

Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут”
Факультет електроніки
Кафедра мікроелектроніки

ЛАБОРАТОРНА РОБОТА № 1

з курсу
«Теорія сигналів»
"Основи програмування мовою Python"

Студента 3 курсу
групи ДП-81
Фіцая Богдана

Київ — 2020

Пункт 3

Ознайомитися:

- з задаванням масиву, елементи якого є арифметичною послідовністю;
 - з роботою функцій генерації випадкових чисел із заданими густинами розподілу імовірності.
- Ознайомитися з функцією побудови гістограм, побудувати гістограми випадкових чисел з різними розподілами густини ймовірності;

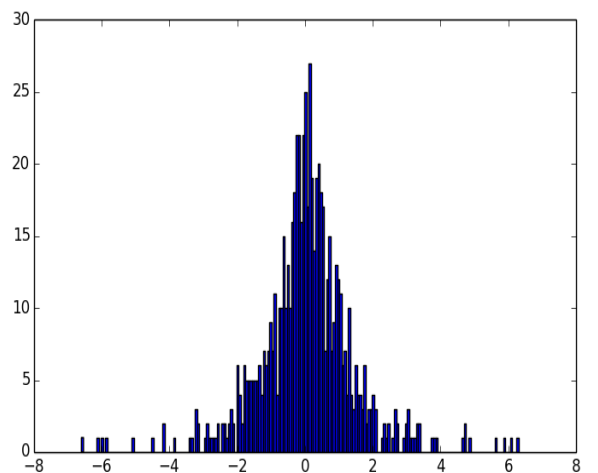
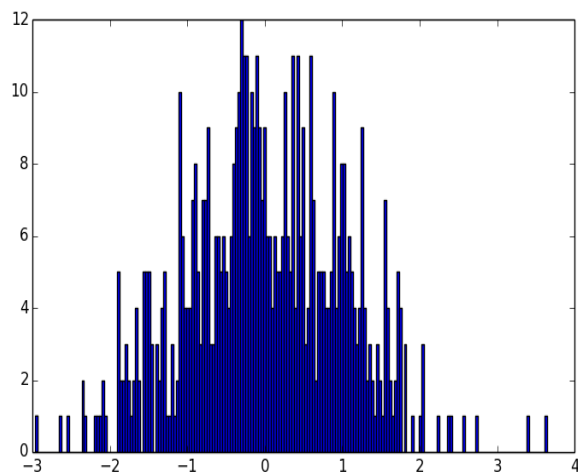
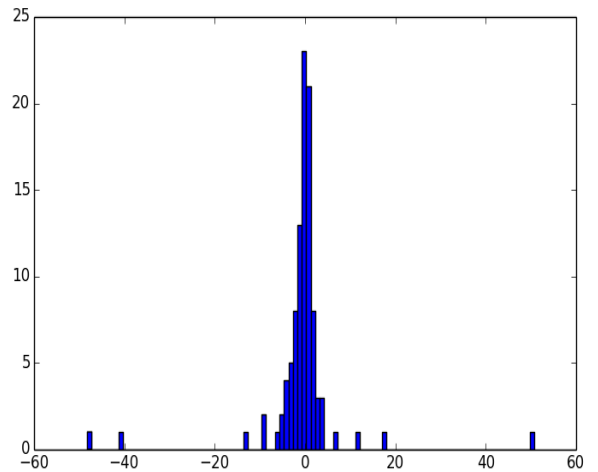
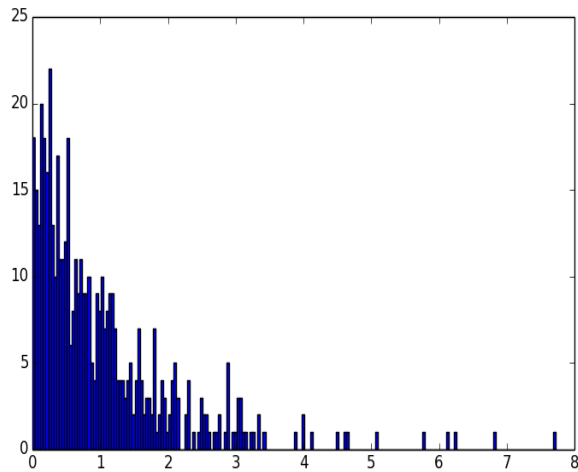
```
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import numpy as np
mas = [i for i in range(50)]
print(mas)

fig ,axs = plt.subplots(2,2)

a = np.random.standard_exponential(500)
b = np.random.standard_cauchy(100)
c = np.random.normal(size = 600)
d = np.random.laplace(size = 700)

axs[0,0].hist(a, bins=200)#Exponential distribution
axs[0,1].hist(b, bins=100)#Cauchy distribution
axs[1,0].hist(c, bins=200)#Normal distribution
axs[1,1].hist(d, bins=200)#Laplace distribution

plt.show()
```

