Pasos iniciales

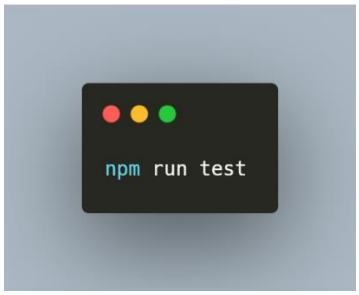
Para levantar la solución, ejecutar lo siguientes pasos a través de línea de comandos:

1. Descargar código fuente:

2. Instalar dependencias.

3. Ejecutar tests unitarios.

   Sin cobertura de código.

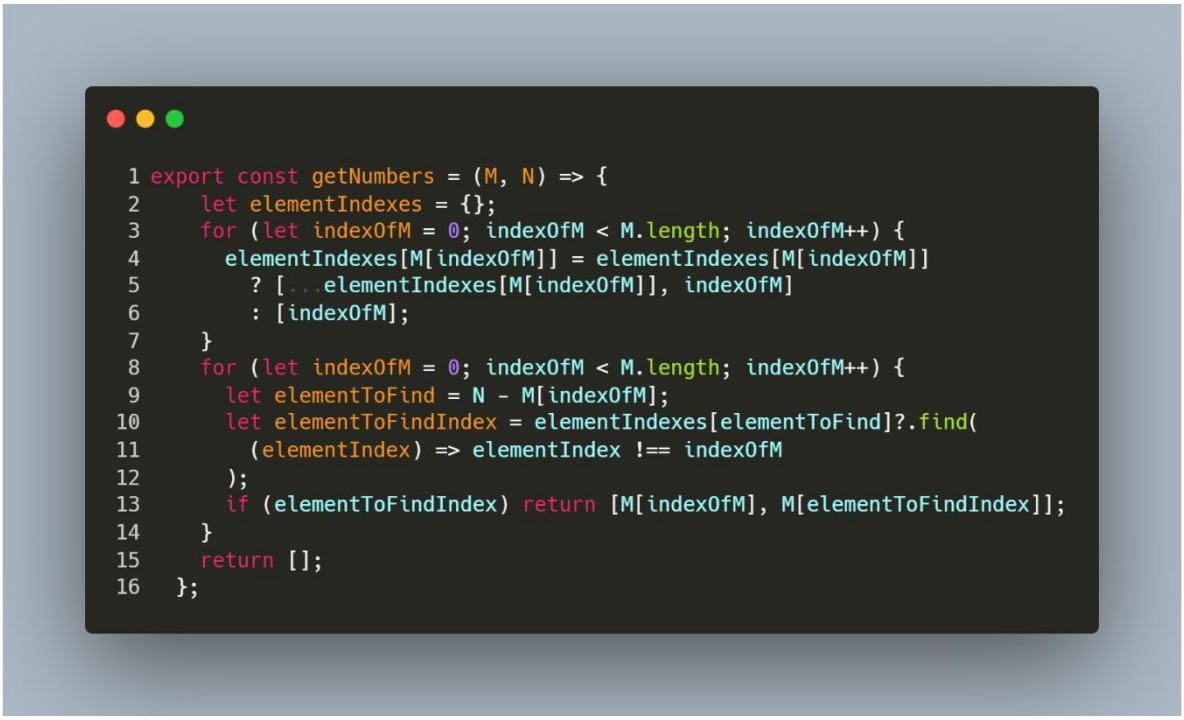


   Con cobertura de código:

Supuestos.

> Se asume que dentro del arreglo M de entrada de la función pueden repetirse los elementos.
> En caso de que ningún par de elementos del arreglo se cumpla que su suma sea igual a N, la función retorna un arreglo vacío.

Descripción de la solución.

