F. Ebrahimi:
تولید 100 عدد تصادفی و ذخیره در فایل

```
import random

def generate_random_numbers(file_name, count=100):
    with open(file_name, 'w') as file:
        for _ in range(count):
            number = random.randint(1, 100)  # تولید عدد تصادفی بین 1 تا 100
            file.write(f"{number}\n")

# فراخوانی تابع
generate_random_numbers('data.txt')
```

مراحل:
وارد کردن کتابخانه random: این کتابخانه برای تولید

اعداد تصادفی استفاده می‌شود.

• تعریف تابع generate_random_numbers: این تابع دو آرگومان می‌گیرد: file_name (نام فایلی که اعداد در آن ذخیره می‌شوند) و count (تعداد اعداد تصادفی که باید تولید شود).

• باز کردن فایل: با استفاده از with open(file_name, 'w')، فایل برای نوشتن باز می‌شود. اگر فایل وجود نداشته باشد، ایجاد می‌شود.

• تولید اعداد تصادفی: با استفاده از یک حلقه for، 100 عدد تصادفی بین 1 تا 100 تولید شده و در فایل نوشته می‌شوند.

• فراخوانی تابع: در انتها، تابع با نام فایل 'data.txt' فراخوانی می‌شود

خواندن اعداد از فایل و تعیین زوج یا فرد بودن آنها

def check_even_odd(file_name):

```
with open(file_name, 'r') as file:
lines = file.readlines()
for line in lines:
number = int(line.strip())
if number % 2 == 0:
print(f"{number} - زوج")
else:
print(f"{number} - فرد")

# فراخوانی تابع
check_even_odd('data.txt')
```

مراحل:

- تعریف تابع check_even_odd: این تابع یک آرگومان می‌گیرد: file_name (نام فایلی که اعداد در آن ذخیره شده‌اند).

- باز کردن فایل: با استفاده از with open(file_name, 'r')، فایل برای خواندن باز می‌شود.

• خواندن خطوط فایل: با استفاده از file.readlines()، تمام خطوط فایل به یک لیست تبدیل می‌شوند.

• بررسی زوج یا فرد بودن: با استفاده از یک حلقه for، هر عدد خوانده شده بررسی می‌شود که آیا زوج است یا فرد و نتیجه چاپ می‌شود.

• فراخوانی تابع: در انتها، تابع با نام فایل 'data.txt' فراخوانی می‌شود

تابعی برای دریافت نمرات دانشجویان و ذخیره در فایل

```
def save_student_grades(file_name):
grades = []
while True:
 grade = input("لطفا نمره دانشجو را وارد کنید
(" :(را وارد کنید 'done' برای پایان دادن)
if grade.lower() == 'done':
break
try:
```

```
            grades.append(float(grade))
        except ValueError:
            print("لطفا یک عدد معتبر وارد کنید.")

    with open(file_name, 'w') as file:
        for grade in grades:
            file.write(f"{grade}\n")

# فراخوانی تابع
save_student_grades('grades.txt')
```

مراحل:
تعریف تابع save_student_grades: این تابع یک آرگومان می‌گیرد: file_name (نام فایلی که نمرات در آن ذخیره می‌شوند).

- دریافت نمرات از کاربر: با استفاده از یک حلقه while، نمرات از کاربر دریافت می‌شود تا زمانی که کاربر عبارت 'done' را وارد کند.

- بررسی ورودی: اگر ورودی عددی نباشد، پیام خطا

نمایش داده می‌شود.

• ذخیره نمرات در فایل: پس از پایان دریافت نمرات، آنها در فایل مشخص شده ذخیره می‌شوند.

• فراخوانی تابع: در انتها، تابع با نام فایل 'grades.txt' فراخوانی می‌شود

تابعی برای محاسبه معدل دانشجو از فایل نمرات

```
def calculate_average_grade(file_name):
    with open(file_name, 'r') as file:
        lines = file.readlines()
        total = 0
        count = 0
        for line in lines:
            total += float(line.strip())
            count += 1
        if count > 0:
            average = total / count
            print(f"معدل دانشجو: {average:.2f}")
```

```
else:
    print("هیچ نمره‌ای موجود نیست.")

# فراخوانی تابع
calculate_average_grade('grades.txt')
```

مراحل:

تعریف تابع calculate_average_grade: این تابع یک آرگومان می‌گیرد: file_name (نام فایلی که نمرات در آن ذخیره شده‌اند).

- باز کردن فایل: با استفاده از with open(file_name, 'r')، فایل برای خواندن باز می‌شود.

- محاسبه مجموع و تعداد نمرات: با استفاده از یک حلقه for، مجموع نمرات و تعداد آنها محاسبه می‌شود.

- محاسبه معدل: اگر تعداد نمرات بیشتر از صفر باشد، معدل محاسبه و چاپ می‌شود. در غیر این صورت، پیام

مناسبی نمایش داده می‌شود