



مبانی برنامه‌سازی کامپیوتر  
آزمون عملی پایتون

سید صالح اعتمادی \*

۴ دی ۹۹

---

\*تشکر ویژه از آقای علی حیدری که نسخه اولیه این قالب را در ترم دوم سال تحصیلی ۹۷-۹۸ برای درس برنامه‌سازی پیشرفته تهیه کردند.

## ۱ آماده‌سازی

### ۱.۱ نکات مورد توجه

- صدا و صفحه نمایش شما باید از طریق نرم‌افزار Flashback recorder به طور کامل از ابتدا تا انتهای امتحان ضبط و ذخیره شود. دقت کنید که پس از نصب نرم‌افزار، در قسمت تنظیمات کیفیت ضبط را ۱ فریم بر ثانیه قرار دهید. ویدیوی امتحان بعد از امتحان جمع‌آوری خواهد شد.
- لازم است از ابتدا تا انتهای امتحان در جلسه Teams با وبکم روشن حضور داشته باشید.
- استفاده از هیچ ویدیویی در طول امتحان مجاز نیست.
- دیدن جزوه یا کدهایی که خود شما قبلاً و برای کلاس زده‌اید جایز است.
- دیدن هرگونه کد از روی اینترنت یا غیراینترنت مجاز نیست. پاسخ‌های هرکس حتماً باید توسط خود او و بدون دیدن هیچ کد دیگری نوشته شده باشد. کمک گرفتن از دیگران در طول مدت امتحان مجاز نیست و منجر به درج نمره‌ی مردود برای این درس می‌شود.
- جستجو به دنبال موضوع سوال یا نمونه کد روی اینترنت مجاز نیست. چنانچه برای رفع خطا یا آشنایی با syntax یک دستور نیاز به جستجو داشتید، اشکالی ندارد. ولی لازم است کلمات کلیدی جستجو شده و آدرس صفحه اینترنتی رویت شده بصورت کامل قبل از تابع کد پیاده‌سازی شده اضافه شود. اضافه نکردن این موارد در بخش کامنت تقلب محسوب می‌شود.
- سوال‌ها در ۴۵ دقیقه‌ای اول امتحان پاسخ داده نمی‌شوند. بعد از آن چنانچه سوالی دارید در بخش چت/گفتگو جلسه یک پیام با عنوان «سوال دارم» بفرستید. یکی از اساتید حل تمرین با شما تماس خواهد گرفت.
- معیار ارزیابی امتحان فقط کدی است که در AzureDevOps با روشی که در ادامه آمده بارگزاری شده است.
- حین امتحان تنها اجازه ارتباط با استاد درس را دارید. هرگونه ارتباط با هر فرد دیگری در جلسه امتحان یا خارج از جلسه امتحان به صورت حضوری یا مجازی مجاز نمی‌باشد.
- خروج از جلسه Teams یا محدوده تصویر وبکم در طول امتحان مجاز نیست.
- خوردن و آشامیدن بلا مانع است.

### ۲.۱ آماده‌سازی‌های اولیه

لازم است پوشه‌ای به نام Exam1 در ریشه گیت شما ایجاد شود و فایل‌های مربوط به امتحان مستقیماً در این پوشه قرار بگیرند.

#### ۱.۲.۱ آماده‌سازی‌های مربوط به git

اگر سر کلاس و کارگاه چند بار مفاهیم و روش کار با git آموزش داده شد اما بار دیگر در اینجا کارهایی را که باید در ابتدای آزمون انجام دهید را مرور می‌کنیم.

✓ ابتدا از یکسان/هماهنگ بودن گیت محل خود با سرور اطمینان حاصل نمایید.

```

1 C:\git\FC99001>git status
2 On branch main
3 Your branch is up to date with 'origin/main'.
4
5 nothing to commit, working tree clean
6
7 C:\git\FC99001>git pull
8 Already up to date.
9
10 C:\git\FC99001>
```

توصیه می‌شود پس از پیاده‌سازی هر تست تغییرات انجام شده را commit و push کنید.

#### ۲.۲.۱ آماده‌سازی‌های مربوط به VSCode

پوشه‌ای با نام Exam1 در ریشه گیت درست کرده و فایل تست exam1\_test.py به همراه فایل‌های کمکی phonebook1.txt و phonebook2.txt را در آن قرار دهید. سپس پوشه Exam1 را با VSCode باز کنید.

## ۲ پیاده‌سازی

### ۱.۲ آماده سازی تست

سوال‌های امتحان بصورت تعدادی تست طراحی شده‌اند که لازم است تابع لازم برای پاس شدن تست را پیاده‌سازی کنید. همه تست‌ها comment شده‌اند.

#### ۱.۱.۲ فعال‌سازی تست‌ها

در قدم اول کامنت‌های مربوط به تست‌ها را یکی-یکی برداشته و از آن شناخته شدن تست توسط `pytest` در `VSCode` اطمینان حاصل کنید. برای این منظور ابتدا کامنت‌های مربوط به یک تست را بردارید. سپس تست را مطالعه کنید. نام تابع مورد تست و پارامترهای ورودی و نوع مقدار برگشتی تابع مورد تست را تشخیص دهید. سپس در فایل `exam1.py` تابع را پیاده‌سازی کنید. اگر این اولین تست است توسط دستور `Python: Configure Tests` در `VSCode` بستر تست `pytest` را فعال کنید. سپس از شناخته شدن تستی که کامنت آن را برداشتید اطمینان حاصل کنید. این فرایند را تا برداشته شدن کلیه کامنت‌های تست ادامه دهید. چنانچه سوالی را نتوانستید حل کنید، تست مربوط به آن سوال را به حالت کامنت برگردانید.

