

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Лабораторна робота №2

«Моделювання Пуассонівського та
Вінерівського випадкового процесу»
з кредитного модуля «Випадкові процеси»

Варіант 11

Виконав:

студент групи КМ-93

Костенко Олександр Андрійович

Викладач:

Пашко Анатолій Олексійович

Зміст

Завдання 1	3
Завдання 2	7

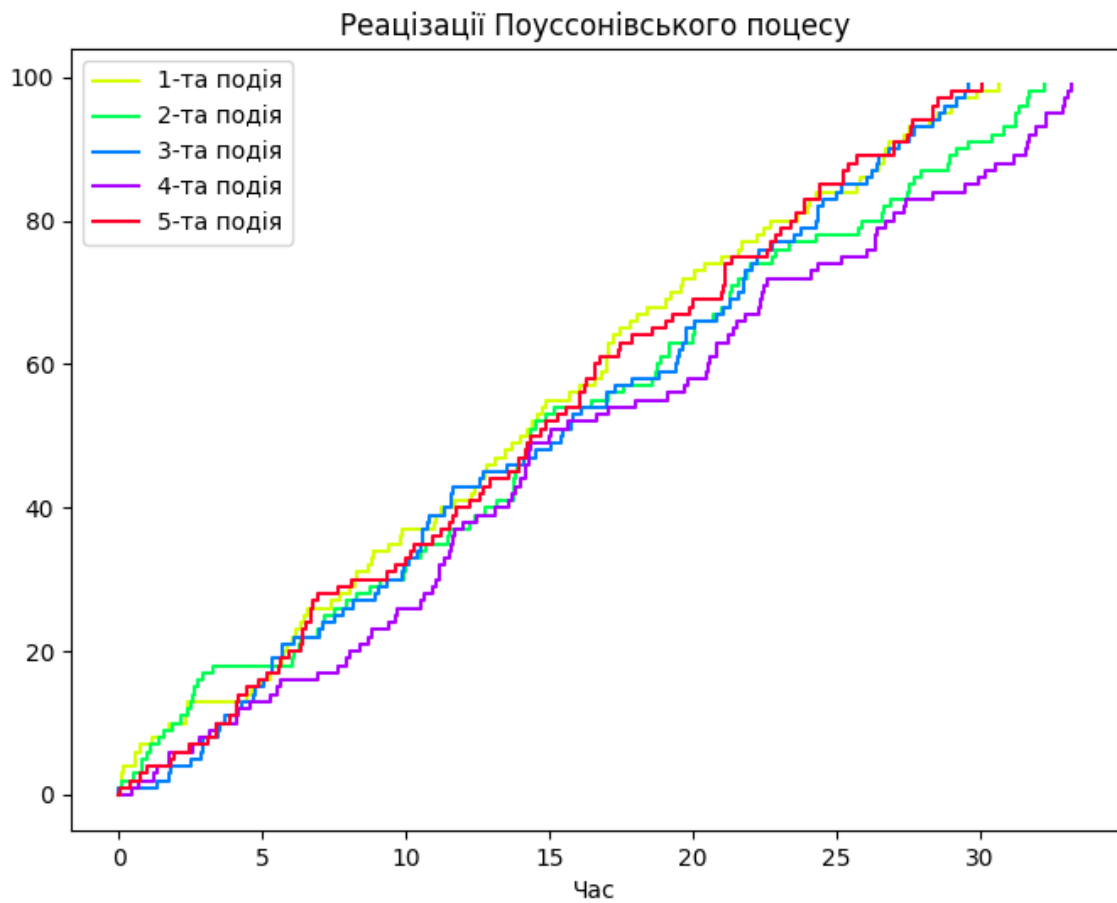
Завдання 1

1. Змодельувати Пуассонівський потік з заданою інтенсивністю.
Побудувати графіки реалізацій процесу.

