## سوال اول

برای پاسخ به سوال از ساب ماژول انتگرال‌گیری در ماژول scipy استفاده شده است. همچنین با توجه به اینکه در صورت مسئله گام حل عددی بیان شده است، از روش – تابع سیمپسون برای انتگرال‌گیری استفاده شده است. برای مقایسه با روش‌های حل بهینه گام متغیر، انتگرال‌گیری با روش – تابع کواد نیز انجام شده است.

روش کار کلی نیز در این برنامه به این صورت است که تابعی که قصد انتگرال‌گیری داریم، با def و با نام sin2x (x) تعریف شده است که در خروجی sin(x*x) را برمی‌گرداند. همچنین بازه انتگرال‌گیری توسط a = -3 و b = 4 و گام حرکت توسط step = 1/200 عملا بعنوان ورودی تعریف شده‌اند. نهایتا انتگرال‌گیری توسط توابع integrate.simps() برای گام حرکتی ثابت و integrate.quad() برای گام حرکتی متغیر بصورت عددی محاسبه می‌شود و در خروجی برنامه نتایج به شرح زیر چاپ می‌شوند:

```
Integration with fixed step size - Simpson approach:  1.520696364204743
Integration with variable step size - Quad approach:  (1.5206963715418835, 3.631605848744416e-09)
```

روش کواد خروجی دقیق‌تری می‌دهد. همچنین عبارت بسیار کوچک دوم در خروجی کواد، خطای محاسبه است.

## سوال دوم

روش حل بسیار ساده است. عدد n توسط تابع isPrime(n) گرفته می‌شود. این عدد بر تمام اعداد کمتر از خودش تقسیم می‌شود. اگر باقیمانده برابر ۰ بود، در حافظه شمارنده ۱ واحد بیشتر می‌شود. همچنین این عدد کمتر هم در یک آرایه ذخیره می‌شود. عدد اول عددی است که فقط بر ۱ و خودش بخش پذیر بصورت کامل است. پس اگر شمارنده ۲ باشد (پس از انجام همه تقسیم ها با کمک حلقه)، آن عدد اول است اما اگر شمارنده ۲ نباشد، آن عدد اول نیست.

خروجی تابع نیز یک متغیر Boolean است که برای اعداد ورودی اول مقدار True و برای اعداد ورودی غیراول مقدار False را صادر می‌کند. همچنین با یک تابع پرینت، اگر عددی اول نباشد، چاپ می‌کند که این عدد ب چه اعدادی بصورت کامل تقسیم پذیر بوده و عملا توجیه غیراول بودن آن را نشان می‌دهد. خروجی بررسی عملکرد تابع بصورت زیر است.

```
>>>
 RESTART: D:\Javad\Books\University Books\Image Processing\Home works\HW0_981260
79\codes\question 2\HW0_q2_Zallaghi_98126079.py
2  is a prime number!


4  is not a prime number!
see below:
 [1, 2, 4] : list of perfect devidable numbers...


7  is a prime number!


8  is not a prime number!
see below:
 [1, 2, 4, 8] : list of perfect devidable numbers...


11  is a prime number!


24  is not a prime number!
see below:
 [1, 2, 3, 4, 6, 8, 12, 24] : list of perfect devidable numbers...


Printing bool value returned by isPrime() for upper subjects
 True False True False True False
```

# سوال سوم

## ۱ – تابع ریاضی نگاشت

$$x \in [a,b] \quad and\ x' \in [0,255]$$

$$x' = \left[\frac{abs(255-0)}{abs(b-a)} \times (x-a)\right]$$

## ۲ – پیاده‌سازی در پایتون

تابع convertor(inputArray) در برنامه برای پیاده‌سازی قسمت قبل نوشته شده است. در این برنامه با توابع amin() و amax() از ماژول numpy عملا a و b را از ورودی تعیین می‌کنیم. سپس یک تابع اسکالر mapper(x) نوشته شده که مقدار xprime را تعیین می‌کند. نهایتا با تابع np.vectorize()، تابع اسکالر را به برداری تبدیل کردیم تا براحتی روی تمام عناصر آرایه ورودی اعمال کنیم و در خروجی آرایه عملا با دیتاتایپ uint8 ست می‌شود. همچنین کامنت‌ها در برنامه عملکرد آن را مشخص کرده‌اند.

## ۳- نمایش تصویر ایجاد شده از ورودی تصادفی

با تابع np.random.uniform() یک آرایه ۳ بعدی با ابعاد ۵۰*۴۰*۳ و در بازه -۳,۲ تا ۹,۳ بصورت رندوم در کد ایجاد می‌شود. سپس این آرایه را به تابع convertor() می‌دهیم تا تمام اعضای آن به استاندارد uint8 بصورت ۰-۲۵۵ نگاشت شوند. سپس با تابع cv2.imwrite() با آرایه استاندارد شده یک تصویر دیجیتال RGB ایجاد می‌کنیم. نهایتا با توابع imshow() و waitKey() نیز آن را نمایش می‌دهیم. این تصویر بصورت زیر است:

ابعاد کوچک تصویر بخاطر رزولوشن پایین آن است که در متن سوال خواسته شده است.

## سوال چهارم

تابع polDerCoefVec(coeffVec) تعریف شده است تا با گرفتن شی ضرایب چندجمله‌ای بصورت numpy، ضرایب مشتق چندجمله‌ای را بصورت outPut[i] = (rank-i) * coeffVec[i] محاسبه نماید و خروجی بدهد. توجه داریم مرتبه چند جمله‌ای ورودی توسط چندجمله-ای و با تابع size() از ماژول numpy محاسبه می‌شود. سپس یک سهمی با ضرایب [1,2,-5] تعریف می‌شود و به تابع تعریف شده داده می‌شود تا ضرایب مشتق آن که عملا معادله خط می‌شود را بدهد. سپس در دامنه دلخواه و با تابع np.polyval() مقدار چند جمله‌ای و مشتق آن را محاسبه و ذخیره کردیم. سپس نتایج را با رعایت موارد خواسته شده برای چاپ نمودار نتایج با ماژول matplotlib ترسیم کردیم. خروجی نمودار بصورت زیر می‌شود: