

## Programming Coding Project 1

**Name:** Yusupov Boburjon

**Neptun code:** YTAJDI

Specification: <https://tinyurl.com/ym992d2d>

Structogram: <https://tinyurl.com/5n6vx54r>

### Task

We have N cubes in a building game. Create a program that:

- Gives the total volume of an object that can be built by using all cubes.
- Gives the index of the smallest cube.

### Specification

**In:**  $n \in \mathbb{N}$ ,  $\text{cubes} \in \mathbb{N}[1..n]$

**Out:**  $\text{volume} \in \mathbb{N}$ ,  $\text{min} \in \mathbb{Z}$

**Pre:**  $1 \leq n \leq 100$  and  $\forall i \in [1..n]: (1 \leq \text{cubes}[i] \leq 100)$

**Post:**  $\text{volume} = \text{SUM}(i=1..n, \text{cubes}[i] * \text{cubes}[i] * \text{cubes}[i])$   
and  $(\text{min},) = \text{MIN}(i=1..n, \text{cubes}[i])$

### Algorithm

n	~	n
volume	~	volume
min	~	min
A(i, min)	~	$\text{cubes}[\text{min}] > \text{cubes}[i]$
b..e	~	$i < n$ ( $i = i + 1$ in every iteration)

In: n, cubes[]	
volume:=0, min:=0	
i<n	
cube:=cubes[i]*cubes[i]*cubes[i]	
volume:=volume+cube	
$\text{cubes}[\text{min}] > \text{cubes}[i]$	
min:=i	-
i:=i+1	
Out: volume, min	

In: n, cubes[]	
volume:=0, min:=0	
i=b..e	
cube:=cubes[i]*cubes[i]*cubes[i]	
volume:=volume+cube	
A(i, min)	
min:=i	-
Out: volume, min	

## Test cases

1. Invalid input: n: 2, [-4, 2] -> invalid...
2. Invalid array length: n: 3, [3, 18, 4, 2, 9] -> invalid...
3. Correct input and output: n: 4, [3, 5, 5, 9] -> volume=1006, min=0
4. Minimum cube value at the end: n: 3, [9, 7, 1] -> volume=1072, min=2
5. Minimum cube at the beginning: n: 3, [1, 4, 2] -> volume = 73, min=0