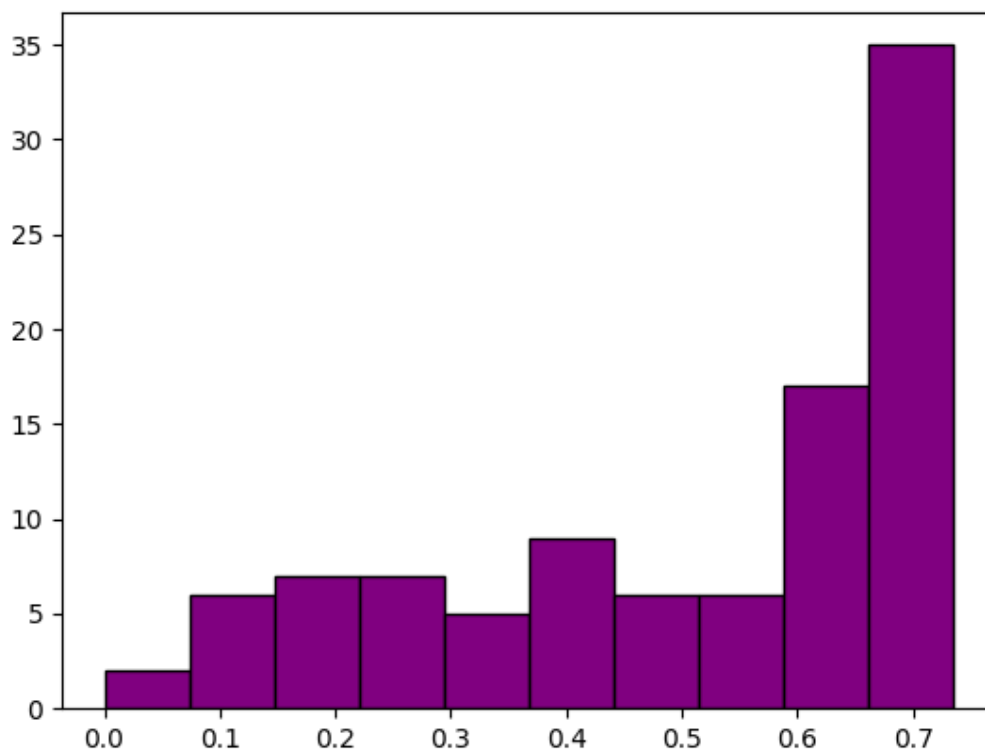
Побудувати гістограми розподілів:

- часу появи заданої події (перша, друга, n - та);
- інтервалу між подіями;
- появи рівно n - подій.

