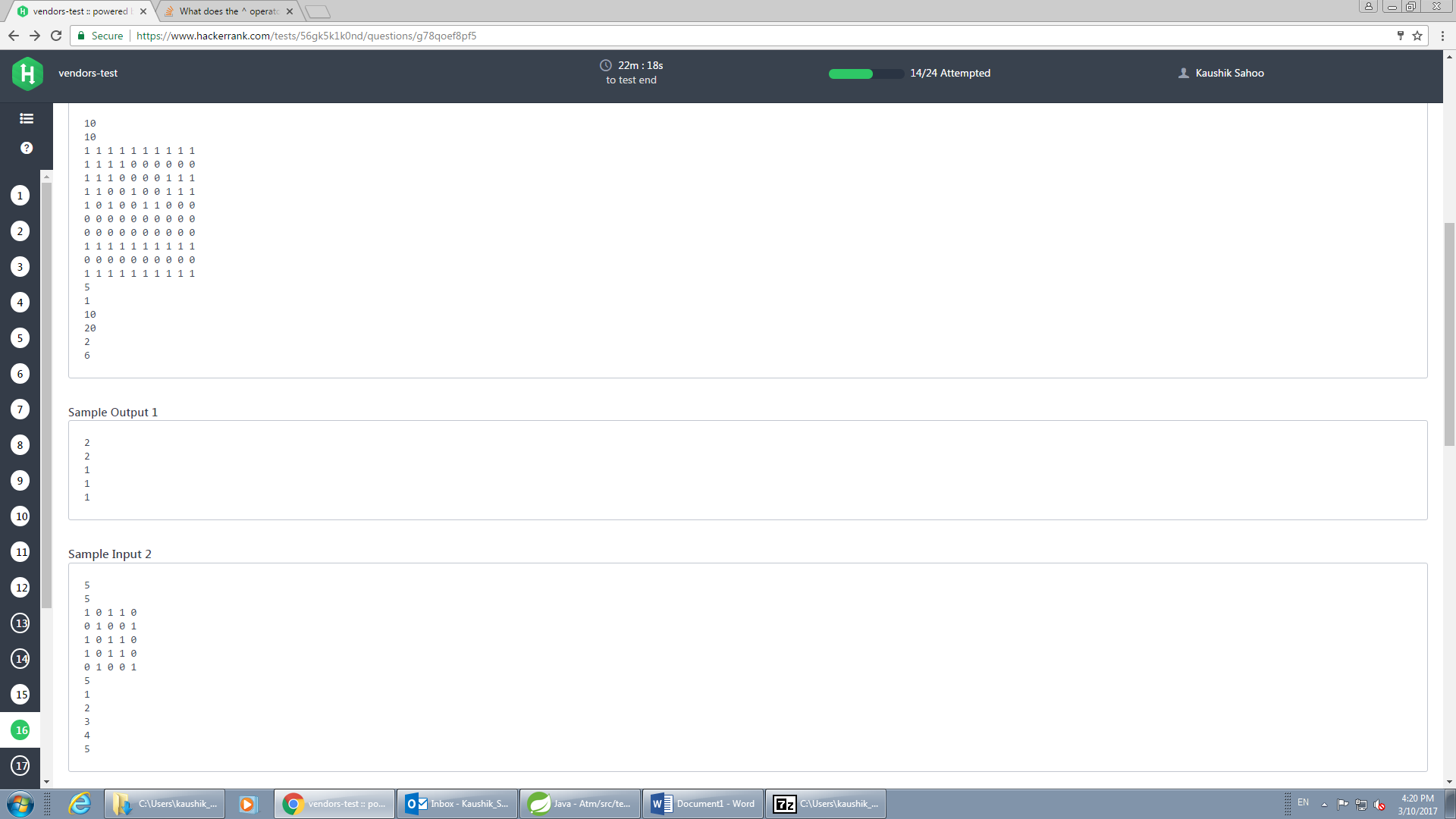
return (i < np) ? i : np;

