Input/Output File Description: -

The input mainly consists of 1 File: [inputPS1.txt]

- inputPS1.txt : - It consists of the initial input given by the user at the start of the program. Where each line contains an array of integers separated via space.

The output consists of a single File: [outputPS1.txt]

- outputPS1.txt: - This file consists of output generated from our python code which tells us whether there is maxima or minima present. Or if the values are strictly in increasing or decreasing order.

Sample Input ***inputPS1.txt:***

3 5 7 9 11 13 15 17

17 15 13 11 9 7 5 3

3 5 7 9 8 7 6 5

17 15 13 10 7 12 13 14 15

Sample Output ***outputPS1.txt***

increasing 3

decreasing 3

maxima 9

minima 7

Actual Input ***inputPS1.txt*** *:*

2 6 8 10 11 14 16 19

19 16 12 10 8 5 3 1

3 5 8 11 7 6 4 1

17 14 12 10 5 12 13 14 15

Sample Output ***outputPS1.txt***

increasing 2

decreasing 1

maxima 11

minima 5

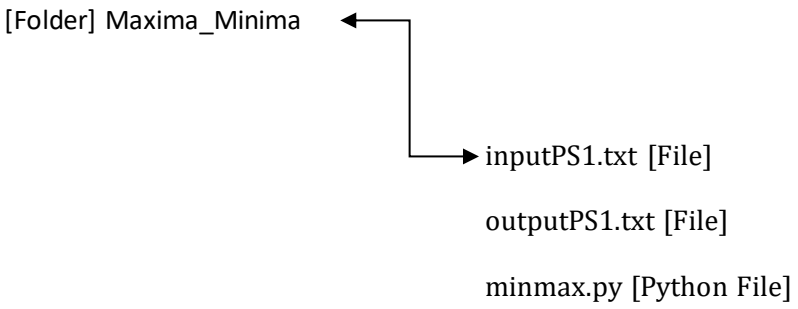
Program Description :-

In this program we are trying to find the maxima or minima from the array if integer provided. We are calculating its value and displaying them in the output file. We are using Divide and Conquer strategy to solve this problem.

Function Name	Description
getminmax	This function takes four parameters [start, end, min, max] and it breaks down the input array into two half and returns the overall minimum and maximum value present and their indexes from the input array after comparing the min and max value obtained from each half of the array.
getoutput	This function displays the corresponding maxima or minima for the input array provided

Folder Structure :-

We should have the following files in a folder before executing the program.



Complexity Analysis:-

For counting the number of comparisons, since this is a recursive function, let us define the recurrence relation:

$T(n) = 2\,T(n/2) + 2$

$T(2) = 1$

$T(1) = 0$

We can solve this recurrence relation by master method/recursion tree method.

if n is a power of 2

$T(n) = 3n/2 - 2$

Time complexity = O(n) (For recursion call stack)
Space complexity = O(logn) (For recursion call stack)

If n is a power of 2, the algorithm needs exactly $((3n/2)-2)$ comparisons to find min and max. If it's not a power of 2, it will take a few more