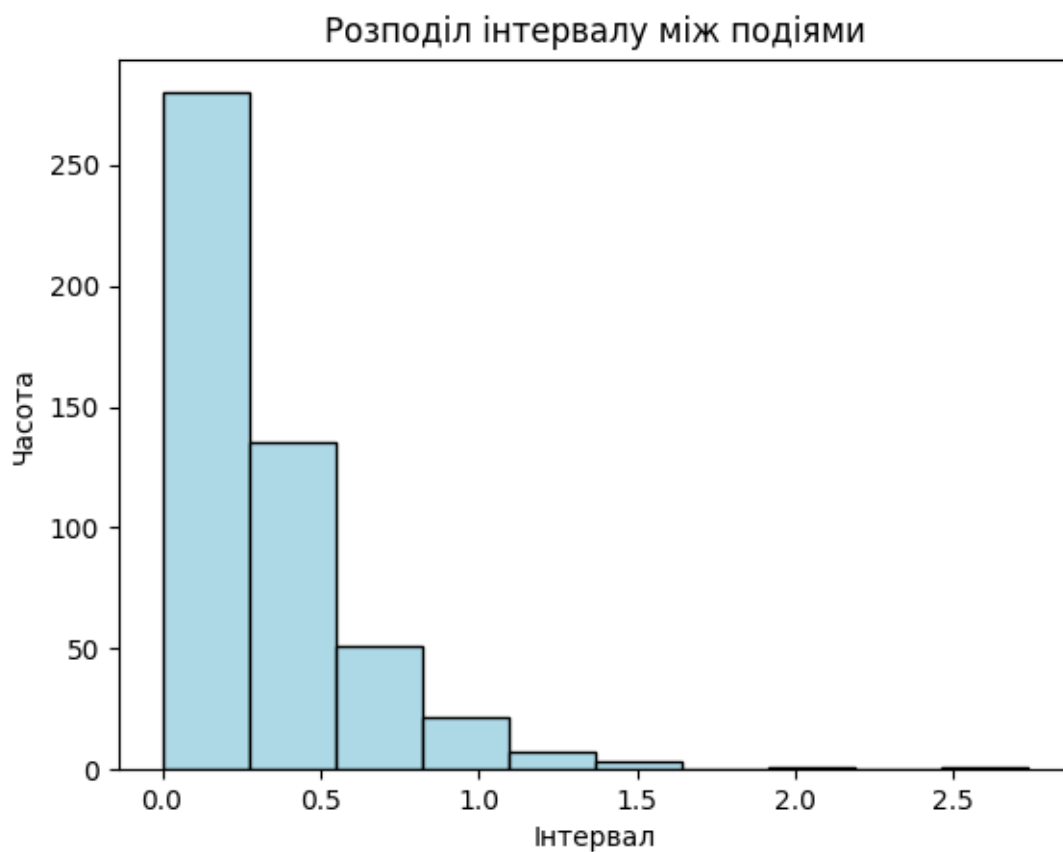
1.0



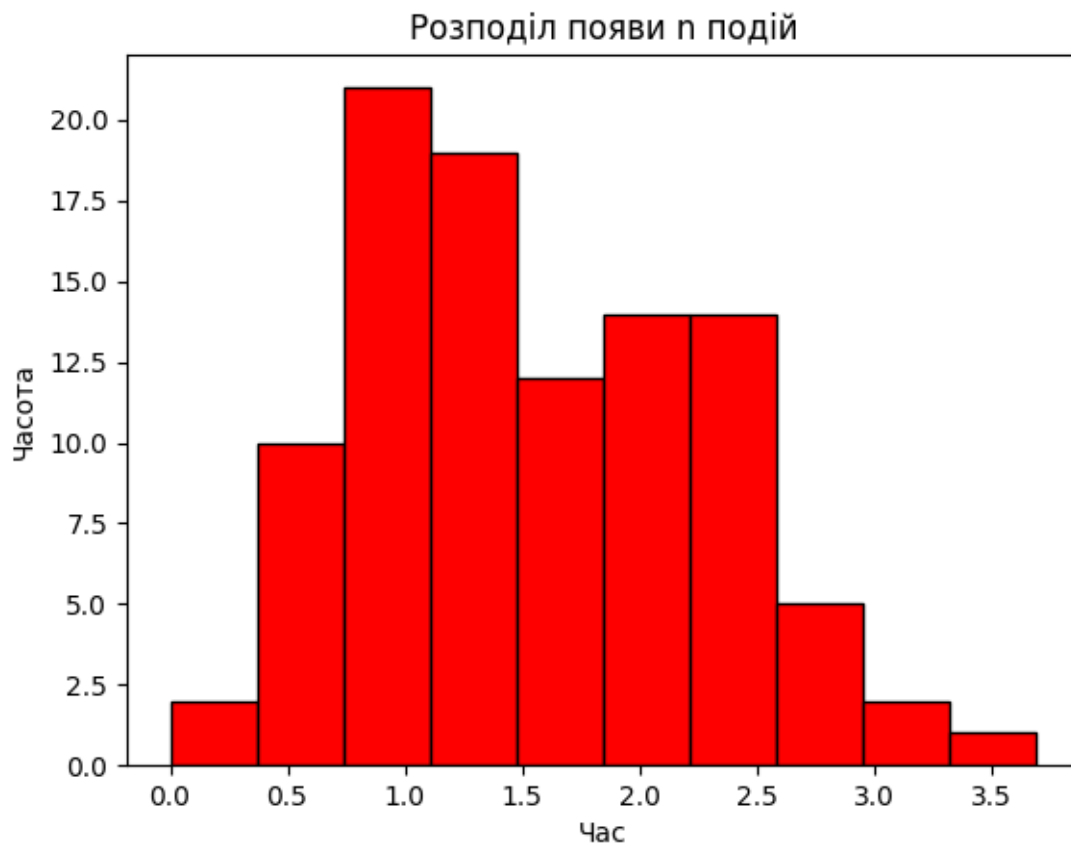
1.1 Час появи заданої події



1.2



1.3



Код програми:

```
#Task 1: Poisson process
from matplotlib import pyplot as plt
import math
import random
import statistics
import numpy as np
from colorsys import hls_to_rgb

def factorial(n):
    if n==0:
        return 1
    else:
        fact=1
        for i in range(1,n+2):
            fact=fact*i
        return(fact)

n=5
m=100
l=3
```

```

s=[]
t=[]
x_values=[]
y_values=[]
for i in range(n):
    s.append([])
    x_values.append([])
    y_values.append([])

for i in range(n):
    s[i].append(0)
    for j in range(1,m):
        s[i].append(-(1/l)*np.log(1-random.random()))

fig, ax=plt.subplots(figsize=(8,6))
hue=0/360
luminosity= 0.5
saturation=1
leg=[]
for i in range(n):
    x_prev=0
    for j in range(m-1):
        x=x_prev+s[i][j+1]
        x_values[i].extend([x_prev,x])
        y_values[i].extend([j,j])
        x_values[i].extend([x,x])
        y_values[i].extend([j,j+1])
        x_prev+=s[i][j+1]