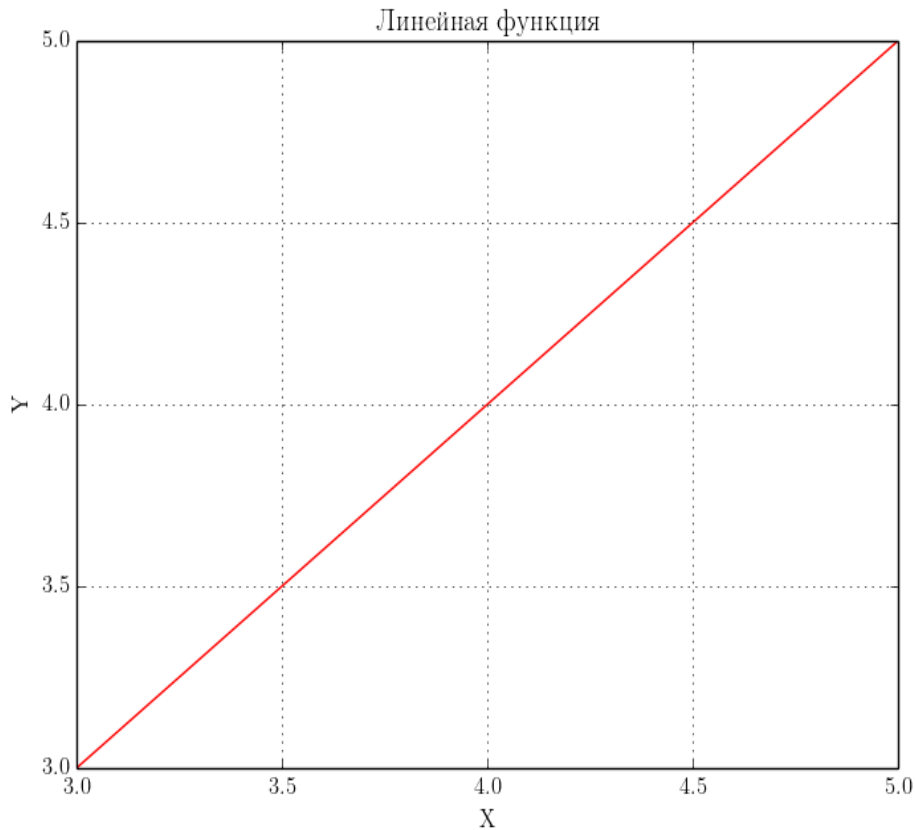


Пункт 4

Ознайомитися з написанням власних файлів-сценаріїв. У власному файлі-сценарії побудувати графік лінійної функції однієї змінної. Позначити вісі та заголовок графіку, нанести координатну сітку.

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import rc
rc('text', usetex=True)
rc('text.latex', unicode=True)
rc('text.latex', preamble='\usepackage[utf8]{inputenc}')
rc('text.latex', preamble='\usepackage[russian]{babel}')

x = np.linspace(3, 5, 10)
y = 1*x
plt.title(u"Линейная_функция")
plt.xlabel('X')
plt.ylabel('Y')
plt.grid(True)
plt.plot(x, y, "r")
plt.show()
```



Пункт 5

5.1 Побудувати графіки синусоїд частот 1, 10, 50 Гц. Тривалість сигналів – 1 сек., частота дискретизації 256 Гц. Графіки будувати в одному вікні, але в різних осях. Амплітуди кожної синусоїди повинні бути випадковими числами.

5.2 Виконати теж саме, але задавати амплітуду кожної синусоїди з клавіатури.

5.3 Підписати заголовок кожного графіку текстом, який буде містити значення частоти та амплітуди відповідної синусоїди.

