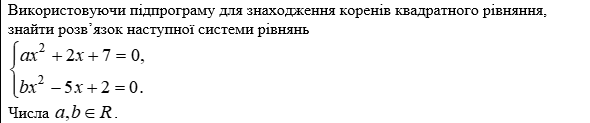
Lab 8

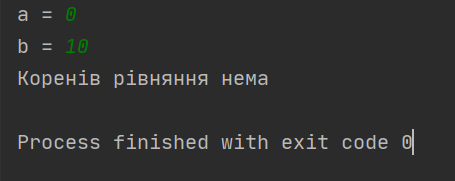
1.



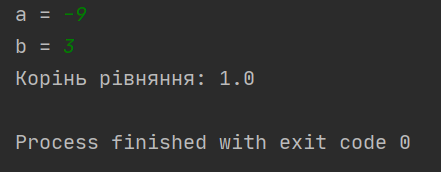
import math  
  
  
def square(a, b, c):  
 d = b \*\* 2 - 4 \* a \* c  
 if d >= 0:  
 x = [((-b + math.sqrt(d)) / (2 \* a)),   
 ((-b - math.sqrt(d)) / (2 \* a))]  
 else:  
 x = None  
 return x  
 pass  
  
  
a1 = float(input('a = '))  
b1 = float(input('b = '))  
x1\_2 = square(a1, 2, 7) if a1 != 0 else [-3.5] \* 2  
x3\_4 = square(b1, -5, 2) if b1 != 0 else [-2.5] \* 2  
if x3\_4 is None or x1\_2 is None:  
 total = 'Коренів рівняння нема'  
else:  
 x1 = x1\_2[0] in x3\_4  
 x2 = x1\_2[1] in x3\_4  
 if x1 and x2:  
 total = 'Корені рівняння: {0} та {1}'.format(x1\_2[0], x1\_2[1])  
 elif x1:  
 total = 'Корінь рівняння: {0}'.format(x1\_2[0])  
 elif x2:  
 total = 'Корінь рівняння: {0}'.format(x1\_2[1])  
 else:  
 total = 'Коренів рівняння нема'  
print(total)

Приклади:

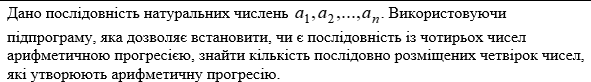
1)



2)



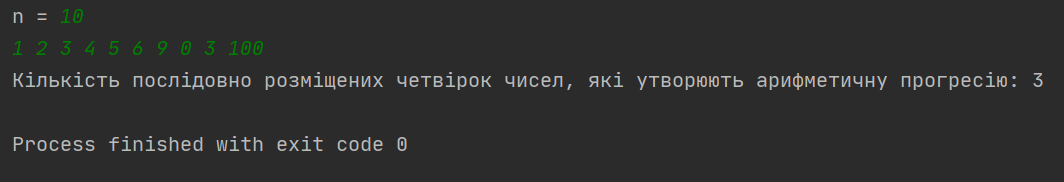
2.



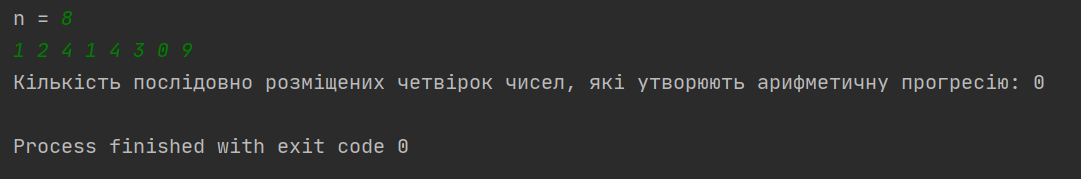
def arithmetic\_progression(b, j):  
 d1 = b[j] - b[j - 1]  
 d2 = b[j - 1] - b[j - 2]  
 d3 = b[j - 2] - b[j - 3]  
 return d1 == d2 == d3  
  
  
n = int(input('n = '))  
a = list(map(float, input().split()))  
total = 0  
for i in range(3, n):  
 total += arithmetic\_progression(a, i)  
print('Кількість послідовно розміщених четвірок чисел,'  
 ' які утворюють арифметичну прогресію: {0}'.format(total))

Приклади:

1)



2)



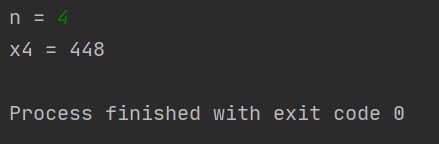
3.



def recursion(i):  
 if i == 0:  
 return 0  
 elif i == 1:  
 return 7  
 else:  
 return recursion(i-1) \* (1 + recursion(i-2))  
 pass  
  
  
n = int(input('n = '))  
print('х{0} = {1}'.format(n, recursion(n)))

Приклади:

1)



2)