for i in range(n):
    hue+=70/360
    color=hls_to_rgb(hue, luminosity, saturation)
    ax.plot(x_values[i],y_values[i], color=color, label=str(i+1)+"-та
подія")
    ax.legend()
plt.xlabel("Час")
plt.title("Реалізації Пуассонівського поцесу")
plt.show()

#_____1_____
k=2
p=[]

ktime=[]

```

```

for i in range(m):
    ksum=0
    lower_lim=(i-1)*k
    upper_lim=lower_lim+k
    for j in range(lower_lim, upper_lim):
        ksum+=math.exp(-l*s[k][i])*l*s[k][i]
    ktime.append(ksum)

plt.hist(ktime, color="purple", edgecolor="black")
plt.show()

# _____2_____
s1=[]
for i in range (n):
    s1.extend(s[i])

plt.hist(s1, color="lightblue", edgecolor="black")
plt.xlabel("Інтервал")
plt.ylabel("Часота")
plt.title("Розподіл інтервалу між подіями")
plt.show()

# _____3_____
mid=n/l
time=[]
for j in range(m):
    current=0
    for i in range(n):
        current+=s[i][j]
    time.append(current)

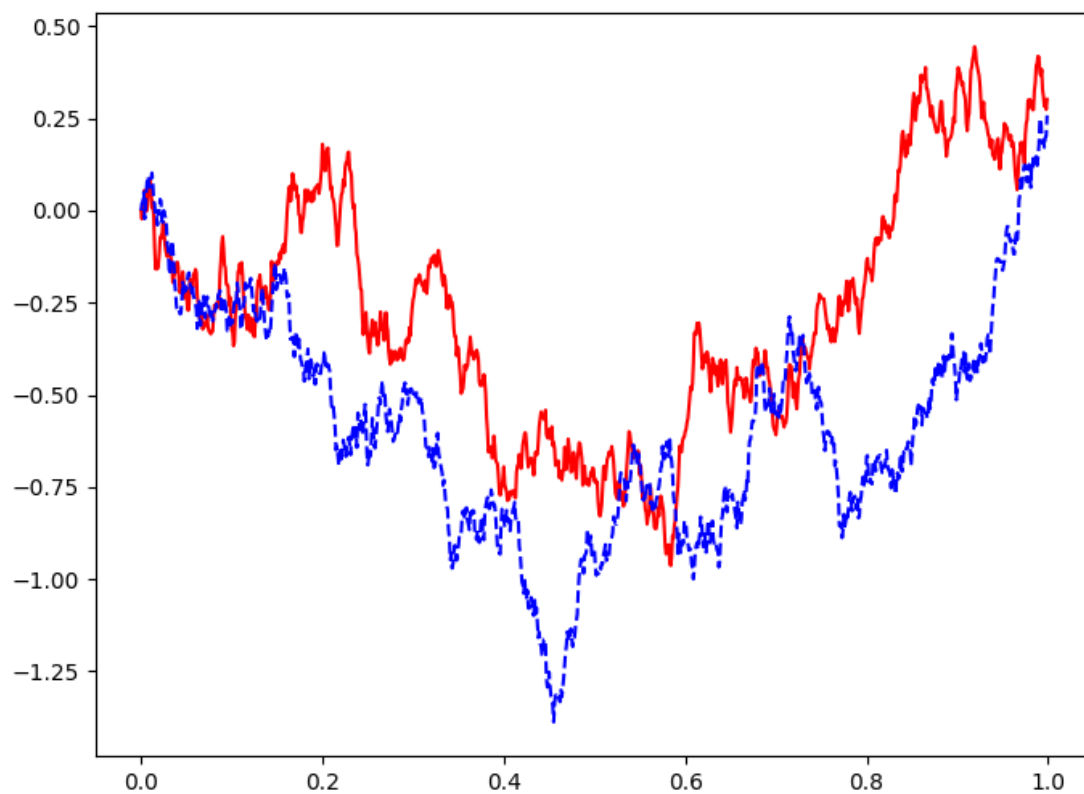
plt.xlabel("Час")
plt.ylabel("Часота")
plt.title("Розподіл появи n подій")
plt.hist(time, color="red", edgecolor="black")
plt.show()

