Recursion

1. Write a recursive function to determine whether all digits of the number are odd or not.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4211133 | false |
| 7791 | true |
| 5 | true |

1. Given an array of numbers. Write a recursive function to find its minimal positive element. (if such element does not exist, return -1)․

|  |  |
| --- | --- |
| **Input** | **Output** |
| [56, -9, 87, -23, 0, -105, 55, 1] | 0 |
| [45, -9, 15, 5, -78] | 5 |
| [-5, -9, -111, -1000, -7] | -1 |

1. Given an array of numbers which is almost sorted in ascending order. Find the index where sorting order is violated.

|  |  |
| --- | --- |
| **Input** | **Output** |
| [2, 12, 15, 48, 64] | -1 |
| [-9, -4, -4, 3, 12, 4, 5] | 5 |

1. Given an array. Write a recursive function that removes the first element and returns the given array. (without using *arr.unshift()*)

|  |  |
| --- | --- |
| **Input** | **Output** |
| [6, 78, ‘n’, 0, 1] | [78, ‘n’, 0, 1] |
| [5] | [] |
| [] | [] |

1. Given an array of nested arrays. Write a recursive function that flattens it. (Hint create function that concats arrays).

|  |  |
| --- | --- |
| **Input** | **Output** |
| [1, [3, 4, [1, 2]], 10] | [1, 3, 4, 1, 2, 10] |
| [14, [1, [[[3, []]], 1], 0] | [14, 1, 3, 1, 0] |

1. Given an array and a number N. Write a recursive function that rotates an array N places to the left. (*Hint*: to add element to the beginning use *arr.unshift()*)

|  |  |
| --- | --- |
| [‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’] 3 | [‘d’, ‘e’, ‘f’, ‘g’, ‘h’, ‘a’, ‘b’, ‘c’] |
| [‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’] -2 | [‘g’, ‘h’, ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’] |

1. Given a number. Write a function that calculates its sum of the digits and if that sum has more than 1 digit find the sum of digits of that number. Repeat that process if needed and return the result.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 14 | 5 |
| 29 | 2 |
| 999999999999 | 9 |