**Міністерство освіти і науки України**

**Національний університет «Запорізька Політехніка»**

Кафедра програмних засобів

**ЗВІТ**

з лабораторної роботи №2

з дисципліни «Спортивне програмування» на тему:

«Алгоритми довгої арифметики»

**Виконав:**

Студент групи КНТ-122 О. А. Онищенко

**Прийняли:**

Викладач: С. Д. Леощенко

2023

**Рекурентні послідовності**

**Мета роботи**

Вивчити основні можливості та принципи роботи із довгою арифметикою.

**Завдання до роботи**

Ознайомитися з основними теоретичними відомостями за темою роботи, використовуючи ці методичні вказівки, а також рекомендовану літературу.

Обрати та виконати дві задачі із запропонованого переліку.

Скласти програму обчислення точного значення 𝑛!, де 𝑛>12.

Скласти програму обчислення точного значення 𝑛^𝑛, де 𝑛>10.

Обчислити 100!+2^100.

Обчислити 100!−2^100.

Обчислити 7^123

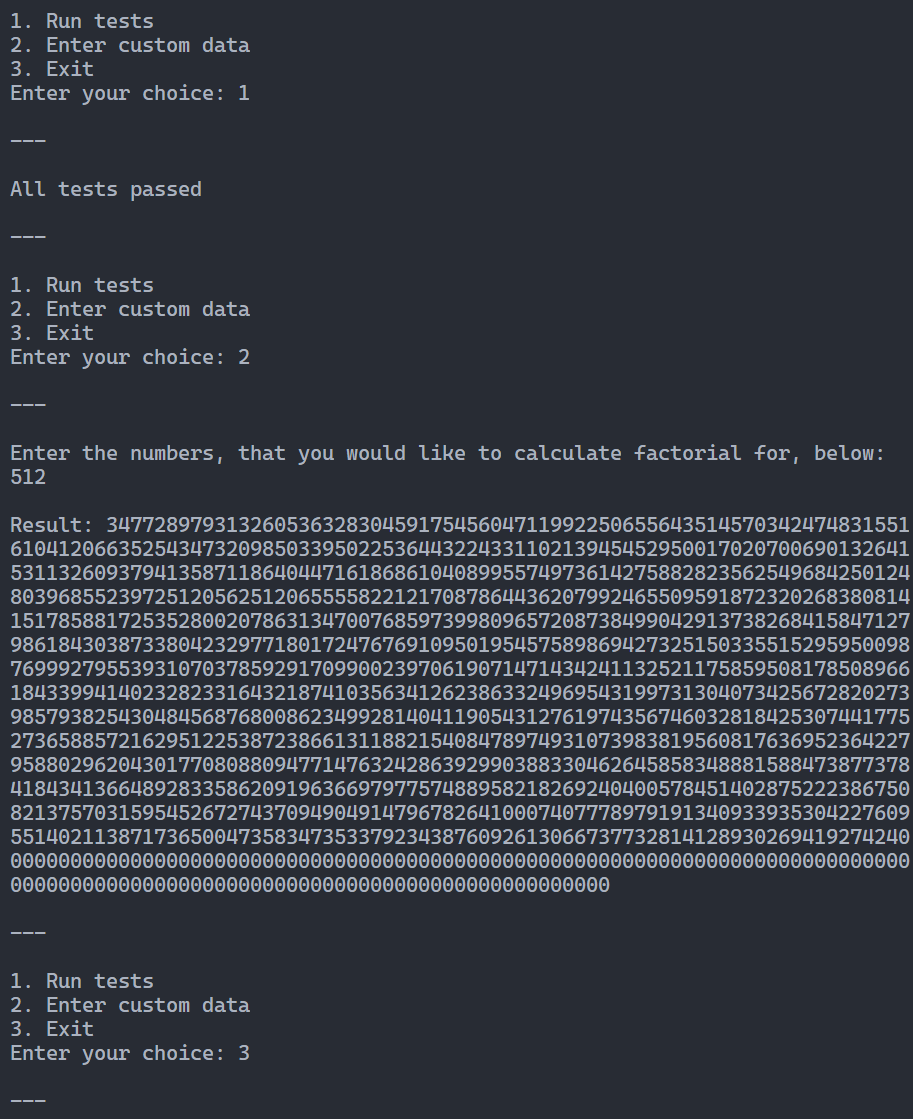
З'ясувати, яке з чисел 𝑎^𝑚 чи 𝑏^𝑛 – більше і на скільки (за умови, що 𝑎,𝑏≤40000, а 𝑚,𝑛≤10)

