



# *Little Professor*

*Problem set*





<https://youtu.be/ZuJwzH9Blgs>

یکی از اولین اسباب بازی های دیوید در کودکی که خیلی هم بامزه بود، یک little professor بود، یک ماشین حساب که برای دیوید 10 مسئله مختلف ریاضی ایجاد میکرد تا حل کند.

برای مثال اگر ماشین حساب مقدار  $4+0=$  را نشان میداد، دیوید می‌بایست 4 را به عنوان جواب وارد میکرد و اگر ماشین حساب مقدار  $4+1=$  را نشان میداد او باید عدد 5 را وارد میکرد.

اگر پاسخ او نادرست بود، ماشین حساب مقدار **EEE** را نمایش میداد و اگر مقدار درستی را وارد میکرد ماشین حساب آن عبارت ریاضی را بصورت کامل نشان میداد. ( برای مثال  $4 + 0 = 4$  یا  $4 + 1 = 5$  )

در فایلی به اسم `professor.py` برنامه ای را پیاده کنید که :

- از کاربر مقداری را به عنوان سطح بازی ( $n$ ) دریافت کند و اگر این مقدار بین 1، 2 و یا 3 نبود، دوباره از کاربر بخواهد تا این مقدار را وارد کند.
- به صورت تصادفی ده (10) مسئله ریاضی با فرمت  $X + Y =$  ایجاد کند که در آن هریک از  $X$  و  $Y$  یک عدد صحیح غیر منفی با  $n$  رقم باشد. نیازی به استفاده از عمل های اصلی دیگر به جز  $(+)$  نیست.
- از کاربر بخواهد هرکدام از مسئله ها را حل کند. اگر پاسخ صحیح نباشد (یا حتی اگر عدد نباشد)، خروجی برنامه باید `EEE` باشد و مجدد از کاربر درخواست جواب کند. البته کاربر فقط 3 بار باید فرصت داشته باشد تا جواب صحیح را وارد کند و اگر کاربر پس از 3 بار تلاش هنوز به پاسخ صحیح نرسید، برنامه باید جواب صحیح را به عنوان خروجی بازگرداند.
- در نهایت برنامه باید امتیاز کاربر را خروجی بدهد که برابر با تعداد سوال های جواب داده شده از 10 می باشد.

ساختار برنامه‌تون را به صورت زیر بنویسید، که در آن `get_level` از کاربر یک مقدار را به عنوان `level` (سطح) درخواست کند (در صورت نیاز مجدداً یک `level` را درخواست کند) و 1، 2 یا 3 را برمی‌گرداند.

سپس `generate_integer` یک عدد تصادفی صحیح غیر منفی را به تعداد ارقام `level` بازگرداند و اگر مقدار `level` بین 1، 2 و 3 نبود، `ValueError` می‌دهد.

```
import random

def main():
    ...

def get_level():
    ...

def generate_integer(level):
    ...

if __name__ == "__main__":
    main()
```

▼ نکته

- توجه کنید که میتوانید ارورهایی مانند `ValueError` را بصورت زیر `raise` کنید.

```
raise ValueError
```

- و همچنین ماژول `random` دارای چندین تابع می باشد که از طریق این لینک می توانید اطلاعات بیشتری نسبت به آنها داشته باشید.

<https://docs.python.org/3/library/random.html>

```
$ python professor.py
Level: 1
2 + 8 = 10
3 + 7 = 10
9 + 4 = 10
EEE
9 + 4 = 11
EEE
9 + 4 = 12
EEE
9 + 4 = 13
4 + 7 = 11
6 + 4 = 10
7 + 4 = 11
1 + 4 = cat
EEE
1 + 4 = dog
EEE
1 + 4 = bird
EEE
1 + 4 = 5
9 + 3 = 12
2 + 2 = 4
0 + 0 = 0
Score: 8
$
```

وارد [code.cs50.io](https://code.cs50.io) شوید، سپس بر روی پنجره‌ی Terminal کلیک کنید. توجه داشته باشید که دستور پنجره‌ی Terminal شما باید به صورت زیر باشد:

```
$
```

سپس کد زیر را اجرا کنید تا یک پوشه به نام professor در codespace ایجاد شود:

```
$ mkdir professor
```

سپس `cd professor` را اجرا کنید تا به پوشه‌ی `professor` منتقل شوید. اکنون شما باید آدرس `$ professor/` را در Terminal مشاهده کنید:

```
$ cd professor  
professor/ $
```

حالا می‌توانید دستور `code professor.py` را اجرا کنید تا فایل‌ی با نام `professor.py` ایجاد شود:

```
professor/ $ code professor.py
```

اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.



برای تست کد خود مراحل زیر را انجام دهید:

- برنامه خود را با `python professor.py` اجرا کرده و 1- را وارد کنید. برنامه باید دوباره از شما درخواست ورودی داشته باشد.

Level:

- برنامه خود را دوباره اجرا کرده و 4 را وارد کنید. برنامه باید دوباره از شما درخواست ورودی داشته باشد.

Level:

- دوباره برنامه را اجرا کرده و 1 را وارد کنید. برنامه باید شروع به طرح مسائل جمع با اعداد صحیح مثبت و تک رقمی کند.

$6 + 6 =$

- توجه کنید که برنامه شما باید 10 مسئله متمایز را تا قبل از چاپ کردن امتیاز، از شما بپرسد .

- برنامه خود را اجرا کرده و 1 را برای سطح وارد کنید. پاسخ را اشتباه وارد کنید. خروجی باید به صورت زیر باشد:

```
EEE
```

- برنامه خود را اجرا کرده و 1 را برای سطح وارد کنید. به اولین سوال 3 بار اشتباه پاسخ دهید. برنامه باید پاسخ صحیح را در خروجی نشان دهد:

```
6 + 6 = 12
```

- سپس سراغ سوال بعدی بروید و به بقیه سوال‌ها پاسخ صحیح بدهید. در انتها برنامه شما باید امتیاز 9 را در خروجی نمایش بدهد.
- برنامه خود را اجرا کرده و 1 را وارد کنید. به هر 10 سوال پاسخ صحیح دهید. برنامه شما باید امتیاز 10 را در خروجی بدهد.

❖ شما می‌توانید از آدرس زیر برای بررسی کردن کد خود استفاده کنید. CS50 از این برنامه برای آزمایش کد شما استفاده می‌کند. از این دستور استفاده کنید تا کدهایتان را امتحان کنید.

```
$ check50 cs50/problems/2022/python/professor
```

لبخند های سبز :) به این معنی هستند که برنامه‌ی شما در تست **قبول شده** و اخم های قرمز :) یعنی برنامه‌ی شما **دارای ایراد** هست، همچنین پیام های زرد ! بدین معنی است که برای اینکه **check50** بتواند این بخش برنامه شما را بررسی کند باید ابتدا ارورهایی که با اخم های قرمز :) جلوتر آمده اند را برطرف کنید.

. برای دیدن جزئیات بیشتر درباره این تست که توسط **check50** صورت گرفت، وارد لینکی که به شما خروجی داده است شوید.

با اجرا کردن صورت زیر در `Terminal` پاسخ خود را ارسال کنید.

```
$ submit50 cs50/problems/2022/python/professor
```

توجه کنید هرچند بار که لازم باشد میتوانید کد خود را اصلاح و ارسال کنید ، نمره آخرین پاسخ شما در نظر گرفته میشود .

# CS50x Iran

Harvard's Computer Science 50x Iran

