**STEPANOV’S BINARY COUNTER**

Implementing Binary counter to find the second biggest in an array of integers.

The number of elements may or may not be an exact power of 2.

The algorithm needs at most n + logn - 2 comparisons.(However when n is odd the number of comparisons will be little more)

It uses extra space proportional to (logn) ^ 2.

**Functions implemented:**

**void find\_second\_greatest(int \*numbers, int length);**

The function gives the output in the given format (check screenshots).

The initial lines represent a match being played between the numbers on the LHS and the result of that match if on the RHS of the = sign. So, each line except the last two lines represent the match played and it’s result. The last line is the second greatest integer.

Output set will first have the highest of all elements of the two sets.

This will be followed by the highest number in the losing set.

Finally, the numbers that have lost to the highest number from the winner set are printed.

Consider:

5 20 15 x 18 10 2 = 25 18 20 15

Here, the two competing sets are 25 20 15 and 18 10 2

The biggest number is 25, and hence is printed first, post the '=' sign.

Hence 25 20 15 is the winning set, and 18 10 2 is the losing set.

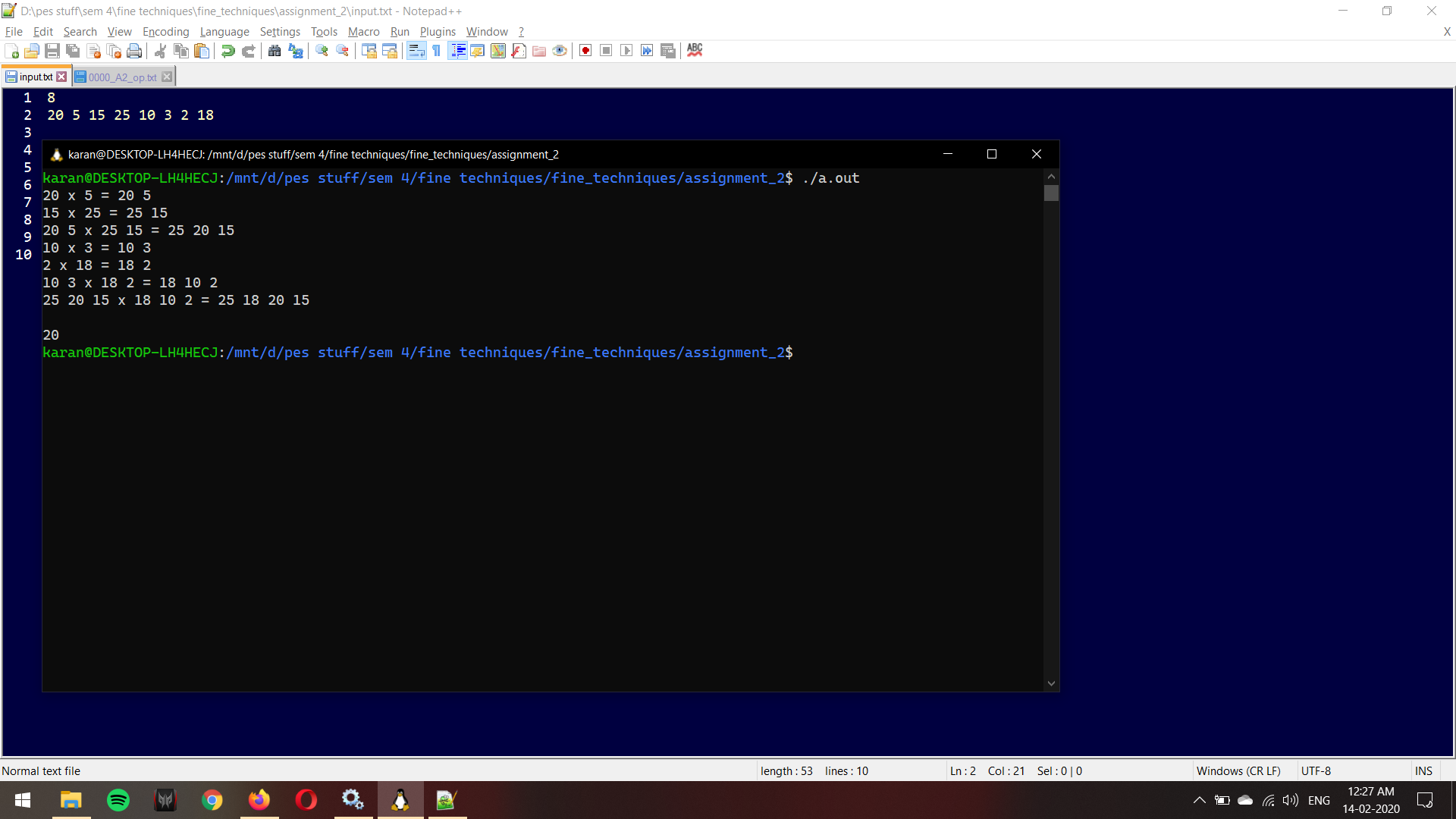
The biggest number of the losing set is 18, and hence is printed as the second number.

This is followed by all the numbers that have lost to 25(ie, 20 15), in the order as that in the winning set itself.

## Example:

numbers: [20, 5, 15, 25, 10, 3, 2, 18]

length: 8



Non power of 2 example:

numbers: [2, 3, 1, 4, 5, 0, 7]

length: 7

