

# Hello World!

---

We are so enthusiastic about our Hello World's that we need to shout it.

Your task is to write a program that prints "Hello World" followed by a configurable number of exclamation marks.

## Input description

The input will be one line for each case, containing an integer number that represents how many exclamation marks should follow `Hello World`.

## Sample

### Input

```
1
5
```

### Output

```
Hello World!
Hello World!!!!
```

## Instructions

In the folder `cases` you will find 2 input files and 1 output file. The first input file is the *easy* one and you can use them to validate your parsing logic and output formatting against the provided output file. The second input is the *hard* one and this one does not have a provided output file (so you are on your own here!).

In this folder you will find a password-protected zip file containing the second challenge. To find out the password that will grant you access to it, you must find the solution of `input_02.txt` and calculate its *sha-1* digest. On Unix the command line tool `sha1sum` will give you the digest, on Windows you can use `certutil -hashfile "output_02.txt" SHA1`.

For reference here are the *sha-1* digest of the provided output.

```
6876e4148ca489f4945a967b575cb7a318187a95  output_01.txt
```

For example, if your solution executable is a Python script named `solve.py`, it reads from *stdin* and writes to *stdout*, in order to access the second problem you could do:

```
unzip -P $(python solve.py < cases/input_02.txt | shasum | cut -f1 -d ' '
') ../sample_success.zip
```

If you are one of those who use Windows, you could try this Powershell method instead to get the password:

```
(Get-Content cases/input_02.txt | python solve.py | Set-Content output);
(Get-Content output -raw | % {$ _ -replace "`r", ""} | Set-Content -
NoNewline clean_output); (Get-FileHash -Algorithm sha1 clean_output)
```

Make sure that the solutions generated by your code have matching hashes (if they look the same but the digest don't match, check for extra carriage return characters, trailing line, whitespace, etc.).

Once you have found the password that succesfully decrypts the zip file, please send an email to [code-challenge@aircall.io](mailto:code-challenge@aircall.io) with the subject **aircall-nova challenge sample 1**, the *sha-1* digest in the body and the source code for your solution as an attachment. This is simply for demonstration purposes, so you don't really need to do it if you don't want to 😊