

## تست جنریتور

سلیب که از طراحی سوالات ورودی و خروجی کوئرا خسته شده بود، تصمیم گرفت تا با پیاده‌سازی اسکریپتی این فرآیند طاقت‌فرسا را برای خود سهل کند! ابتدا با ساختار فایل داوری ورودی خروجی آشنا شویم، ساختار فایلی که برای کوئرا مورد پذیرش است، به فرم زیر است:

```
.
├── in
│   ├── input1.txt
│   ├── input2.txt
│   └── ...
└── out
    ├── output1.txt
    ├── output2.txt
    └── ...
```

کوئرا برای هر ورودی مثل *i*، محتویات فایل `in/input{i}.txt` را به کد ارسالی کاربر می‌دهد و خروجی را با محتویات فایل `out/output{i}.txt` مقایسه می‌کند و در صورت برابر بودن، نمره را به پاسخ ارسالی کاربر می‌دهد.

تا پیش از این سلیب به طور کاملاً دستی محتوای تمامی `output{i}.txt` ها را پر می‌کرد، حال از شما می‌خواهد تا اسکریپتی بنویسید که این کار را برای او انجام دهد. او می‌خواهد تا اسکریپت شما با دریافت تعداد ورودی، ورودی‌ها را از فایل مربوطه بخواند و به عنوان ورودی (*stdin*) به برنامه `main.py` دهد و خروجی برنامه را در فایل خروجی مربوط به تست کیس بنویسد. برای درک بهتر مثالی را پیش می‌بریم.


## مثال

برای مثالی با 3 تست کیس، اسکریپت `generator.sh` شما ابتدا در ساختار زیر قرار دارد:

```
├── generator.sh
├── in
│   └── input1.txt
```

```
|
| ├── input2.txt
| ├── input3.txt
| └── main.py
```

سلیب می‌خواهد که وقتی اسکریپت شما را مانند زیر اجرا کند:

 bash

```
1 | ./generator.sh 3
```

- خروجی مربوط به ورودی `in/input1.txt` درون فایل `out/output1.txt` ذخیره شود.
- خروجی مربوط به ورودی `in/input2.txt` درون فایل `out/output2.txt` ذخیره شود.
- خروجی مربوط به ورودی `in/input3.txt` درون فایل `out/output3.txt` ذخیره شود.

و در نهایت به ساختار زیر برسیم:

```
.
├── generator.sh
├── in
│   ├── input1.txt
│   ├── input2.txt
│   └── input3.txt
├── main.py
└── out
    ├── output1.txt
    ├── output2.txt
    └── output3.txt
```

## پروژه اولیه

برای دریافت پروژه اولیه [این لینک](#) را دانلود کنید. درون لینک ساختار فایلی زیر را مشاهده می‌کنید:

```
Test-Generator
├── generator.sh
└── in
```

```
| | input1.txt
| | input2.txt
| | input3.txt
| main.py
```

راه حل خود را درون فایل `generator.sh` پیاده سازی کنید.

## توجه کنید

- فایل `main.py` صرفاً برای تست روی سیستم شخصی‌تان در اختیار شما قرار گرفته و `main.py` روی سیستم داوری خروجی‌های متفاوتی تولید می‌کند.
- برای اجرای فایل `main.py` باید از دستور زیر استفاده کنید:

```
bash
```

```
1 | python3 main.py
```

بنابراین طبق دستور بالا باید نسخه ۳ پایتون را در سیستم شخصی‌تان نصب داشته باشید.

- در هر تست در سیستم داوری، تعدادی تست‌کیس در دایرکتوری `in/` قرار دارند و تضمین می‌شود که هیچ تست‌کیسی خالی نمی‌باشد.
- دسترسی اجرا به فایل ارسالی شما به طور خودکار در سیستم داوری داده می‌شود.

## نحوه ارسال

برای ارسال جواب، دستورات خواسته شده را در فایلی با نام `generator.sh` بنویسید و سپس آن را انتخاب و ارسال کنید.