

Counting Peaks with numPy

If we take data from this list, $y = [1, 4, 5.1, 5.2, 5.3, 6, 9, 6.1, 6.2, 7.1, 7.2, 7.3, 8.4, 8.5, 9.2, 9.3, 20, 6, 8, 11.2, 11.5]$, and plot it into a graph, then there would be two “peaks” as shown in the graph below. A peak is when the data point is greater than both the value to the left and the value to the right. Write a command in the function `peak_indexes` of the program below that accepts a list of numbers and displays all peaks.

```
import numpy as np
def peak_indexes(x):
    # x is an array containing values
    # return an array listing indexes of "peaks"
    ???

def main():
    d = np.array([float(e) for e in input().split()])
    pos = peak_indexes(np.array(d))
    if len(pos) > 0:
        print(", ".join([str(e) for e in pos]))
    else:
        print("No peaks")

exec(input().strip()) # Don't remove this line
```

Try to write with NumPy without using loops

Input

A list of real numbers, separated by spaces.

Output

The list of all peak indexes, separated by a comma and a space. If there are no peaks, display “No peaks”.

Example

Input (from keyboard)	Output (on screen)
<code>print(peak_indexes(np.array([1,2,3,4])))</code>	<code>[]</code>
<code>main()</code> <code>1 2 3 4</code>	No peaks
<code>main()</code> <code>1 9 1 9 1 9 1 9 1</code>	<code>1, 3, 5, 7, 9</code>