Скласти програму обчислення точного значення суми 1!+2!+3!+⋯+𝑛! при умові, що 𝑛>10

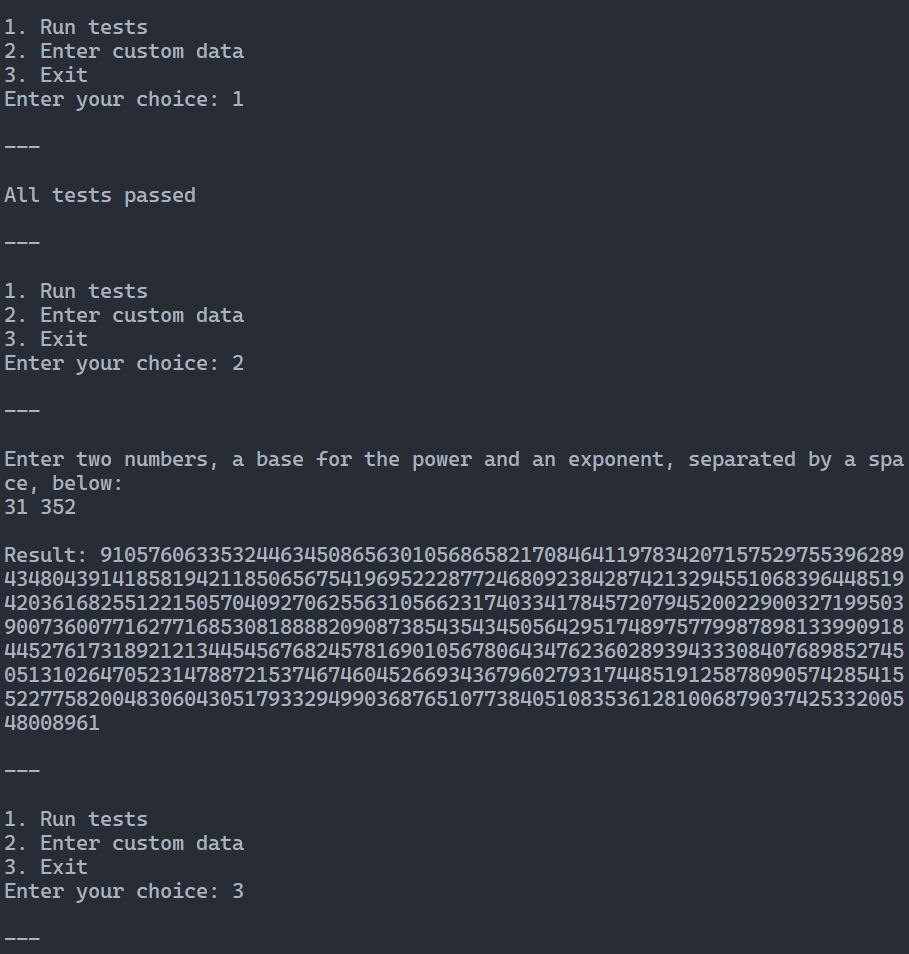
Обчислити точне значення суми 1^2+2^2+3^2+⋯+𝑛^2при умові, що 𝑛≥400