```
from numpy import array, arange, abs as np_abs
```

```
from math import sin, pi
import matplotlib.pyplot as plt
import matplotlib as mpl
import random
```

```
fr=1.
dfr = 256
dlinna = 256
ampl=random.uniform(1,10)
W=2.*pi*fr/dfr
```

```
plt.subplot(3,1,1)
signal = array([ampl*sin(W*a) for a in range(dlinna)])
plt.plot(arange(dlinna)/float(dfr), signal, 'y')
plt.xlabel('Time, s')
```

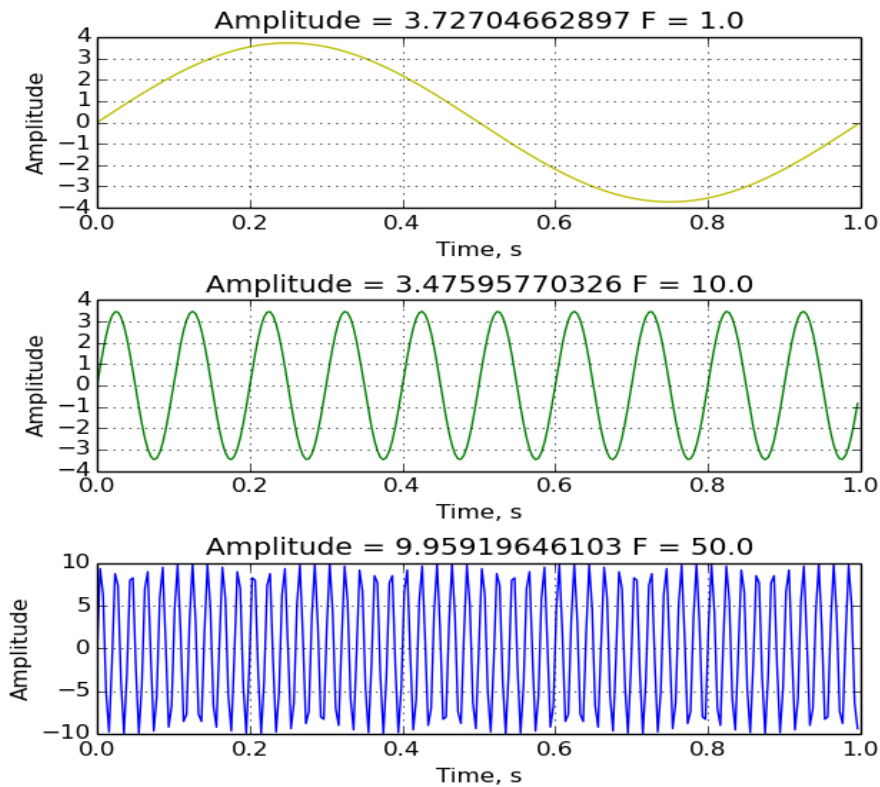
```
plt.ylabel('Amplitude')
plt.title('Amplitude_{}_{}_F_{}_{}'.format(ampl, ' ', fr))
plt.grid(True)
plt.hold(True)
```

```
fr=10.
dfr = 256
dlinna = 256
ampl=random.uniform(1,10)
W=2.*pi*fr/dfr
```

```
plt.subplot(3,1,2)
signal = array([ampl*sin(W*a) for a in range(dlinna)])
plt.plot(arange(dlinna)/float(dfr), signal, 'g')
plt.xlabel('Time,s')
plt.ylabel('Amplitude')
plt.title('Amplitude_{}_{}_F_{}_{}'.format(ampl, ' ', fr))
plt.grid(True)
plt.hold(True)
```

```
fr=50.
dfr = 256
dlinna = 256
ampl=random.uniform(1,10)
W=2.*pi*fr/dfr
```

```
plt.subplot(3,1,3)
signal = array([ampl*sin(W*a) for a in range(dlinna)])
plt.plot(arange(dlinna)/float(dfr), signal, 'b')
plt.xlabel('Time,s')
plt.ylabel('Amplitude')
plt.title('Amplitude_{}_{}_F_{}_{}'.format(ampl, ' ', fr))
plt.grid(True)
plt.hold(True)
plt.show()
```



Пункт 6

6.1 Побудувати одиночний прямокутний імпульс. Задати проміжок значень часу 10 секунд, частота дискретизації 256 Гц. Побудувати графік одиничного прямокутного імпульсу шириною 300 мс, з центром в момент часу 4 с.

