

Challenge 17 - The Failsystem

« [Prev](#) [Next](#) »



We had a very important project at the ITSS (International Tuenti Space Station). We launched the TUENTIVOYAGER, a probe that would collect data about the origin of the universe.

A few years ago, we lost the communication with the old TUENTIVOYAGER space probe due to a malfunction in the radio equipment. We believe that the probe continued collecting valuable data until its power ran out, but we didn't have access to it.

However, a few months ago, space trash made the probe change its orbit and it crashed into Earth. Luckily, we managed to recover its hard disk and we were able to perform a complete dump.

However, there are some issues. It seems that the probe used a very old file system that hardly anyone is familiar with today, and the disk was significantly damaged by the crash and the high levels of space radiation it received. Despite this, we want to

extract all the reliable information possible.

We need your help to extract the valuable information from the disk dump.

You should not mount this filesystem since it may cause problems with your root filesystem, specially if you are using file indexers. There is no need to mount it to solve this

problem.

Input

You must download the contents of the hard disk: [TUENTIDISK](#)

The first line in the test input contains the number of files we want to extract. The next lines will contain an absolute file path for each file we want to extract.

All file names will be in uppercase.

Output

For each requested file, you must calculate its MD5 if the file is intact or the string "CORRUPT" if the file is damaged in any way.

Sample input

```
5
/TUENTI.4/HELLO.TXT
/TUENTI.4/OUTSIDE.JPG
/TUENTI.4/AROUND.TXT
/TUENTI.4/WARNING.TXT
/TUENTI.4/MORE/DATA.TXT
```

Sample output

```
7012acbb1d394b20567dffbf0992b677
CORRUPT
CORRUPT
76910e70524814cfe2138910ae47d66e
CORRUPT
```

Submit & test your code

To test and submit code we provide a set of tools to help you. Download [con-test tools](#) if you haven't already done that. You will then be able to test your

solution to this challenge with the challenge tokens.

```
challenge tokens: CHALLENGE_17, CHALLENGE_SUBMIT_17
```

To test your program

```
./test_challenge CHALLENGE_17 path/program
```

A nice output will tell you if your program got the right solution or not. You can try as many times as you need.

To test your program against the input provided in the submit phase

```
./test_challenge CHALLENGE_SUBMIT_17 path/program
```

During the submit phase, in some problems, we might give your program harder inputs. As with the test token, a nice output will tell you if your program got the right solution or not. You can try as many times as you need.

In the actual contest you first need to solve the test phase before submitting the code, you must provide the source code used to solve the challenge and you can only submit once (once your solution is submitted you won't be able to amend it to fix issues or make it faster).

If you have any doubts, please check the [info section](#).

« Prev Next »

Tweet about this! [#TuentiChallenge4](#)



Follow @Tuentieng