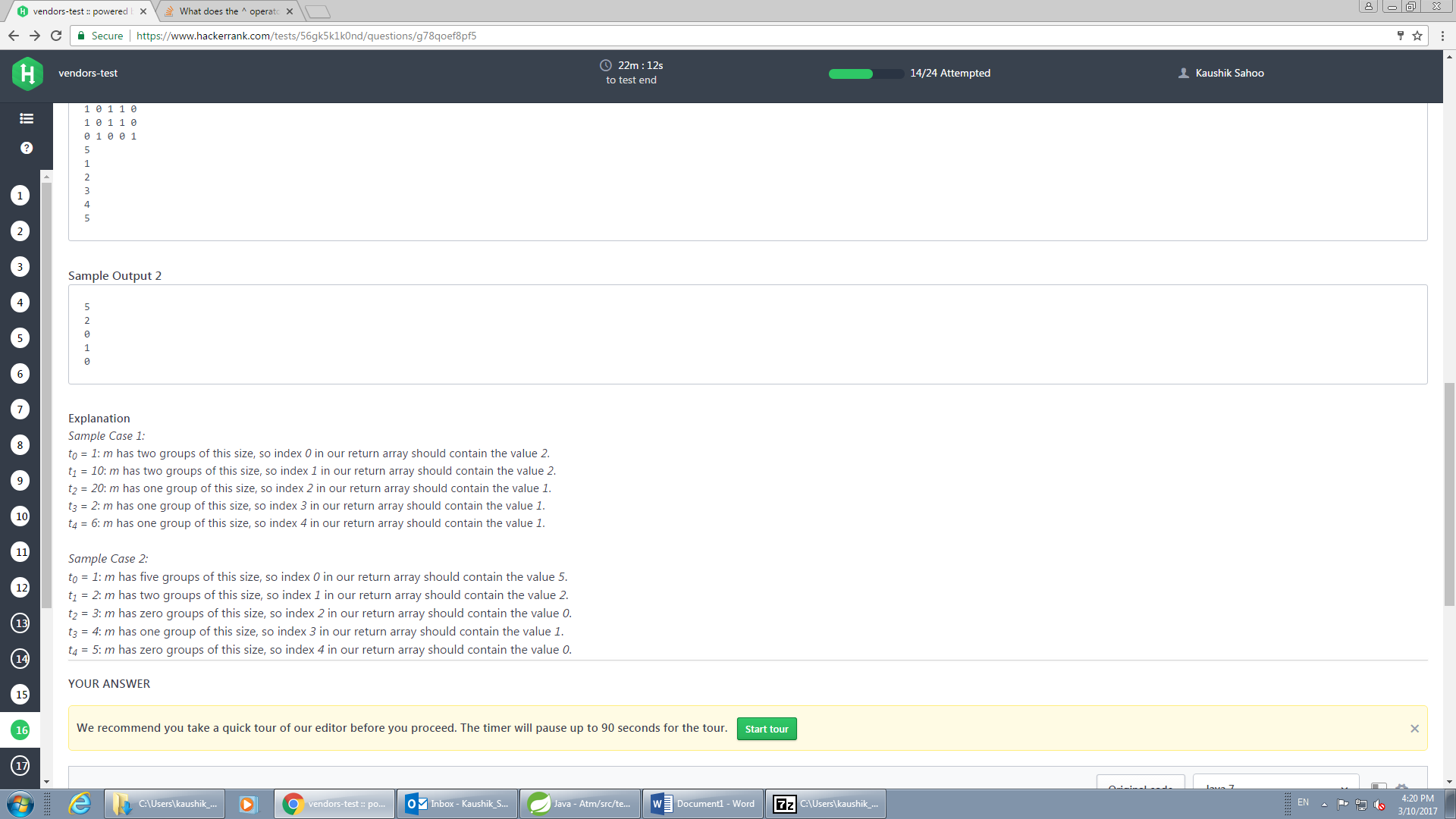
}

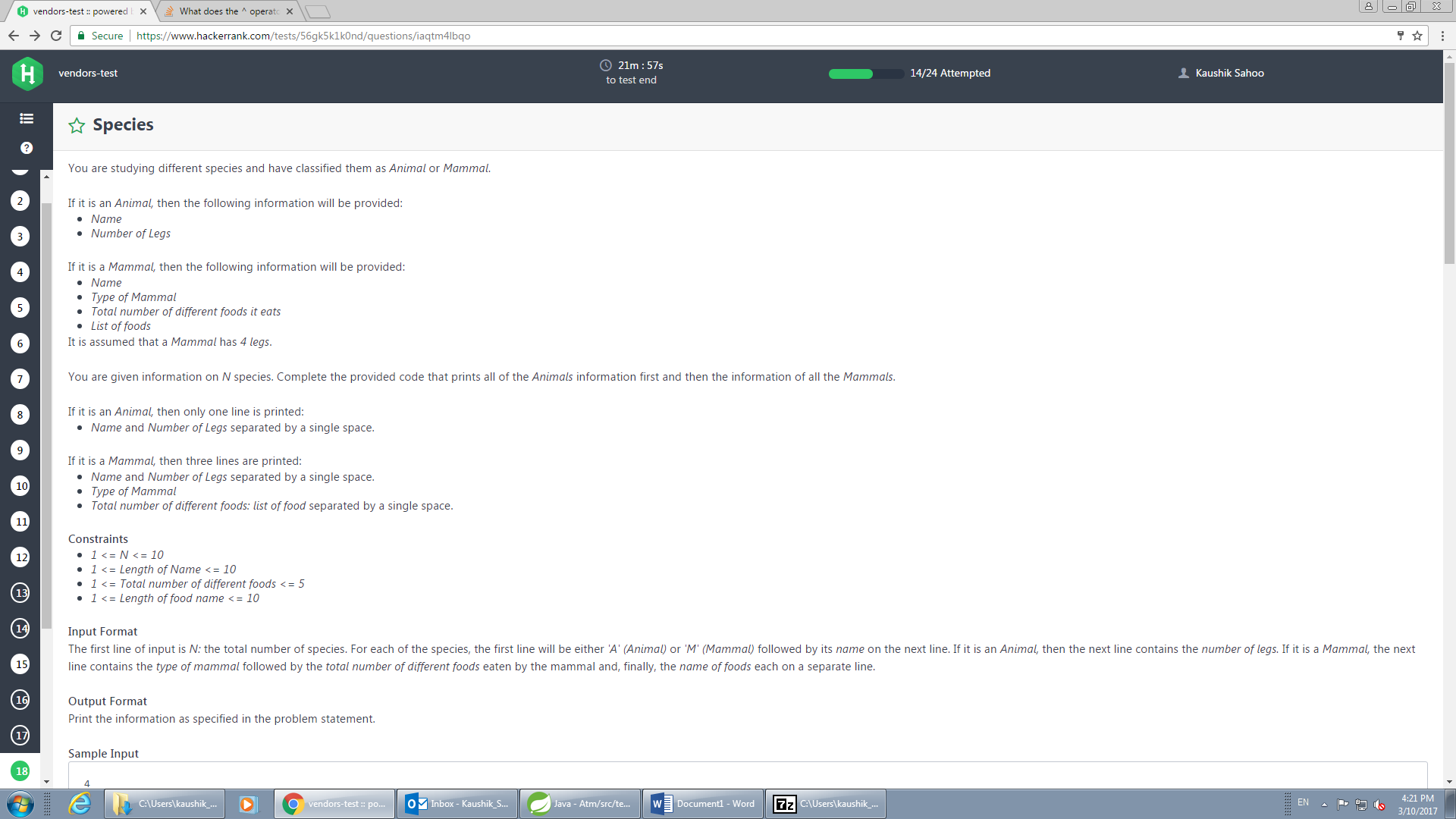


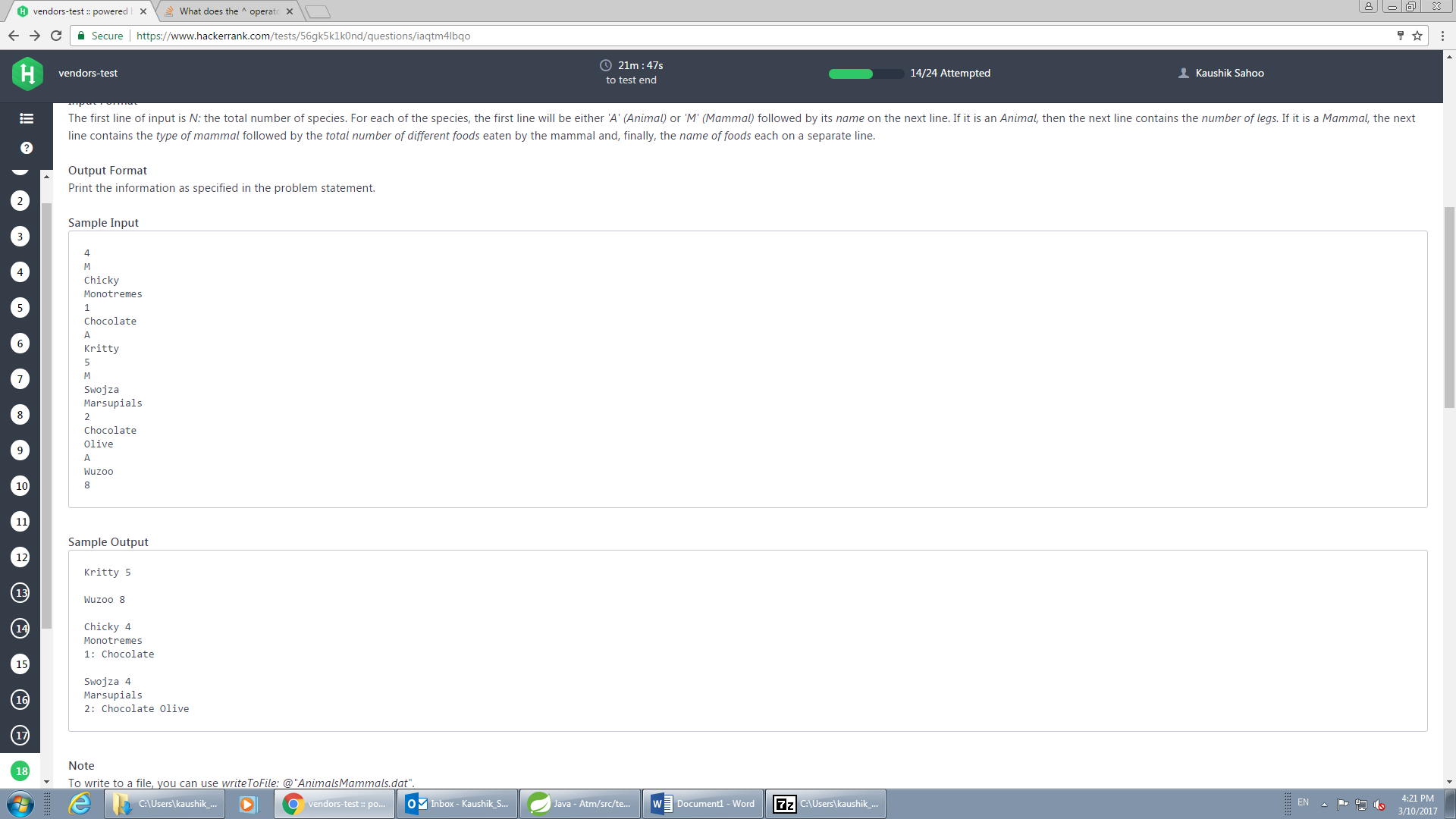


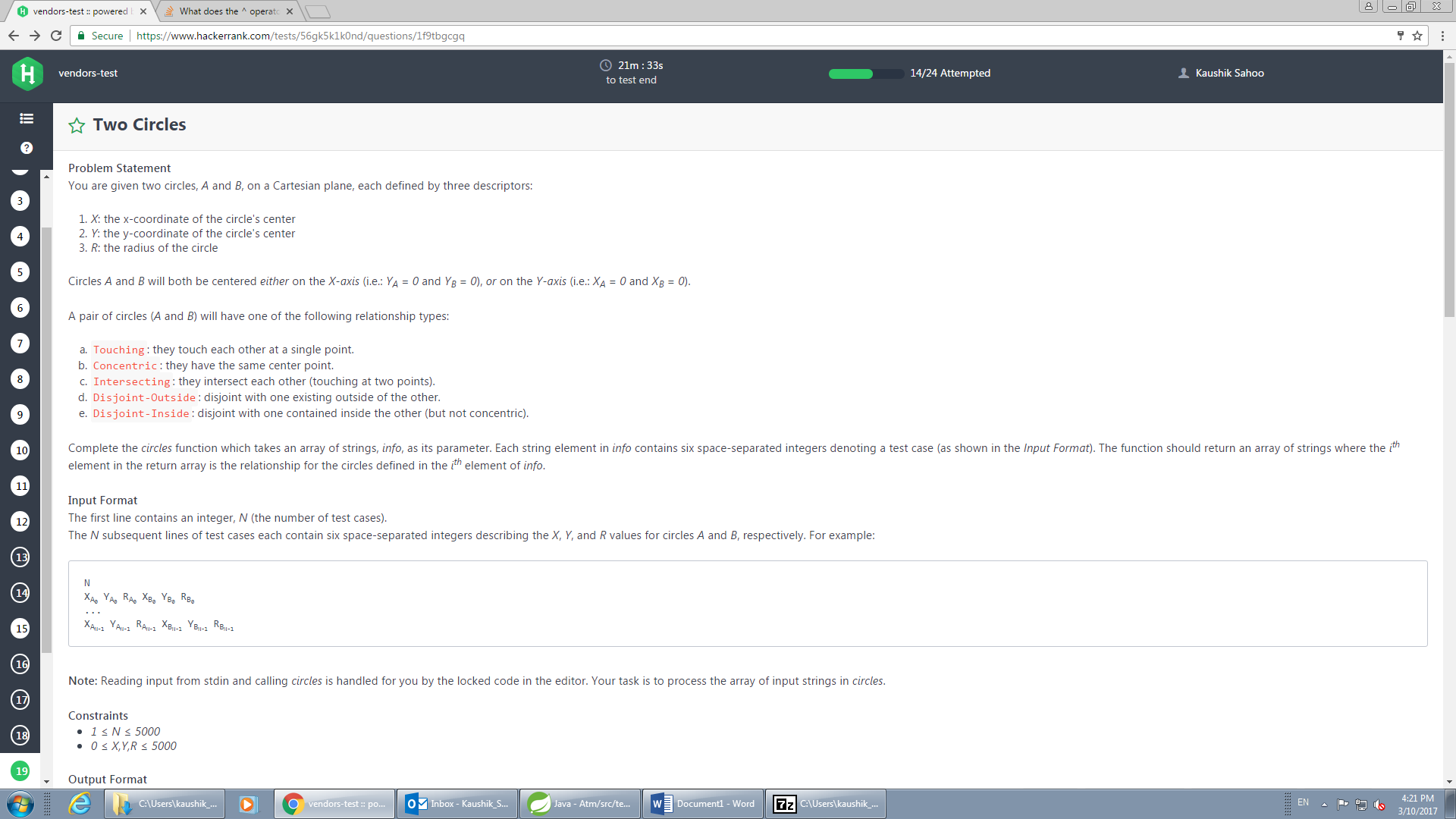


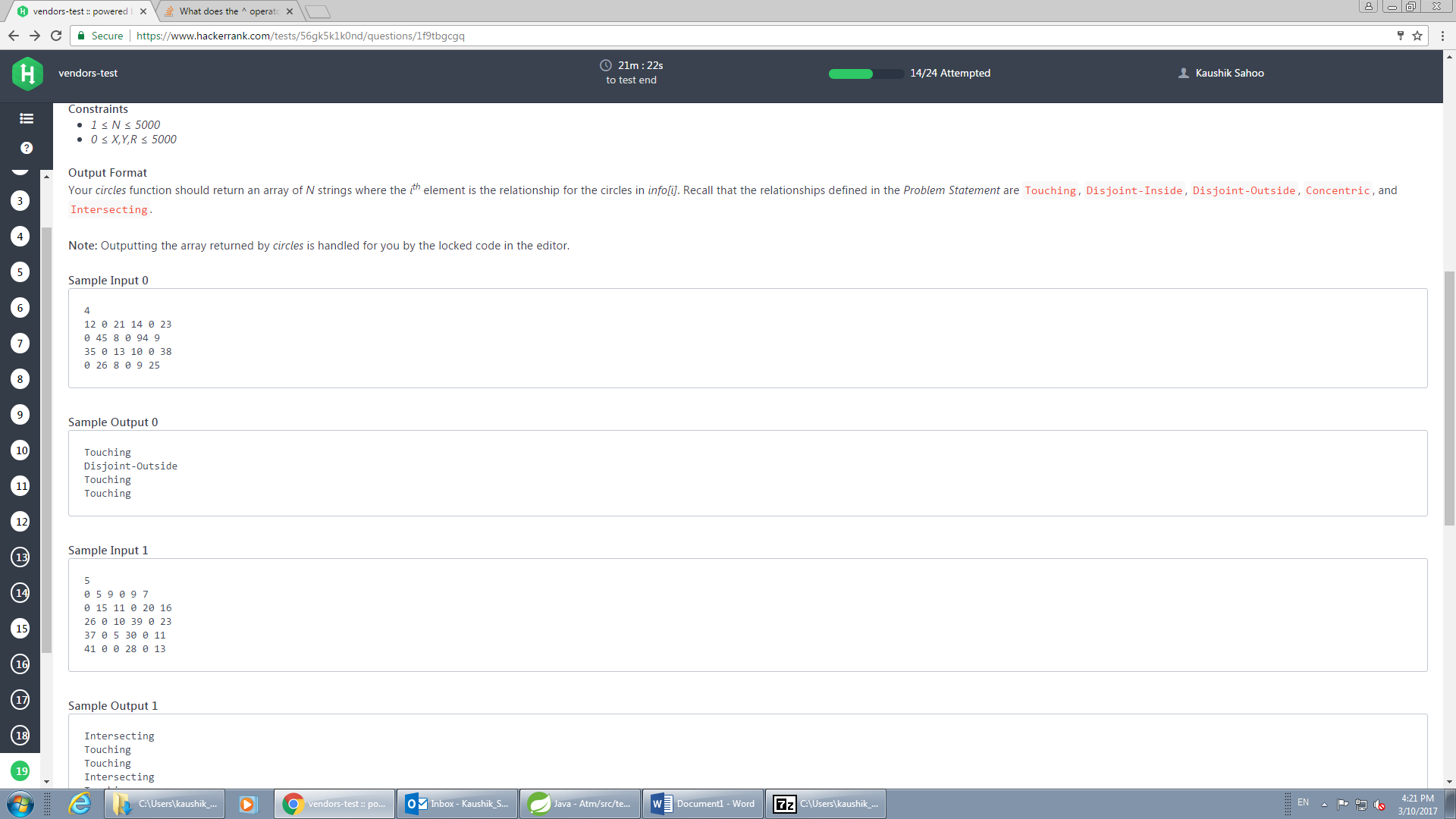


