```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats, linalg, interpolate, optimize
from scipy.fftpack import fft2
from scipy.signal import butter, filtfilt
from scipy.integrate import odeint

a = np.random.rand(10)
print(a.mean(), np.median(a), a.var())

A = np.random.rand(4,4)
print(fft2(A))

B = np.array([[2,3],[1,4]])
print(linalg.det(B), linalg.inv(B), linalg.eigvals(B))

x = np.array([0,1,2,3])
y = np.array([2,3,5,10])
f = interpolate.interp1d(x, y, kind='cubic')
print(f(1.5))

t = np.linspace(0,10,200)
x = np.sin(t) + 0.4*np.random.randn(200)
b,a = butter(3, 0.1)
print(filtfilt(b,a,x))

sales = np.random.randint(100,500,(12,4))
print(sales.sum(axis=1))
print("Best:", np.argmax(sales.sum(axis=1))+1)
print("Worst:", np.argmin(sales.sum(axis=1))+1)

M = np.array([[85,78,92,88],[79,82,74,90],[90,85,89,92],[66,75,80,78],[70,68,75,85]])
total = M.sum(axis=1)
avg = M.mean(axis=1)
print(total, avg)
print("Top:", total.argmax())
print("Bottom:", total.argmin())
print("Passing %:", np.mean(total>=200)*100)

t = np.array([0,1,2,3,4,5])
v = np.array([2,3.1,7.9,18.2,34.3,56.2])
fit = lambda t,a,b,c: a*t*t + b*t + c
p,_ = optimize.curve_fit(fit,t,v)
print(p)

plt.plot(M.sum(axis=1))
plt.savefig("q9.png")

plt.figure()
plt.scatter(t,v)
plt.plot(t,fit(t,*p))
plt.savefig("q10.png")

year = np.array([2000,2005,2010,2015,2020])
pop = np.array([50,55,70,80,90])
print(np.corrcoef(year,pop)[0,1])
fi = interpolate.interp1d(year,pop)
print("2008:", fi(2008))
plt.figure()
plt.plot(year,pop,"o-")
plt.savefig("q11.png")

from numpy.polynomial import Polynomial as P
poly = P([-8,2,-5,3])
print(poly.roots())
xv = np.linspace(-3,3,200)
plt.figure()
plt.plot(xv, poly(xv))
plt.savefig("q12.png")

import time
sizes=[200,400,600,800,1000]
```

```
for s in sizes:
    text=("a"*1_000_000)*s
    t0=time.time()
    text.upper()
    print(s, time.time()-t0)

f2=lambda x: x**4 - 3*x**3 + 2
from scipy.optimize import minimize
res=minimize(f2,0)
print(res.x)
xv=np.linspace(-2,3,200)
plt.figure()
plt.plot(xv,f2(xv))
plt.savefig("q14.png")

def model(y,t):
    x,v=y
    return [v, -0.2*v - x]

t=np.linspace(0,20,400)
y=odeint(model,[1,0],t)
plt.figure()
plt.plot(t,y[:,0])
plt.savefig("q15.png")
print("Max:", y[:,0].max())
```