```
from numpy import array, arange, abs as np_abs
import matplotlib.pyplot as plt
from matplotlib import rc
```

```
rc('text', usetex=True)
rc('text.latex', unicode=True)
rc('text.latex', preamble='\usepackage[utf8]{inputenc}')
rc('text.latex', preamble='\usepackage[russian]{babel}')
```

```
fd = 256.
dt = 1./fd
Amp = 5.
b = 0.3
a = 4.0 - b/2
t_minimum = 0
t_maximum = 10.
```

```
def qwe(x):
    if x < a:
        return 0
    if x < a+b:
        return Amp
    return 0
```

```
x = [];
```

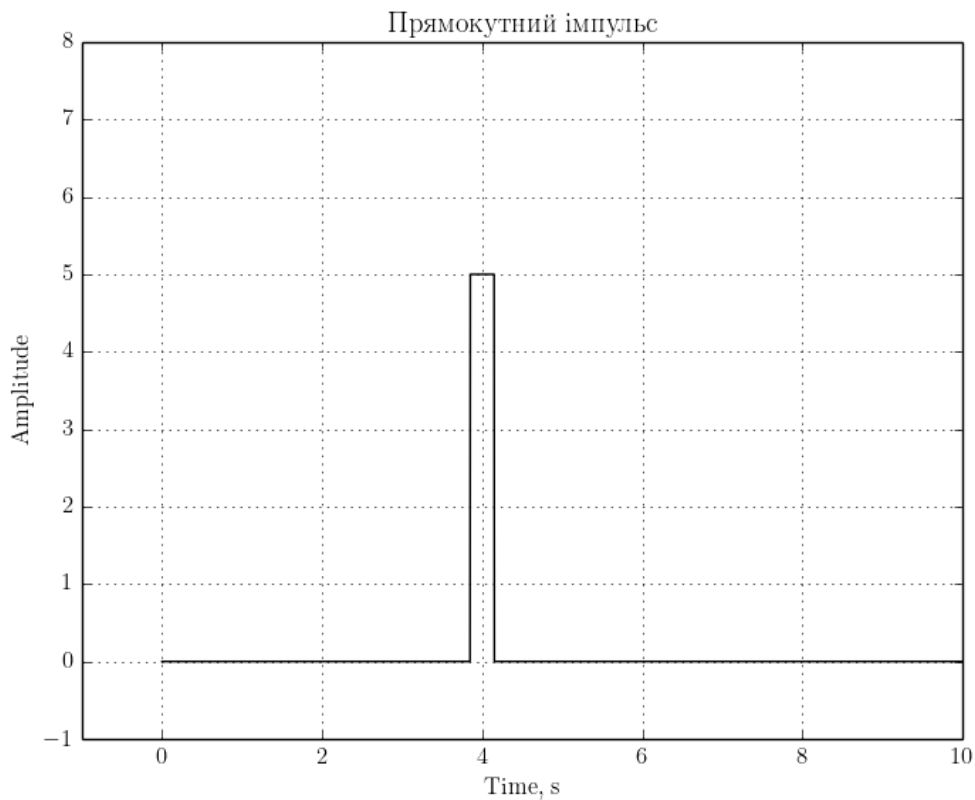
```

y = [];
t = t_minimum
while t<=t_maximum:
    x.append(t)
    y.append(qwe(t))
    t += dt

axes = plt.gca()
axes.set_xlim([-1,10])
axes.set_ylim([-1,8])
plt.plot(x, y, 'k')
plt.xlabel('Time, s')
plt.ylabel('Amplitude')
plt.title(u'Прямокутний імпульс')
plt.grid(True)

plt.show()

```



6.2 Написати файл-сценарій для побудови графіку прямокутного імпульсу, тривалість та амплітуда якого буде задаватися з клавіатури. Розташування імпульсу задавати випадковим числом, але передбачити перевірку, чи не виходить імпульс за межі графіка.

