

Roman Numeral Decoder

Create a function that takes a Roman numeral as its argument and returns its value as a numeric decimal integer. You don't need to validate the form of the Roman numeral.

Modern Roman numerals are written by expressing each decimal digit of the number to be encoded separately, starting with the leftmost digit and skipping any 0s. So 1990 is rendered "MCMXC" (1000 = M, 900 = CM, 90 = XC) and 2008 is rendered "MMVIII" (2000 = MM, 8 = VIII). The Roman numeral for 1666, "MDCLXVI", uses each letter in descending order.

Example:

```
solution('XXI'); // should return 21
```

My solution :

```
const lookup : any = {
  "M":1000,
  "D":500,
  "C":100,
  "L":50,
  "X":10,
  "V":5,
  "I":1
};
export function solution(roman: string): number {
  let final : number = 0
  roman.split("").map((x, i) => lookup[x] < lookup[roman[i+1]] || 0 ? final -= lookup[x] : final += lookup[x])
  return final
}
```

Mine was the best practice