

# Map and Lambda Function

Let's learn some new Python concepts! You have to generate a list of the first  $N$  fibonacci numbers,  $0$  being the first number. Then, apply the *map* function and a *lambda* expression to cube each fibonacci number and print the list.

## Concept

The `map()` function applies a function to every member of an iterable and returns the result. It takes two parameters: first, the function that is to be applied and secondly, the iterables.

Let's say you are given a list of names, and you have to print a list that contains the length of each name.

```
>> print (list(map(len, ['Tina', 'Raj', 'Tom'])))  
[4, 3, 3]
```

*Lambda* is a single expression anonymous function often used as an inline function. In simple words, it is a function that has only one line in its body. It proves very handy in functional and GUI programming.

```
>> sum = lambda a, b, c: a + b + c  
>> sum(1, 2, 3)  
6
```

## Note:

*Lambda* functions cannot use the return statement and can only have a single expression. Unlike *def*, which creates a function and assigns it a name, *lambda* creates a function and returns the function itself. Lambda can be used inside lists and dictionaries.

## Input Format

One line of input: an integer  $N$ .

## Constraints

$$0 \leq N \leq 15$$

## Output Format

A list on a single line containing the cubes of the first  $N$  fibonacci numbers.

## Sample Input

```
5
```

## Sample Output

[0, 1, 1, 8, 27]

### Explanation

The first **5** fibonacci numbers are **[0, 1, 1, 2, 3]**, and their cubes are **[0, 1, 1, 8, 27]**.