```

from numpy import array, arange, abs as np_abs
from math import sin, pi
import matplotlib.pyplot as plt
import matplotlib as mpl
import random
import numpy as np
from matplotlib import rc

```

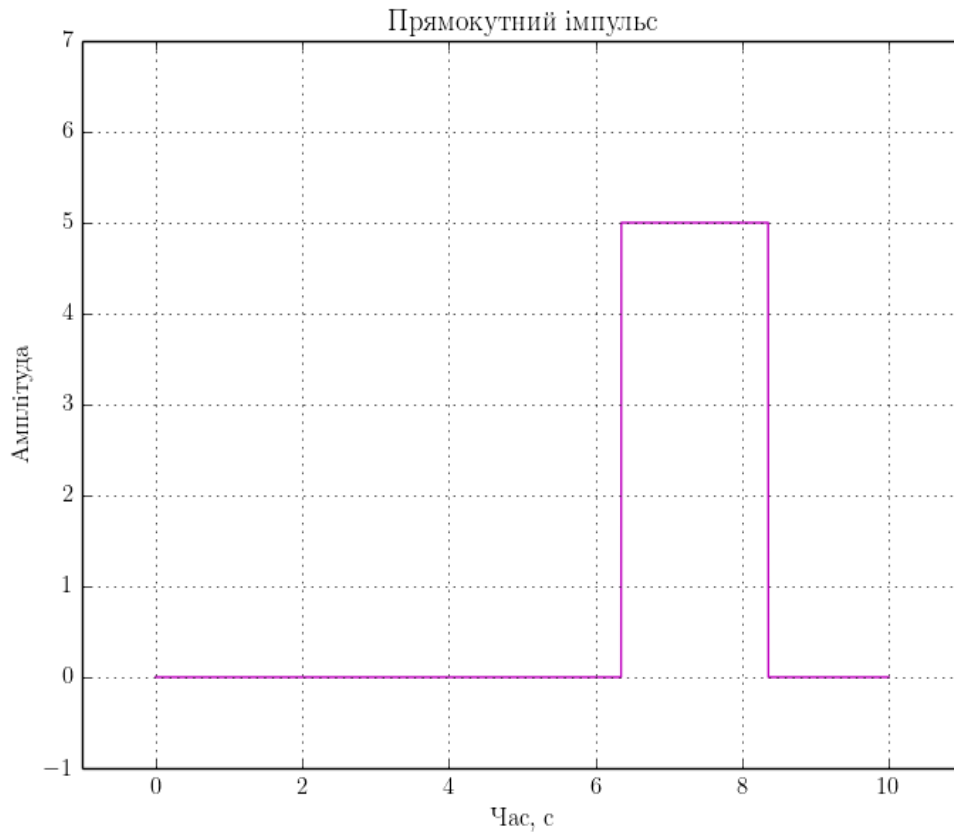
```

rc('text', usetex=True)
rc('text.latex', unicode=True)
rc('text.latex', preamble='\usepackage[utf8]{inputenc}')
rc('text.latex', preamble='\usepackage[russian]{babel}')

def sqwave(p,X,T,aa,A,fd):
    x = []
    r = np.arange(p,X,1.0/fd)
    taue = aa+T
    x00 = r[(np.abs(r - aa)).argmin()]
    x01 = r[(np.abs(r - taue)).argmin()]
    if aa+T > X:
        return 1
    for i in r:
        if i >= x00 and i <= x01:
            x.append(A)
        else:
            x.append(0.0)
    return [r, x]

print('Input_duration_of_impuls:')
duration = float(input())
print('Enter_Amplitude:')
Amp = float(input())
a = sqwave(0, 10, duration, np.random.rand(1) * 10.0, Amp, 256.0)
while a == 1:
    a = sqwave(0, 10, duration, np.random.rand(1) * 10.0, Amp, 256.0)
plt.plot(a[0], a[1], 'm')
plt.title(u'Прямокутний_імпульс')
plt.ylabel(u'Амплітуда')
plt.xlabel(u'Час,_с')
plt.grid(True)
axes = plt.gca()
axes.set_xlim([-1,11])
axes.set_ylim([-1,Amp+2.0])
plt.show()

```

6.3 Побудувати послідовність прямокутних імпульсів для двох випадків: а) коли інтервали між імпульсами однакові, б) коли інтервали між імпульсами випадкові і задаються програмно.

