Challenge 17 - The Failsystem

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 contest tools if you haven't already done that. You will then be able to test and submit your solution to this challenge with the challenge token.

```
Challenge token: cjd_xJZOI9s6rlPLRbwS
```

To test your program

```
./test_challenge cjd_xJZOI9s6rlPLRbwS 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 submit your program to the challenge

```
./submit_challenge cjd_xJZOI9s6rlPLRbwS
path/source_pkg.tgz path/program
```

Note that you first need to solve the test phase before submitting the code. During the submit phase, in some problems, we might give your program harder questions, so try to make your program failsafe.

Important: In this phase, you must provide the source code used to solve the challenge and, if necessary, a brief explanation of how you solved it.

Remember **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, so please be sure your solution is finished before submitting it.

If you have any doubts, please check the info section.

Go ahead

I'm done! :)

Once you have submitted your code, hit refresh and continue to next challenge.

I'm stuck! :(

Be sure you follow the Tuenti Engineering twitter for updates and possible hints during the contest.

If this challenge is too hard and you are blocked, you will be able to skip it after two hours. Note that **you won't be able to complete it later**, and you have a limited number of challenges to skip.

Finally, if you run out of skips but are still really stuck with one problem, you will be able to skip it after 24 hours.

Challenge status:

Test case	Not done
Solution submitted	Not done
Skip	You can't skip this one

Refresh status

Tweet about this! #TuentiChallenge4 Share Follow @Tuentieng