#### ۲.۱.۲ فرستادن تست در Azure DevOps

پس از شناخته شدن کلیه تست‌ها در `VSCode` کد خود را `add/commit/push` کرده و سپس در `Azure DevOps` از موفقیت‌آمیز بودن بیلد اطمینان حاصل نمایید.

### ۲.۲ پیاده‌سازی توابع

از تست شماره یک شروع کرده و کامنت تست را برداشته و تابع متناظر را در فایل `exam1.py` تعریف کنید (بدون پیاده‌سازی کامل تابع) تا تست شناخته شده و فعال شود. سپس تست را اجرا کرده و از عدم اجرای موفقیت‌آمیز آن اطمینان حاصل کنید. سپس تابع مورد استفاده تست را بگونه‌ای پیاده‌سازی کنید که تست با موفقیت پاس شود.

#### ۱.۲.۲ تست `test_q1_add`

تابع `q1_add` دو عدد به عنوان پارامتر گرفته و جمع آنها را برمی‌گرداند.

#### ۲.۲.۲ تست `test_q2_print_add`

تابع `q2_print_add` دو عدد از ورودی گرفته و حاصل جمع آنها را در صفحه نمایش چاپ می‌کند.

#### ۳.۲.۲ تست `test_q3_print_square`

تابع `q3_print_square` یک کاراکتر `c` و یک عدد `n` به عنوان پارامتر دریافت می‌کند و در صفحه نمایش مربعی با طول ضلع `n` توسط کاراکتر `c` رسم می‌کند. به عنوان مثال، برای کاراکتر `A` و عدد `5` خروجی زیر را چاپ می‌کند.

```
1 AAAAA
2 A    A
3 A    A
4 A    A
5 AAAAA
```

#### ۴.۲.۲ تست `test_q4_sum_odd_squares`

تابع `q4_sum_odd_squares` دو عدد به عنوان پارامتر دریافت می‌کند و مجموع توان ۲ اعداد فرد بین این دو عدد (شامل خود دو عدد) را بر می‌گرداند. برای مثال به تست مراجعه کنید.

#### ۵.۲.۲ تست `test_q5_sum_value_squares`

تابع `q5_sum_value_squares` لیستی از اعداد به عنوان پارامتر دریافت می‌کند و مجموع توان ۲ آنها را بر می‌گرداند. برای مثال به تست مراجعه کنید.

**۶.۲.۲ تست test\_q6\_sum\_num\_indices**

تابع `q5_sum_value_squares` لیستی از اعداد به همراه لیستی از اندیس‌ها در قالب دو لیست به عنوان پارامتر دریافت می‌کند و مجموع اعدادی از لیست اول که اندیس آنها در لیست دوم موجود است برمی‌گرداند. برای مثال به تست مراجعه کنید.

**۷.۲.۲ تست test\_q7\_get\_new\_fib\_array**

تابع `q7_get_new_fib_array` سه عدد  $x_1, x_2$  و  $n$  را به عنوان پارامتر دریافت می‌کند و لیستی  $n$  تایی که با  $x_1, x_2$  شروع می‌شوند را برمی‌گرداند. عناصر بعدی لیست ضرب دو عنصر قبل منهای ۲ می‌باشند:

$$x_{i+1} = x_i \times x_{i-1} - 2$$

برای مثال به تست مراجعه کنید. استفاده از تابع بازگشتی برای حل این سوال جایز نیست.

**۸.۲.۲ تست test\_q8\_get\_phone\_numbers**

تابع `q8_get_phone_numbers` نام یک فایل به همراه لیستی از اسامی به عنوان پارامتر دریافت می‌کند و شماره تلفن این افراد را (به همان ترتیب اسامی) در یک لیست برمی‌گرداند. ساختار فایل ورودی به این شکل است که هر خط از فایل یک نام و شماره تلفن دارد که با کاراکتر کاما از هم جدا شده‌اند. برای مثال فایل `phonebook1.txt` و تست را ببینید.

**۳ ارسال**

اگر موفق به پاس شدن تستی نشدید تست را به حالت کامنت برگردانید. پس از پیاده‌سازی توابع و پاس شدن تست‌هایی که فرصت کردین، نوبت به ارسال آنها می‌رسد. مثل قبل تغییرات را `add/commit/push` کنید و از موفقیت‌آمیز بودن بیلد روی AzureDevOps اطمینان حاصل نمایید.

**۱.۳ ساخت Pull Request**

با مراجعه به سایت [Azure DevOps](#) لز موفقیت بیلد برای Pull Request که در مرحله اول درست کردید اطمینان حاصل کنید و آنرا کامل کنید. دقت کنید که گزینه `Delete source branch` نباید انتخاب شود.