```

from numpy import array, arange, abs as np_abs
from math import sin, pi
import matplotlib.pyplot as plt
import matplotlib as mpl
import random
from matplotlib import rc

rc('text', usetex=True)
rc('text.latex', unicode=True)
rc('text.latex', preamble='\usepackage[utf8]{inputenc}')
rc('text.latex', preamble='\usepackage[russian]{babel}')

fd = 256.
t_minimum = -1.
t_maximum = 101.
delta_t = 1.0/fd;
print ( 'Амплітуда: \_')
A = input()
print ( 'Ширина_from_1_up_to_10_: \_')
b = input()
print ( 'Відстань_між_імпульсами: \_')
a = input()
if b>10:
    print (u'мпульсI_за_графкомі !!!!!!!!!!! ')
    exit()
def qwe(x,imp_n,imp_d):
    if x < imp_n:

```

```

        return 0;
    if x < imp_n+imp_d:
        return A;
    return 0;

```

```

x = [];
y = [];
t = t_minimum;
while t<=t_maximum:
    k = t
    l = t + a + b
    m = t+a
    while k<=l:
        x.append(t);
        y.append(qwe(t,m,b));
        k += delta_t
        t += delta_t

```

```

plt.subplot(2,1,1)
plt.plot(x, y, 'k')
plt.title(u'Рандомна_відстань')
plt.xlabel('Time,s')
plt.ylabel('Amplitude')
plt.grid(True)
axes = plt.gca()
axes.set_xlim([-1,101])
axes.set_ylim([-1,A+1])
plt.hold(True)

```

```

x1 = [];
y1 = [];
t = t_minimum;
while t<=t_maximum:
    k = t
    o = random.uniform(1,10)
    l = t + o + b
    m = t+o
    while k<=l:
        x1.append(t);
        y1.append(qwe(t,m,b));
        k += delta_t
        t += delta_t

```

```

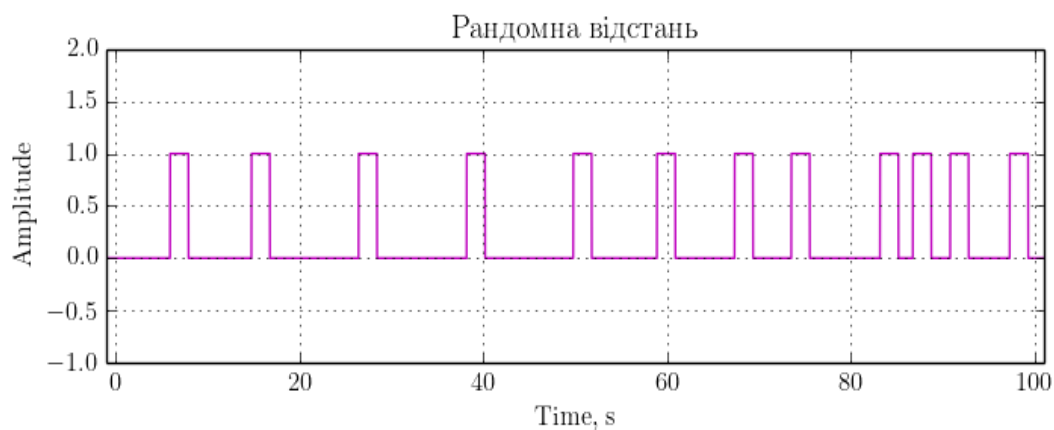
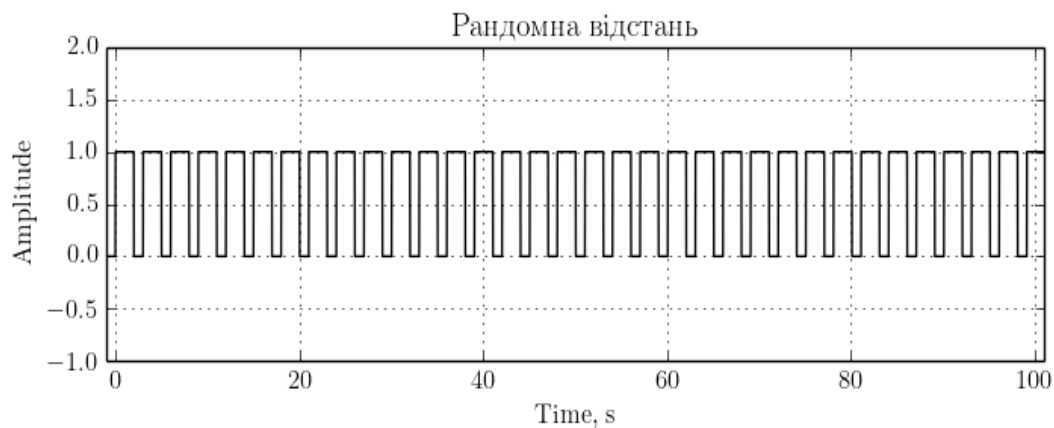
plt.subplot(2,1,2)
plt.plot(x1, y1, 'm')
plt.title(u'Рандомна_відстань')
plt.xlabel('Time,s')
plt.ylabel('Amplitude')
plt.grid(True)
axes = plt.gca()

```

