

# Functions

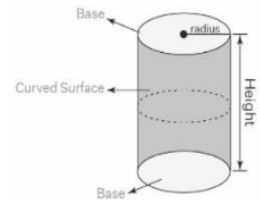
## Exercise 01

Write a program that **defines** and uses:

1. a function, named **FuncA**, which simply displays "Hello world!" (it has no parameters or return value),
2. a function, named **FuncB**, which displays "Hello world!" a number of times equal to the parameter received and does not return any value,
3. a function, named **FuncC**, that does the same thing as **FuncB** and returns 1 if it displayed the message more than 10 times, 0 otherwise.

## Exercise 02

1. Implement a function called **CalculateArea** to determine the surface area of a cylinder. The formula for the area involves multiplying  $2\pi$  by the radius of the base and the height. Develop a function named **CalculateVolume** for computing the volume of a cylinder. The volume is calculated by multiplying  $\pi$  by the square of the base radius and the height.
2. Create a program that prompts the user to input the base radius  $R$  and the height  $H$  of a cylinder. The program should then output the calculated area and volume using the functions defined earlier.



## Exercise 03

We want to write a function **ReviewMultiplication** that asks a user to recite the multiplication table. The user starts by entering a number between 2 and 9 (if the number is incorrect, the function asks for an integer again). Then the function displays one by one the lines of the multiplication table of this number, leaving the result blank and waiting for the user to enter the result. If it is correct, move on to the next line; otherwise, display an error message giving the correct value and finish. If all answers are correct, display a congratulations message. An example of a possible execution is shown below (user inputs are displayed in *italics*).

```
Value of  $n$  : 12
Try again: the value must be between 2 and 9
Value of  $n$  : 6
1 x 6 = 6
2 x 6 = 12
3 x 6 = 21
Error! 3 x 6 = 18 and not 21
...
```

1. Write the function **ReviewMultiplication**.
2. Modify the function **ReviewMultiplication** so that it does not stop when a wrong answer is given, but returns the number of errors at the end.

3. Write a main function to test the function.