Обчислити точне значення суми 1^𝑛+2^𝑛+3^𝑛+⋯+𝑛^𝑛 при умові, що 𝑛≥10

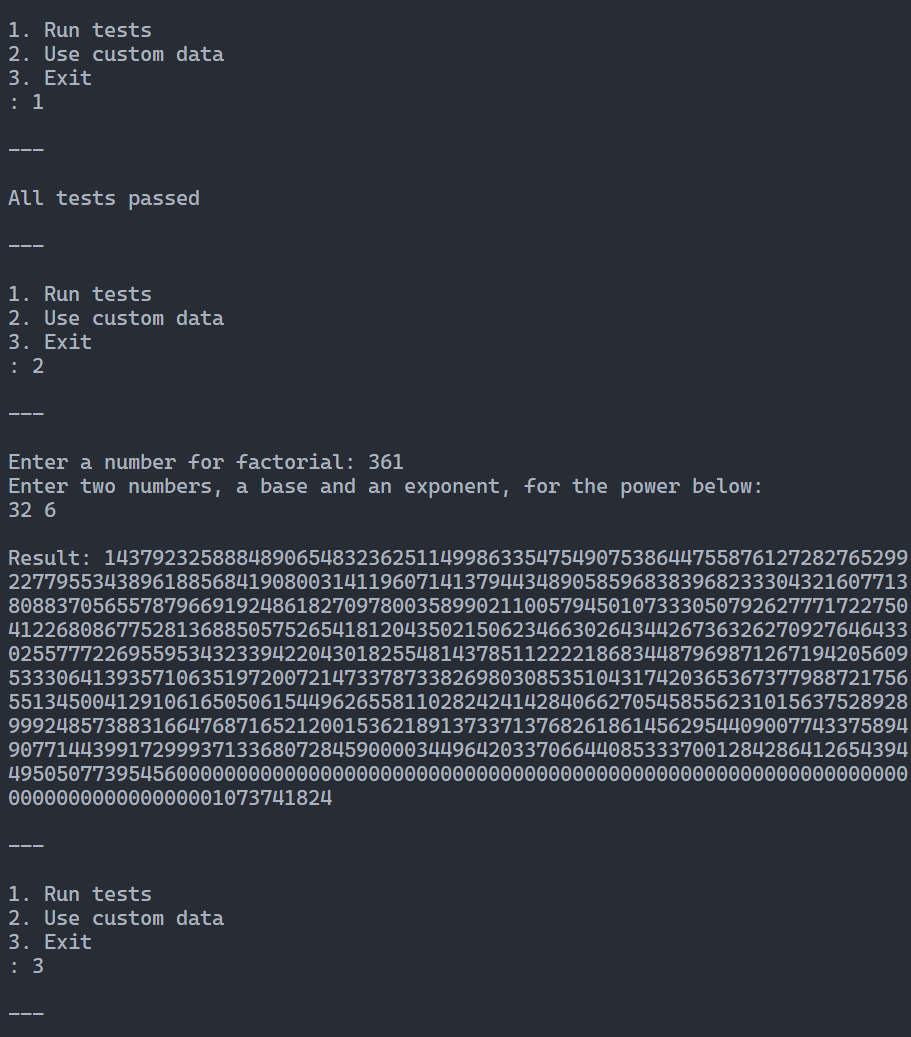