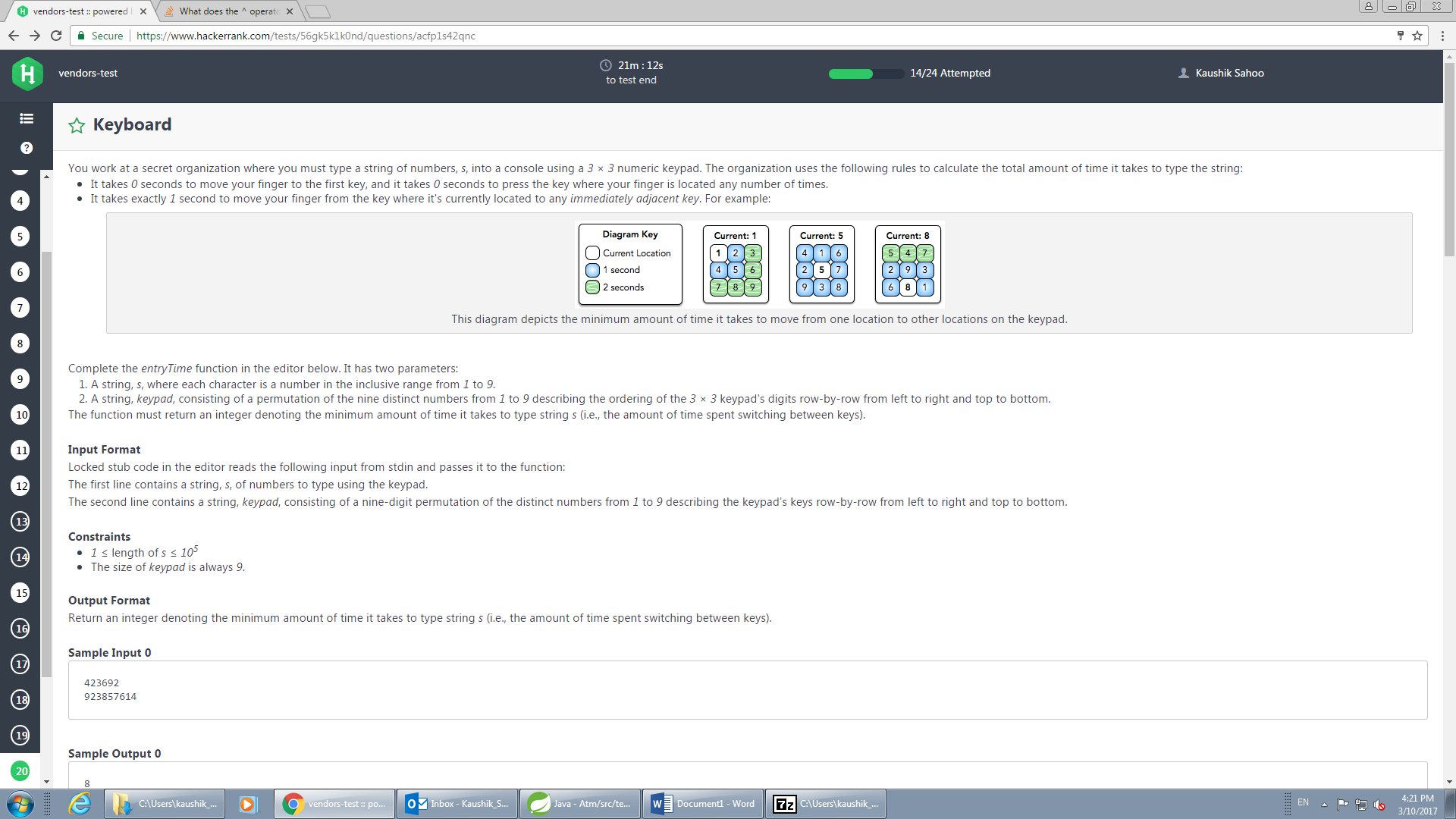


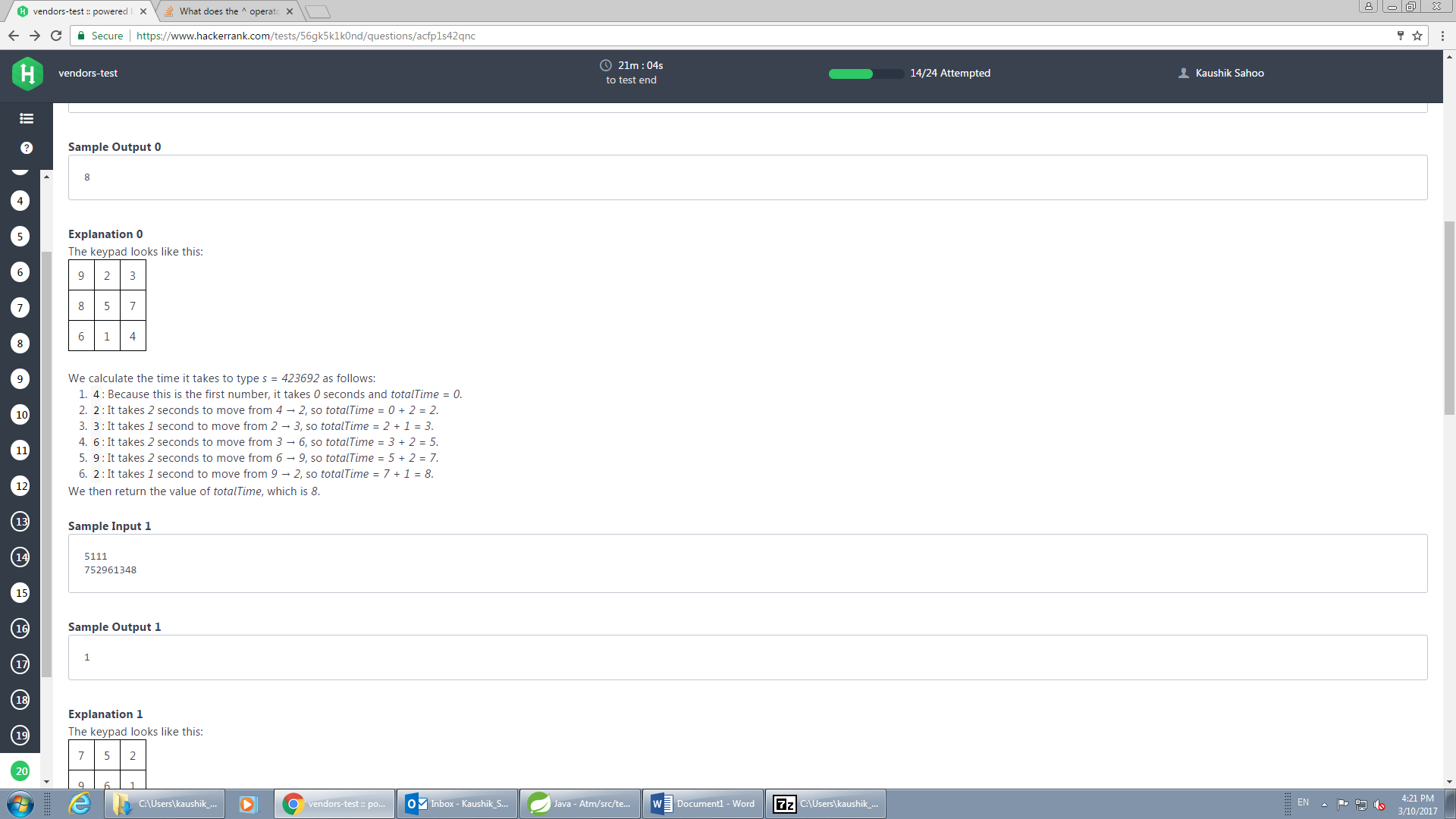


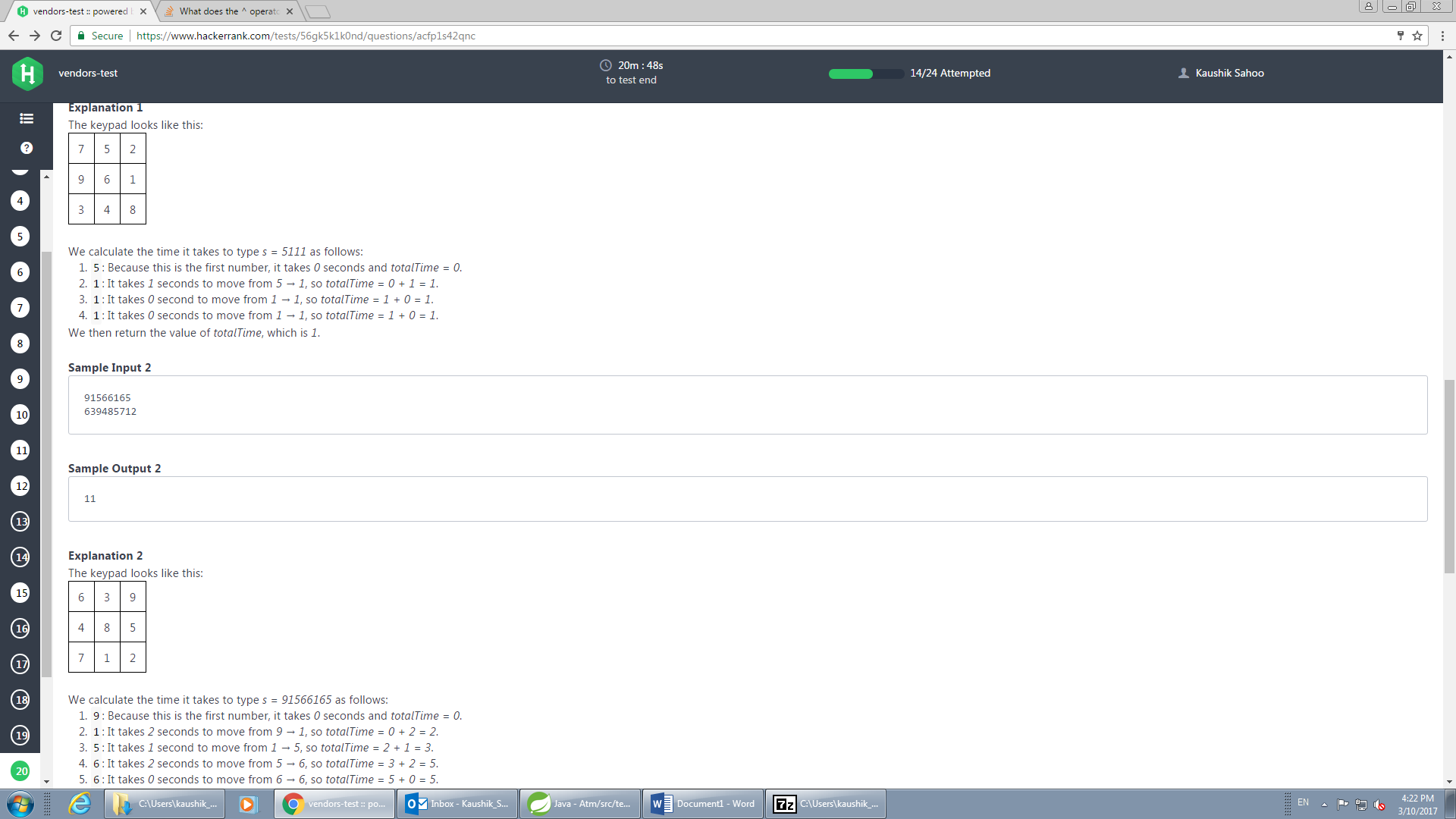


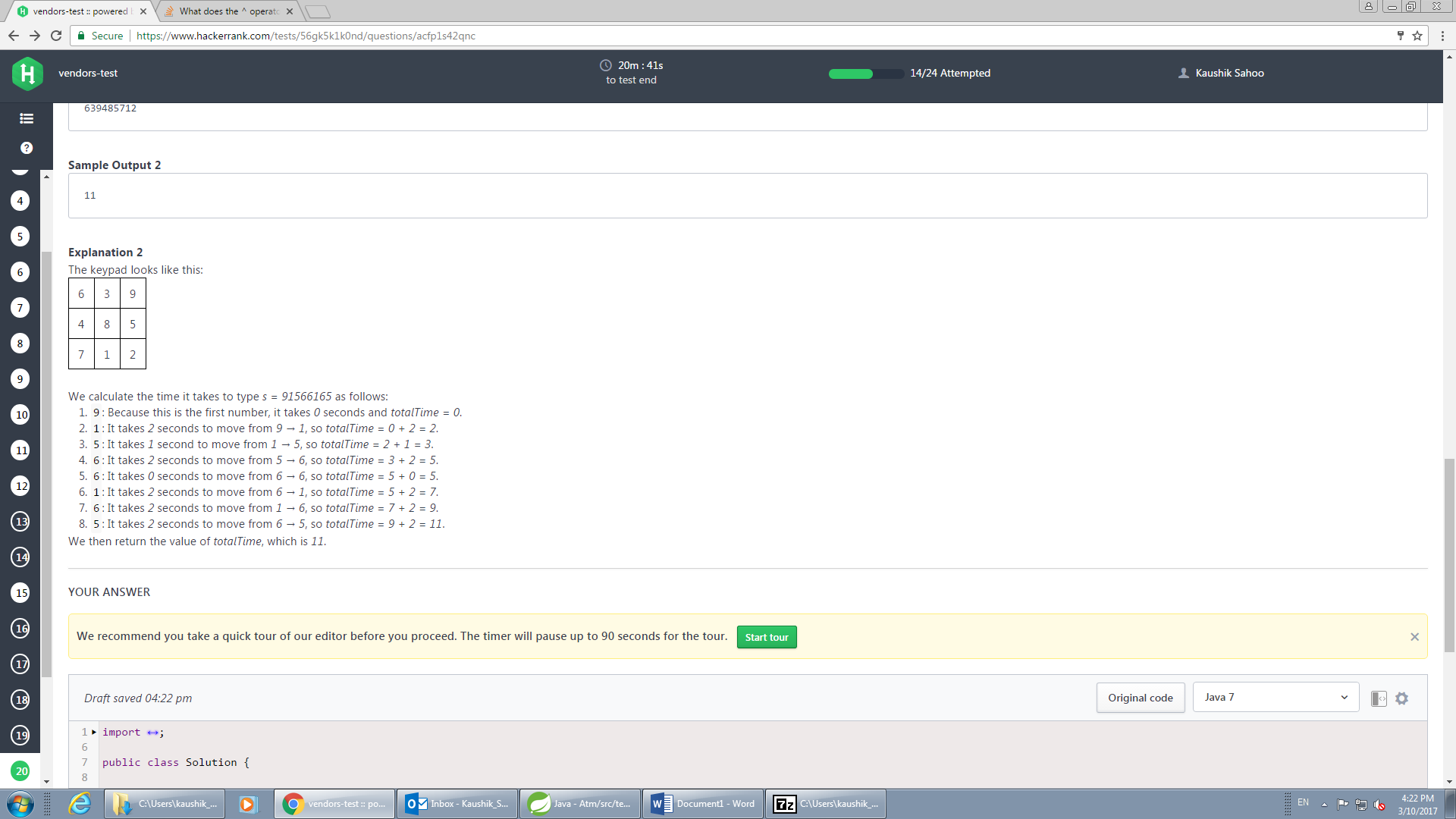


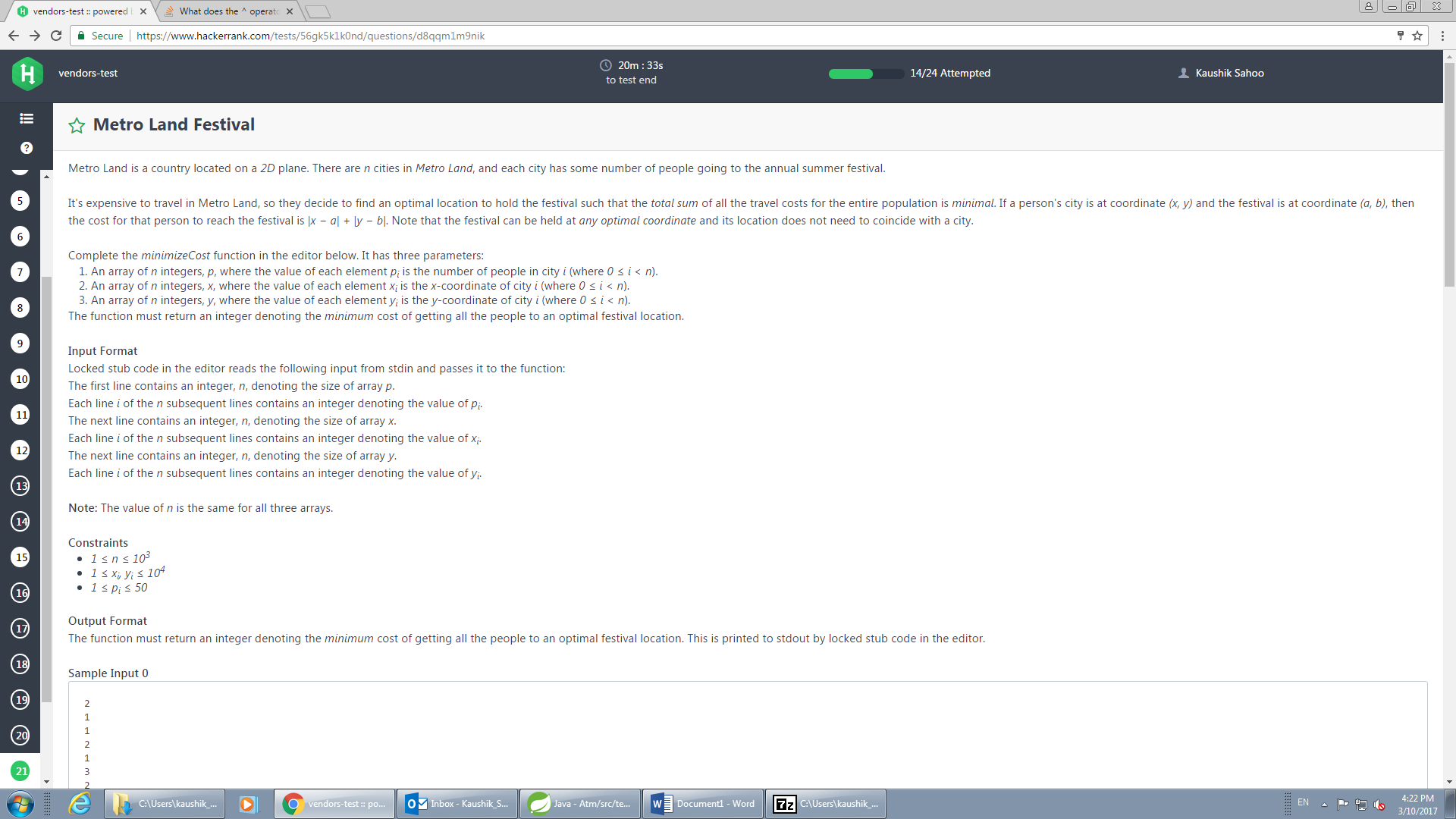