```

axes.set_xlim([-1,101])
axes.set_ylim([-1,A+1])
plt.hold(True)
plt.show()

```



Пункт 7

Зберегти дані розрахунку функції в файл. Прочитати їх із файлу в іншому сценарії, побудувати графік функції.

```

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import rc

```

```

rc('text', usetex=True)
rc('text.latex', unicode=True)
rc('text.latex', preamble='\usepackage[utf8]{inputenc}')
rc('text.latex', preamble='\usepackage[russian]{babel}')

```

```

ex_2 = np.load('SAVE.npz')
ex_2.files
x = ex_2['arr_0']
y = ex_2['arr_1']
Amplitude = ex_2['arr_2']

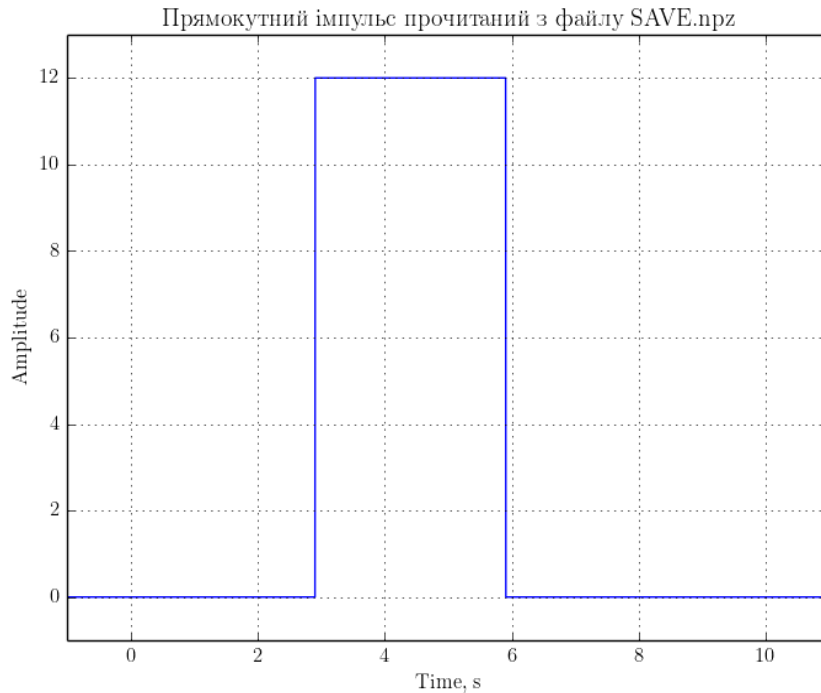
```

```

plt.plot(x, y, Amplitude, 'k')
plt.title(u'Прямокутний_імпульси_прочитаний_з_файлу_SAVE.npz')

```

```
plt.xlabel('Time, s')
plt.ylabel('Amplitude')
plt.grid(True)
axes = plt.gca()
axes.set_xlim([-1, 11])
axes.set_ylim([-1, Amplitude+1])
plt.show()
```



Пункт 8

Побудувати власний файл-функцію для побудови графіка синусоїдального сигналу із заданою частотою, амплітудою та тривалістю для частоти дискретизації 256 Гц. В якості вихідного параметру функції вивести середнє значення синусоїди.

