# Documentation: Min-Max Algorithm (Python)

## 1. Purpose

This program uses the Min-Max algorithm, a method used to find the best value or decision when there are two players — one trying to get the maximum value (Max) and the other trying to get the minimum value (Min). It’s often used in games or decision-making problems.

## 2. What the Program Does

- Takes a list of numbers (called 'values') that represent the final possible results.  
- Works its way back through possible choices, switching between Min and Max.  
- In the end, gives the best possible value assuming both players play perfectly.

## 3. Libraries Used

import math

Used to calculate the number of levels in the data using logarithm base 2.

## 4. Code Breakdown

Step 1 – Data Setup

values = [3, 4, 3, 5, 6, 7, 1, 9]  
levels = int(math.log2(len(values)))

• 'values' → list of numbers to test.  
• 'levels' → tells how deep the algorithm should go (based on how many values there are).  
Note: The list length should be a power of 2 (like 4, 8, 16, ...).

Step 2 – The Function

|  |  |
| --- | --- |
| Parameter | Meaning |
| currdepth | The current level of recursion (starts from 0) |
| nodeindex | The current index we are checking |
| minturn | True if it’s Min’s turn, False if it’s Max’s |
| values | The list of numbers |
| totaldepth | The total number of levels to go down |

Step 3 – Base Condition

if currdepth == totaldepth:  
 return values[nodeindex]

If we’ve reached the end, return the value at that index.

Step 4 – Recursive Calls

If it’s Min’s turn, choose the smaller of the two next results.  
If it’s Max’s turn, choose the larger of the two.

Step 5 – Function Call

print(min\_max(0, 0, True, values, levels))  
Starts the algorithm and prints the result.

## 5. Output Example

For this list:  
values = [3, 4, 3, 5, 6, 7, 1, 9]  
Output:  
3

## 6. Notes

- Make sure your list length is a power of 2 (like 4, 8, 16, etc.).  
- If not, you can pad it with extra numbers (like 0s) to make it work.  
- You can switch the first 'minturn' from True to False to see the opposite player’s result.