**Результати виконання**



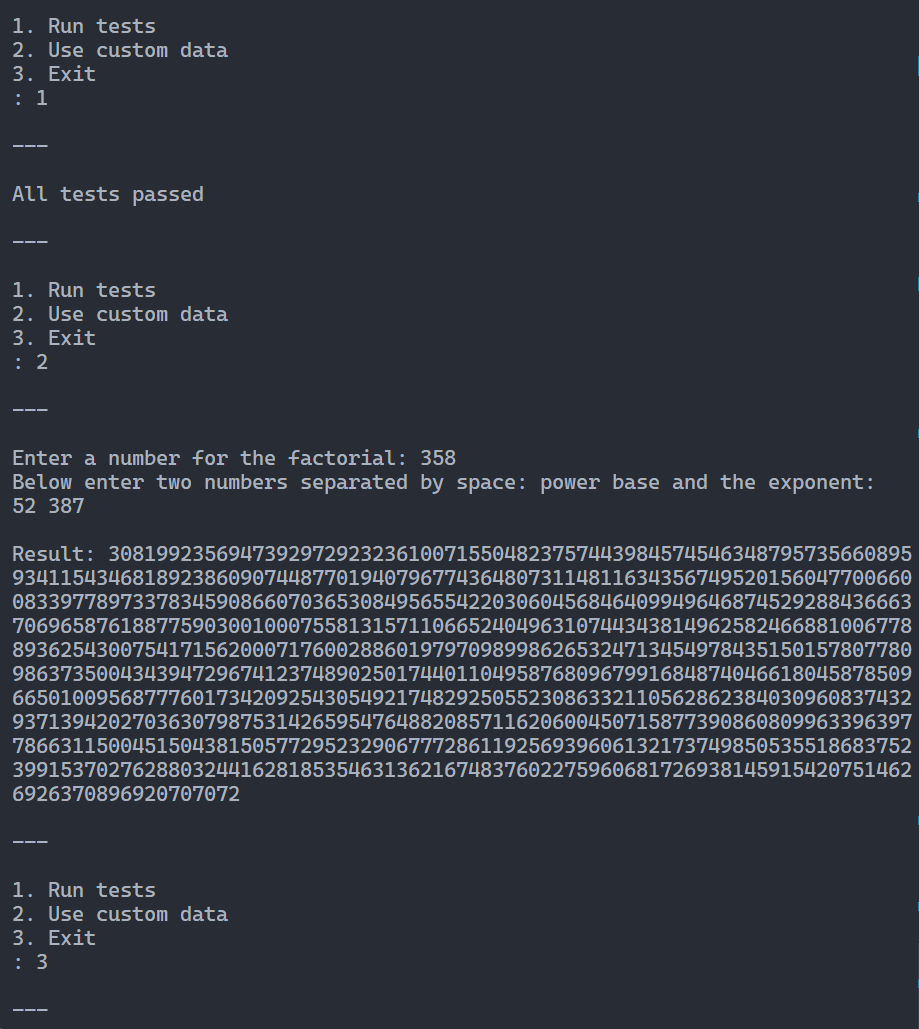
Результати виконання програми 1



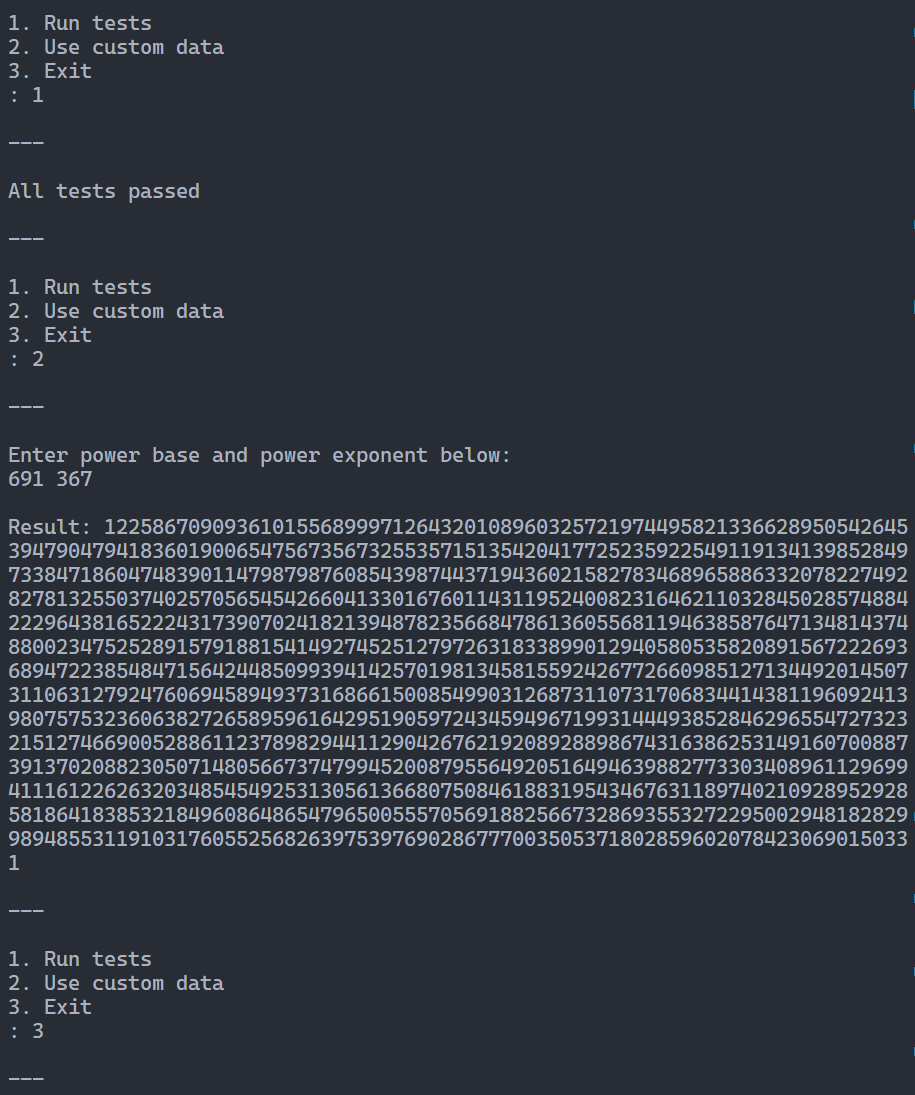