```
from numpy import array, arange, abs as np_abs
from math import sin, pi
import matplotlib.pyplot as plt
import random
from matplotlib import rc
rc('text', usetex=True)
rc('text.latex', unicode=True)
rc('text.latex', preamble='\usepackage[utf8]{inputenc}')
rc('text.latex', preamble='\usepackage[russian]{babel}')

def qwe():
    f=1.
    fd = 256
    n = 256
    W=(2.*pi*f/fd)
    Amplitude=random.uniform(1,10)

    signal = array([Amplitude*sin(W*q) for q in range(n)])
    plt.plot(arange(n)/float(fd), signal, 'b')
```

```

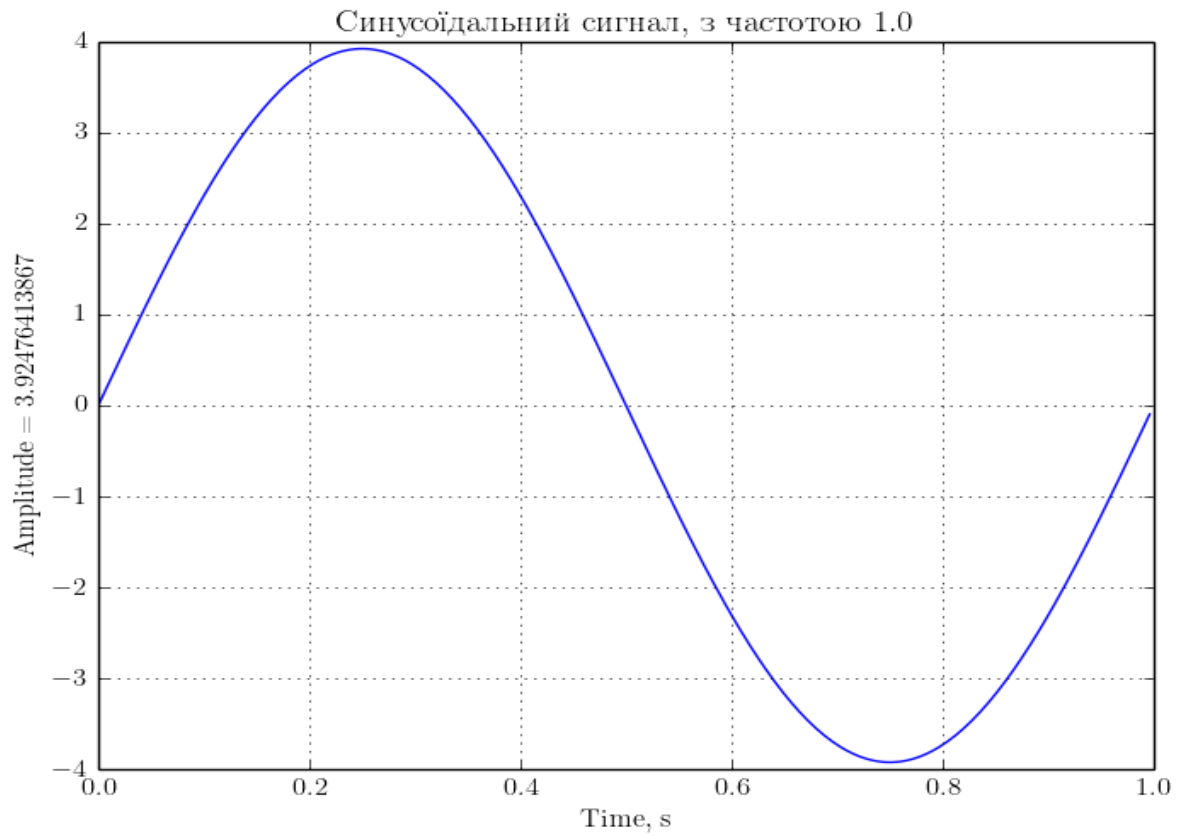
plt.xlabel( 'Time, s ')
plt.ylabel( 'Amplitude=' + str( Amplitude ))
plt.title( u'Синусоїдальний сигнал, з частотою ' + str(( f )))
plt.grid( True )
plt.hold( True )
plt.show()
return A*0.637

```

```

print ( qwe() )

```



Відповідь: середнє значення синусоїди = 2.18