
Lab report 1 (Variant 2)

Student :

Herashchenko Serhii

Teacher :

Zhuravlova Zinaida Yuriyivna

14 вересня 2024 р.

3mict

1	Task 1	2
1.1	Condition	2
1.2	Solution code	2
1.3	Output	2
2	Task 2	3
2.1	Condition	3
2.2	Solution code	3
2.3	Output	3
3	Task 3	3
3.1	Condition	3
3.2	Test case	4
3.3	Solution code	4
3.4	Output	5
4	Task 4	5
4.1	Condition	5
4.2	Solution code	5
4.3	Input	5
4.4	Output	5

1 Task 1

1.1 Condition

У 1655 році англійський математик Джон Валліс (John Wallis) знайшов наступне подання числа π у вигляді нескінченного добутку

$$\frac{\pi}{2} = \frac{2}{1} \times \frac{2}{3} \times \frac{4}{3} \times \frac{4}{5} \times \frac{6}{5} \times \frac{6}{7} \times \frac{8}{7} \times \frac{8}{9} \times \frac{10}{9} \times \dots$$

Скільки множників у даному добутку потрібно взяти, щоб отримати значення числа π з точністю $\varepsilon = 10^{-4}$?

1.2 Solution code

```
import math as M

def pi_evaluater(eps):
    pi_result = 4
    num = 2
    den = 3
    numcount = 1
    dencounter = 0
    return_counter = 0
    pi_round = round(M.pi,eps)

    while round(pi_result,eps) != pi_round:
        pi_result *= (num/den)
        numcount += 1
        dencounter += 1
        if numcount == 2 :
            num += 2
            numcount = 0
        if dencounter == 2:
            den += 2
            dencounter = 0
        return_counter += 1
    return return_counter

eps = 4
result = pi_evaluater(eps)
print(result)
```

1.3 Output

27390

2 Task 2

2.1 Condition

В теорії чисел числами-близнюками називають пару простих чисел, різниця між якими дорівнює 2.

Вивести на екран усі пари чисел-близнюків, що не більше 100. $\varepsilon = 10^{-4}$?

2.2 Solution code

```
def primes_detector(bound):
    primes = [2]
    start = 1

    def is_prime(num):
        for i in range(2, num):
            if num % i == 0:
                return False
        return True

    for i in range(3, bound):
        if is_prime(i):
            primes.append(i)

    return primes

def answer(list):
    ans_list = []

    for i in range(len(list) - 1):
        if list[i + 1] - list[i] == 2:
            ans_list.append((list[i], list[i + 1]))

    return ans_list

print(answer(primes_detector(100)))
```

2.3 Output

[(3, 5), (5, 7), (11, 13), (17, 19), (29, 31), (41, 43), (59, 61), (71, 73)]

3 Task 3

3.1 Condition

У поїзді 15 вагонів, у кожному з яких 36 місць. Інформація про продані квитки міститься у матриці, номери строк якої відповідають номерам вагонів, а номери стовбців –

номерам місць, причому непарний номер позначає місце на нижній полці, парний – на верхній. Якщо квиток на те чи інше місце проданий, то відповідний елемент матриці дорівнює 1, якщо же місце вільне – 0. Знайти номер вагону, у якому залишилось найбільша кількість місць на нижніх полках.

3.2 Test case

```
1 0 0 1 0 0 0 0 1 1 0 1 0 1 0 1 0 0 1 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 0 0
1 0 0 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 1 1 0 0 0 1 1 0 0 0 1 1 1 1 0 1 0 0
1 0 0 1 0 0 0 0 1 1 0 1 0 0 0 1 1 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 0 0
1 1 1 1 0 0 0 0 1 1 0 1 0 1 0 1 0 0 1 0 0 0 0 1 1 0 1 0 0 1 0 1 0 1 1 0
1 0 0 0 0 1 0 0 0 1 1 1 0 1 0 1 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0
1 0 0 1 0 0 0 0 1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 1 0 1 0 0
0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 1 0 1 1 0 1 0 0
1 0 0 0 1 1 1 1 1 1 0 0 0 1 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0
1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 1 0 0
1 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0 1 0 1 1 0 1 0 0 0 0 1 0 1 0 0
1 0 1 1 0 0 0 0 1 1 0 1 0 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0
0 0 0 0 0 0 0 0 1 1 0 1 0 1 1 1 1 0 0 0 1 0 0 1 1 0 1 0 0 1 1 1 0 1 1 0
1 0 0 1 0 0 1 0 1 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 1 0 0 1 0 1 0 1 0 0
1 0 1 0 0 0 1 0 1 1 0 1 0 0 0 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 1 1 1 1 0 0
```

3.3 Solution code

```
file = open("Python lab 1\\tasks\\3_third_task\\test.txt")
carriage = []
rows = 15
cols = 36

data = file.read().split("\n")
for i in range(rows):
    row = data[i].strip()
    temp_row = row.split(" ")
    carriage.append([int(x) for x in temp_row])

file.close()

sold_seats = []

for row_index, row in enumerate(carriage):
    counter = 0
    for seat_index, seat in enumerate(row):
        if (seat_index + 1) % 2 != 0 and seat == 0:
            counter += 1
    sold_seats.append(counter)
```

```
counter = 0

max_seats = max(sold_seats)
print(sold_seats.index(max_seats)+1)
```

3.4 Output

14

4 Task 4

4.1 Condition

Дано файл input.txt, що містить деякий текст (переноси слів не допускаються). Видалити з вихідного тексту усі слова, що складаються з 3 і менше букв. Результат записати у файл output.txt.

4.2 Solution code

```
file = open("Python lab 1\\tasks\\4_fourth_task\\input.txt")

data = file.read().split("\n")

return_text = ""

for row in data:
    words = row.split(" ")
    for word in words:
        if len(word) > 3:
            return_text += word
            if words.index(word) != len(words) - 1:
                return_text += ' '
    if data.index(row) != len(data) - 1:
        return_text += "\n"
file.close()
file = open("Python lab 1\\tasks\\4_fourth_task\\output.txt", "w")
file.write(return_text)
file.close()
```

4.3 Input

Hello my name is Seryoja Sulfur does know that is an chemical element

4.4 Output

Hello name Seryoja Sulfur does know that chemical element