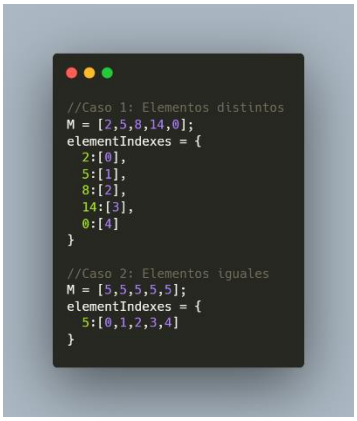


```
1 export const getNumbers = (M, N) => {
2     let elementIndexes = {};
3     for (let indexOfM = 0; indexOfM < M.length; indexOfM++) {
4       elementIndexes[M[indexOfM]] = elementIndexes[M[indexOfM]]
5         ? [...elementIndexes[M[indexOfM]], indexOfM]
6         : [indexOfM];
7     }
8     for (let indexOfM = 0; indexOfM < M.length; indexOfM++) {
9       let elementToFind = N - M[indexOfM];
10      let elementToFindIndex = elementIndexes[elementToFind]?.find(
11        (elementIndex) => elementIndex !== indexOfM
12      );
13      if (elementToFindIndex) return [M[indexOfM], M[elementToFindIndex]];
14    }
15    return [];
16  };
```

1. Línea 1: Se declara el método **getNumbers**, cuyos parámetros son **M** = el arreglo a procesar, **N** = la suma a encontrar.

2. Líneas 2 al 7: Se declara diccionario **elementIndexes**, cuyas claves corresponden a los elementos no repetidos del arreglo **M**, y sus valores son los índices del arreglo **M** donde se encuentran los elementos. El objetivo del ciclo **for** es poblar el diccionario **elementIndexes.**

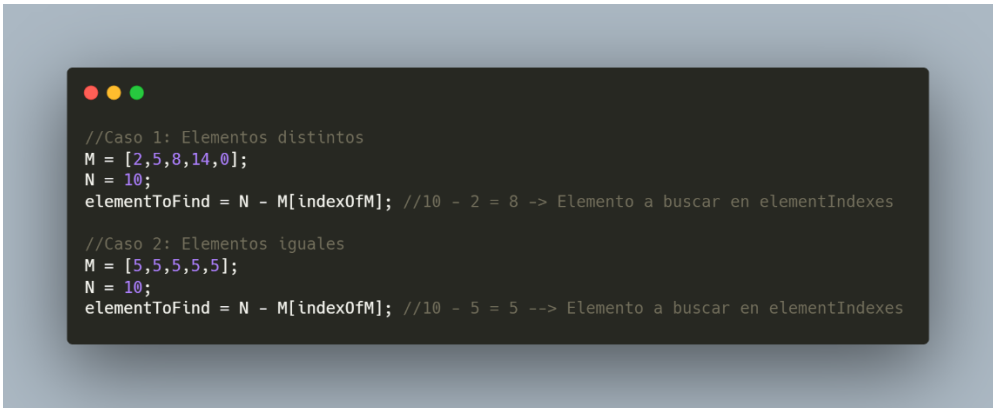
   Ejemplos:



```
//Caso 1: Elementos distintos
M = [2,5,8,14,0];
elementIndexes = {
  2:[0],
  5:[1],
  8:[2],
  14:[3],
  0:[4]
}

//Caso 2: Elementos iguales
M = [5,5,5,5,5];
elementIndexes = {
  5:[0,1,2,3,4]
}
```

3. Líneas 8 al 14: Se itera nuevamente el arreglo **M**, con el fin de encontrar los dos elementos que cumplan que su suma sea igual a **N**.
   > Línea 9: La variable **elementToFind**, corresponde a la diferencia entre N y el elemento actual del arreglo **M**.



```
//Caso 1: Elementos distintos
M = [2,5,8,14,0];
N = 10;
elementToFind = N - M[indexOfM]; //10 - 2 = 8 -> Elemento a buscar en elementIndexes

//Caso 2: Elementos iguales
M = [5,5,5,5,5];
N = 10;
elementToFind = N - M[indexOfM]; //10 - 5 = 5 --> Elemento a buscar en elementIndexes
```

   > Línea 10: La variable **elementToFindIndex** corresponde al índice donde se encuentra **elementToFind**. **elementToFindIndex** debe ser distinto a **indexOfM** (el índice con el que se recorre el arreglo **M**);
   > Línea 13: Si **elementToFindIndex** tiene valor, entonces se retornan los elementos de **M** en las posiciones **indexOfM** y **elementToFindIndex** y finaliza la ejecución de la función **getNumbers**
   > Línea 15: Si la ejecución de **getNumbers** llega hasta aquí, quiere decir que ningún par de elementos de **M** sumados son iguales a **N,** por lo tanto, se retorna un arreglo sin elementos (**[]**).