

50 coding challenges – Part I

1. Print numbers from 1 to 10
2. Print the odd numbers less than 100
3. Print the multiplication table with 7
4. Print all the multiplication tables with numbers from 1 to 10
5. Calculate the sum of numbers from 1 to 10
6. Calculate 10!
7. Calculate the sum of even numbers greater than 10 and less than 30
8. Create a function that will convert from Celsius to Fahrenheit
9. Create a function that will convert from Fahrenheit to Celsius
10. Calculate the sum of numbers in an array of numbers
11. Calculate the average of the numbers in an array of numbers
12. Create a function that receives an array of numbers as argument and returns an array containing only the positive numbers
13. Find the maximum number in an array of numbers
14. Print the first 10 Fibonacci numbers without recursion
15. Create a function that will find the n^{th} Fibonacci number using recursion
16. Create a function that will return a Boolean specifying if a number is prime
17. Calculate the sum of digits of a positive integer number
18. Print the first 100 prime numbers
19. Create a function that will return in an array the first “p” prime numbers greater than “n”
20. Rotate an array to the left 1 position
21. Rotate an array to the right 1 position
22. Reverse an array
23. Reverse a string
24. Create a function that will merge two arrays and return the result as a new array
25. Create a function that will receive two arrays of numbers as arguments and return an array composed of all the numbers that are either in the first array or second array but not in both
26. Create a function that will receive two arrays and will return an array with elements that are in the first array but not in the second



50 coding challenges – Part II

27. Create a function that will receive an array of numbers as argument and will return a new array with distinct elements
28. Calculate the sum of first 100 prime numbers and return them in an array
29. Print the distance between the first 100 prime numbers
30. Create a function that will add two positive numbers of indefinite size. The numbers are received as strings and the result should be also provided as string.
31. Create a function that will return the number of words in a text
32. Create a function that will capitalize the first letter of each word in a text
33. Calculate the sum of numbers received in a comma delimited string
34. Create a function that returns an array with words inside a text.
35. Create a function to convert a CSV text to a “bi-dimensional” array
36. Create a function that converts a string to an array of characters
37. Create a function that will convert a string in an array containing the ASCII codes of each character
38. Create a function that will convert an array containing ASCII codes in a string
39. Implement the Caesar cypher
40. Implement the bubble sort algorithm for an array of numbers
41. Create a function to calculate the distance between two points defined by their x, y coordinates
42. Create a function that will return a Boolean value indicating if two circles defined by center coordinates and radius are intersecting
43. Create a function that will receive a bi-dimensional array as argument and a number and will extract as a unidimensional array the column specified by the number
44. Create a function that will convert a string containing a binary number into a number
45. Create a function to calculate the sum of all the numbers in a jagged array (contains numbers or other arrays of numbers on an unlimited number of levels)
46. Find the maximum number in a jagged array of numbers or array of numbers
47. Deep copy a jagged array with numbers or other arrays in a new array
48. Create a function to return the longest word in a string
49. Shuffle an array of strings
50. Create a function that will receive n as argument and return an array of n random numbers from 1 to n. The numbers should be unique inside the array.
51. Find the frequency of letters inside a string. Return the result as an array of arrays. Each subarray has 2 elements: letter and number of occurrences.
52. Calculate Fibonacci(500) with high precision (all digits)
53. Calculate 70! with high precision (all digits)