```

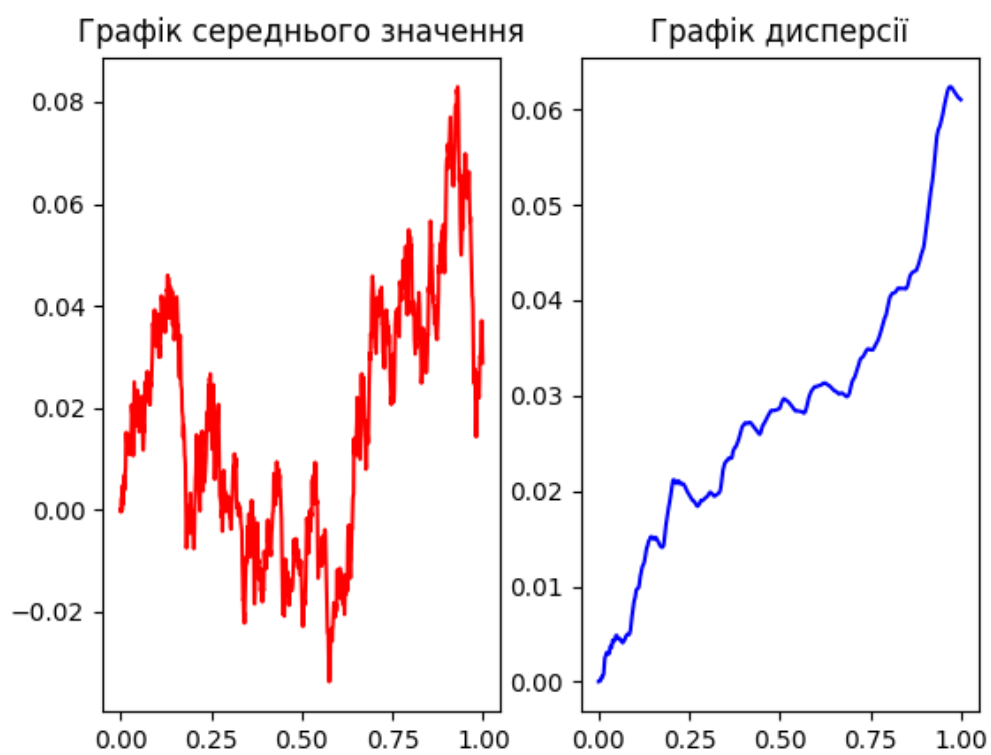
Завдання 2

1. Змодельовати неперервний вінерівський випадковий процес.
2. За реалізаціями (кількість реалізацій > 100) оцінити середнє значення та дисперсію.
3. Знайти емпіричний закон розподілу ймовірностей часу першого виходу вінерівського процесу за заданий рівень.