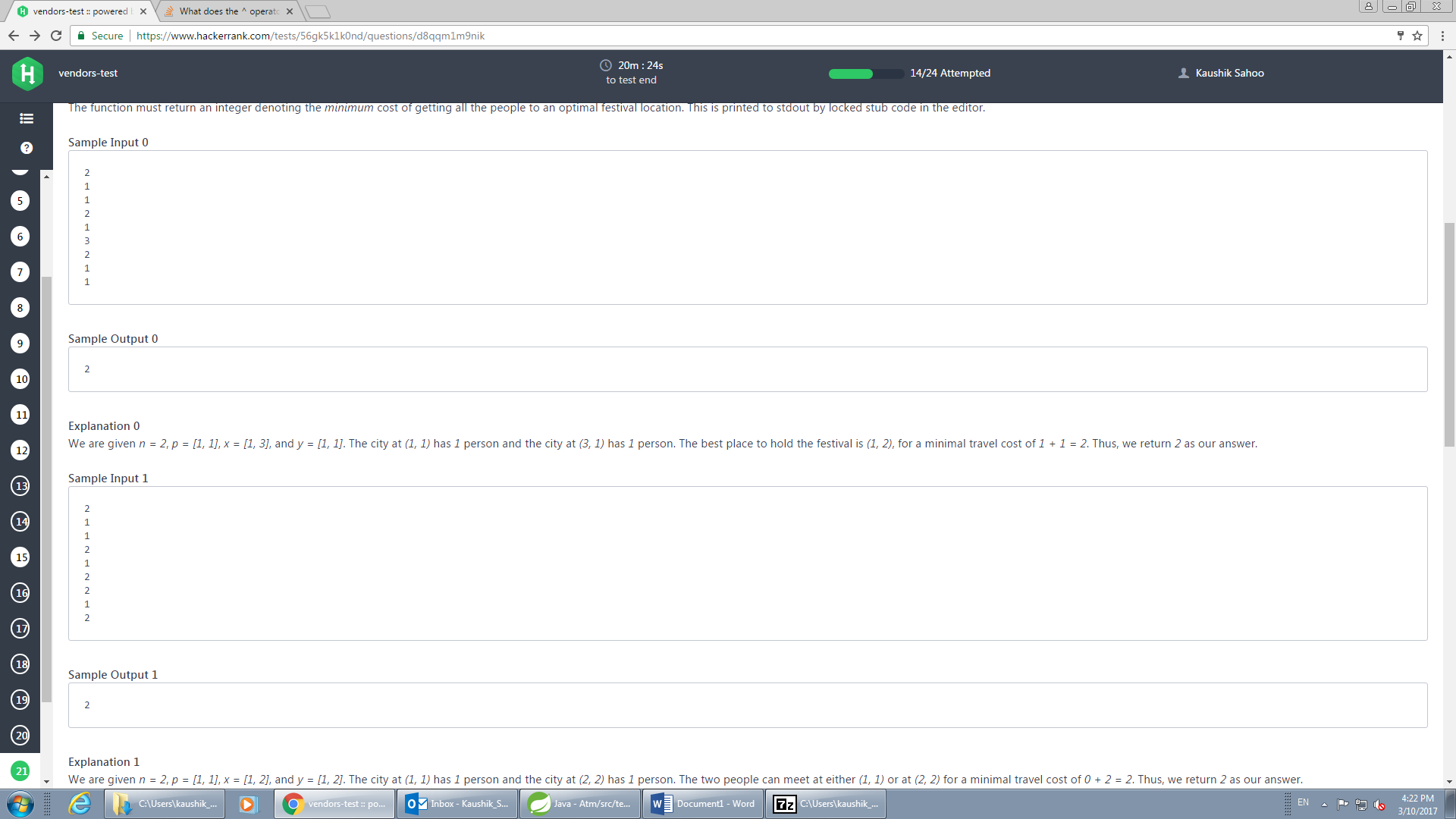


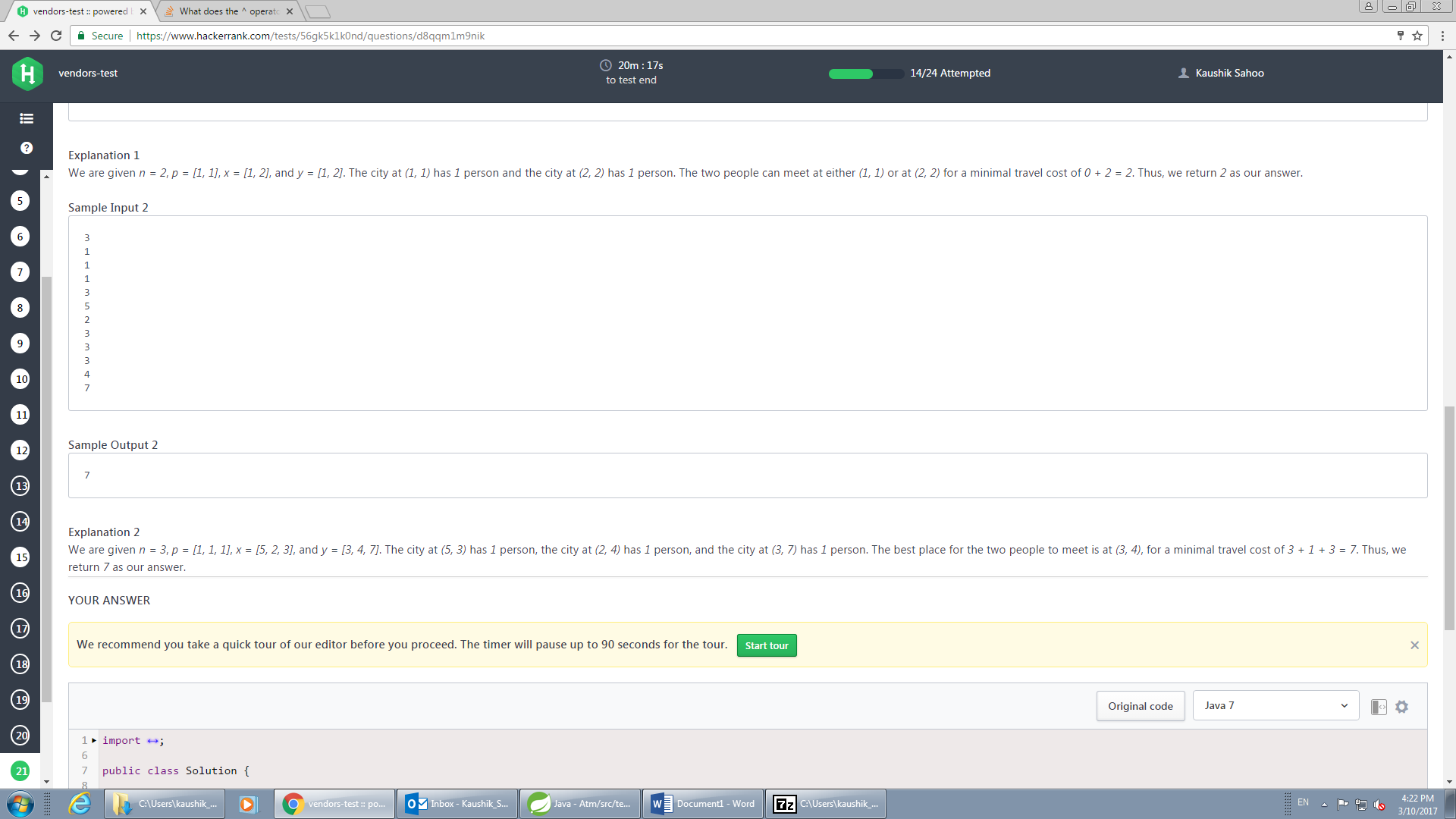






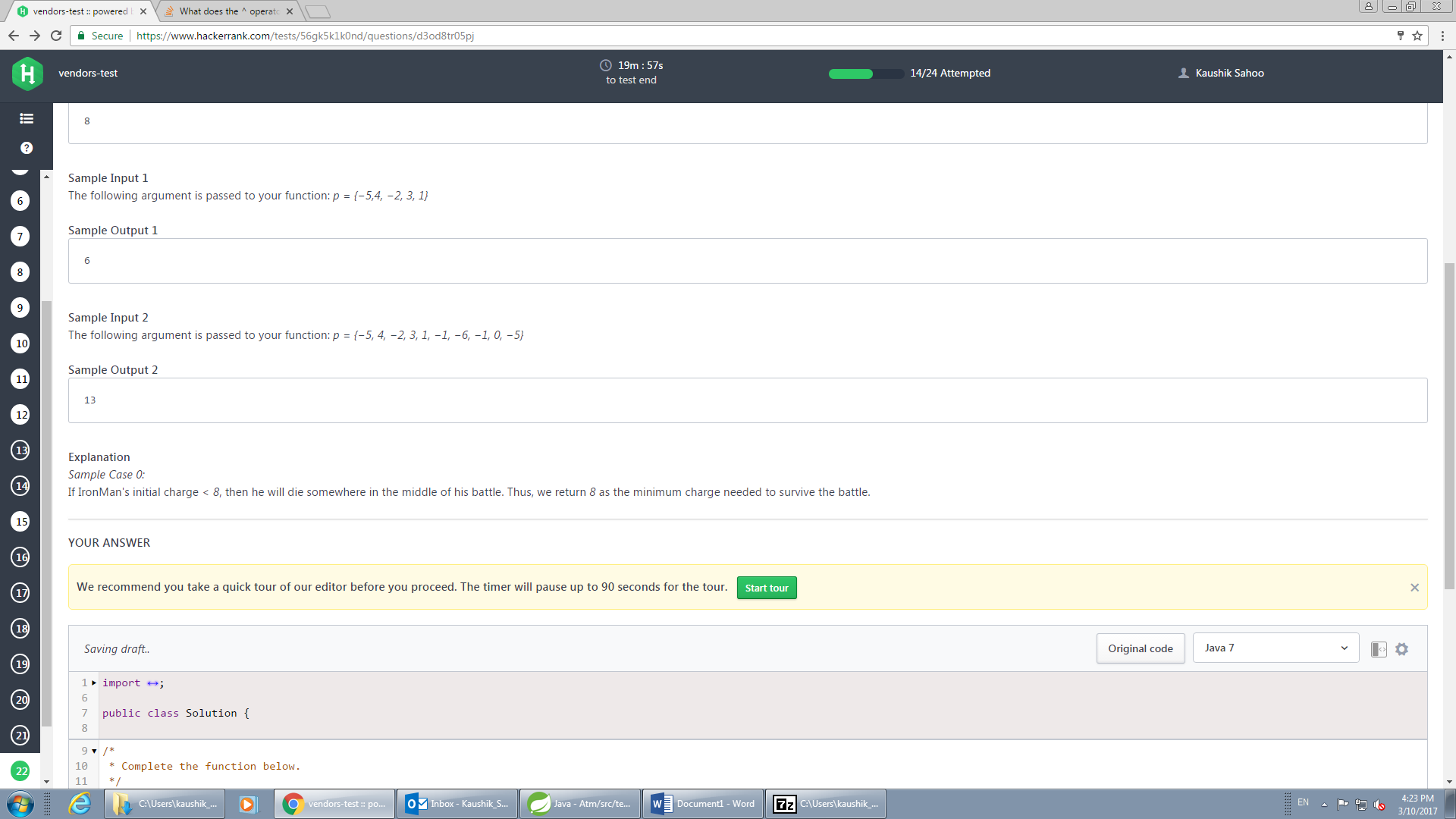






\





1)      Given a number n, how many ways can you arrive at the sum=n only by adding the consecutive numbers,

For instance if given n=15, you can arrive at 15 in following ways:

1+2+3+4+5 =15

4+5+6 =15

7+8 = 15

2)      Given a number\_Of\_Permitted\_Replacements & a string containing only ‘<’ & ‘>’ it is balanced if it’s of the form <<<>>> or <><> or <>

Task is to determine whether or not the given string is balanced.

If it’s unbalanced, can it be converted in to balanced string  by replacing ‘<’ or ‘> ’ with ‘<>’ up to the given number of replacements.

For instance if given string <>>  and number of permitted replacements =2

<>> can be made into  a balanced string by replacing ‘>’ with <> so only one replacement, which is lesser than or equal to the permitted replacements.

3) find the sum of Divisors of a given number,

Given 2 & 4 as input, 2 is divisible by 1  & 2. So the answer would be 1+2 = 3

For 4: divisors would be 1,2 & 4, so the result would be 1+2+4 = 7