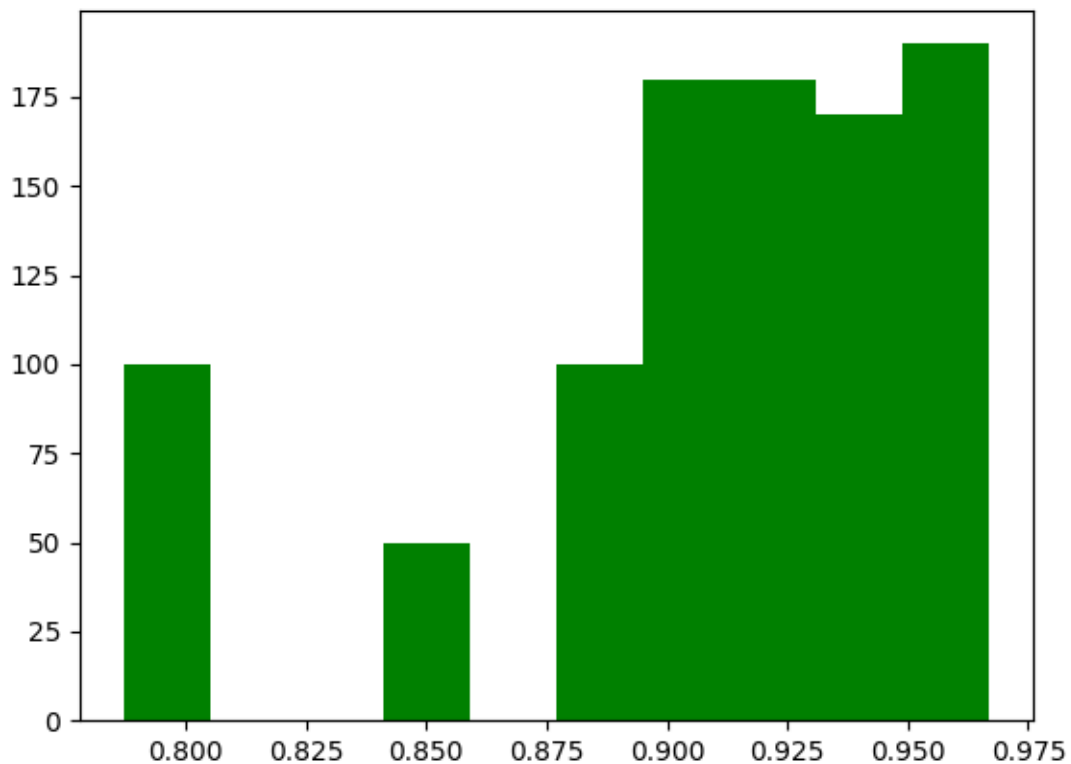
1.



2.



3. Якщо заданий рівень дорівнює 0,5:



Код програми:

```
#Task 2:Wiener process
#__1__
import math
import numpy as np
from matplotlib import pyplot as plt
from numpy.lib.function_base import average
m=1000
mu0=np.random.normal(0,1)
mu1=list(np.random.normal(0,1,m))
mu2=[]
mu2.append(mu1[0])
mu2.append(list(np.random.normal(0,1,m)))
mu2.append(list(np.random.normal(0,1,m)))

x=[]
w1=[]
w2=[]
for i in range(m):
    x.append(i/m)
```

```

for i in range(m):
    sum1=0
    sum2=0
    for j in range (1,m):
        sum1+=math.sin(j*math.pi*x[i])*mu1[j]/(j*math.pi)
        sum2+=math.sin(2*j*math.pi*x[i])*mu2[1][j]/(2*j*math.pi)+(1-
math.cos(2*j*math.pi*x[i]))*mu2[2][j]/(2*j*math.pi)
    w1_value=x[i]*mu0+math.sqrt(2)*sum1
    w2_value=x[i]*mu0+math.sqrt(2)*sum2
    w1.append(w1_value)
    w2.append(w2_value)

fig, ax=plt.subplots(figsize=(8,6))
ax.plot(x, w1, color="red")
ax.plot(x, w2, "b--")
plt.show()

w1=[]
w1_list=[]
#___2___
for j in range (10):
    mu0=np.random.normal(0,1)
    mu1=list(np.random.normal(0,1,m))
    for i in range(m):
        sum1=0
        sum2=0
        for j in range (1,m):
            sum1+=math.sin(j*math.pi*x[i])*mu1[j]/(j*math.pi)
        w1_value=x[i]*mu0+math.sqrt(2)*sum1
        w1.append(w1_value)
    w1_list.append(w1)

d=[]
disp=[]
avr=[]
for i in range (m):
    s=0
    for j in range(10):
        d.append(w1_list[j][i])
        s+=w1_list[j][i]
    avr.append(s/100)
    disp.append(np.var(d))
plt.subplot(1,2,1)
plt.plot(x, avr, color="red", label="Середнє значення")

```

```
plt.title("Графік середнього значення")
plt.subplot(1,2,2)
plt.plot(x, disp, color="blue", label="Диспесія")
plt.title("Графік дисперсії")
plt.show()
```

```
#____3____
a=float(input("Введіть бажаний рівень: "))
t=[]
for i in range(10):
    for j in range(m):
        if w1_list[i][j]>=a:
            index=j
            t.append(index/m)
        pass
plt.hist(t, color="green")
plt.show()
```