Результати виконання програми 2



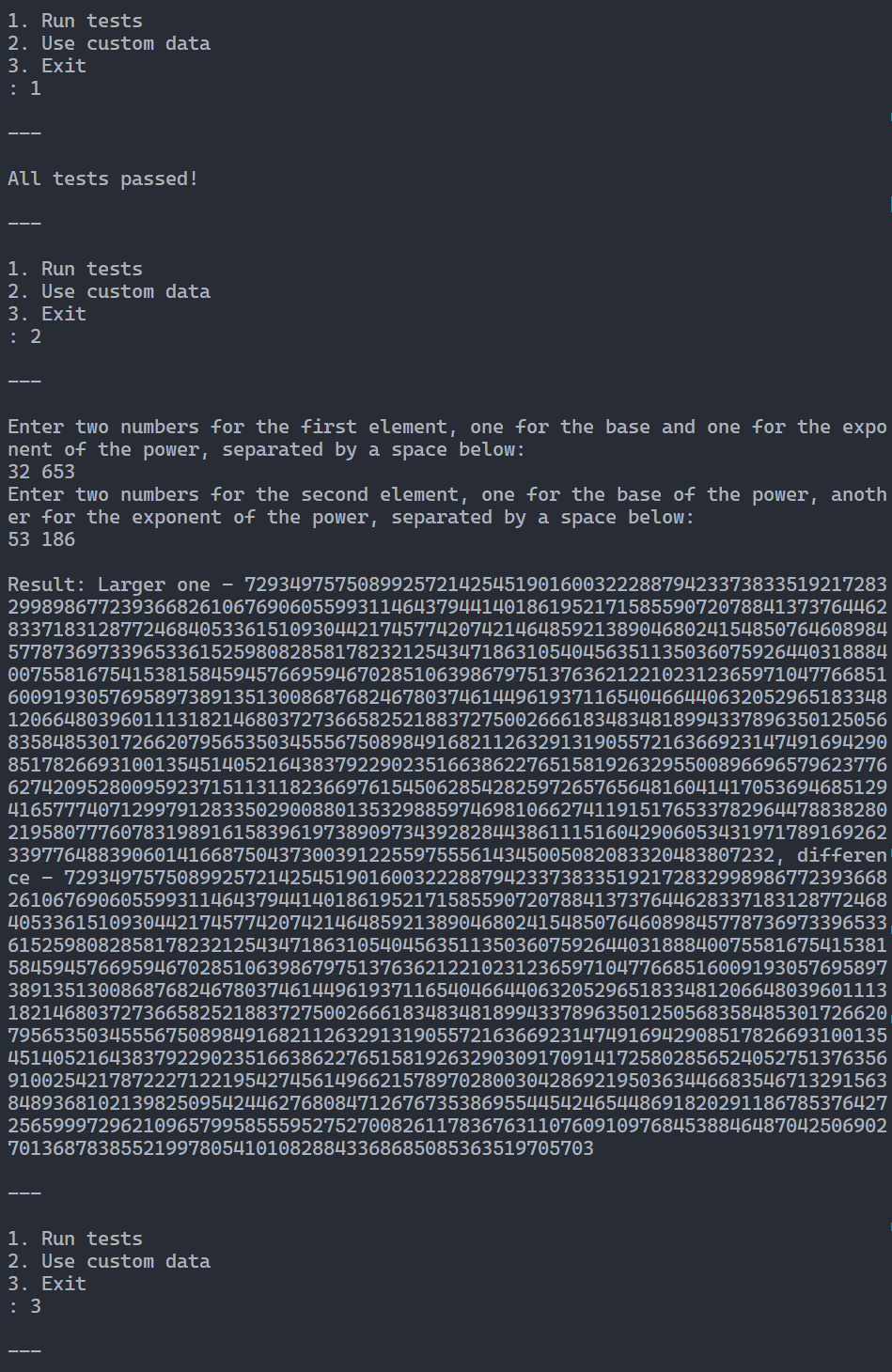
Результати виконання програми 3



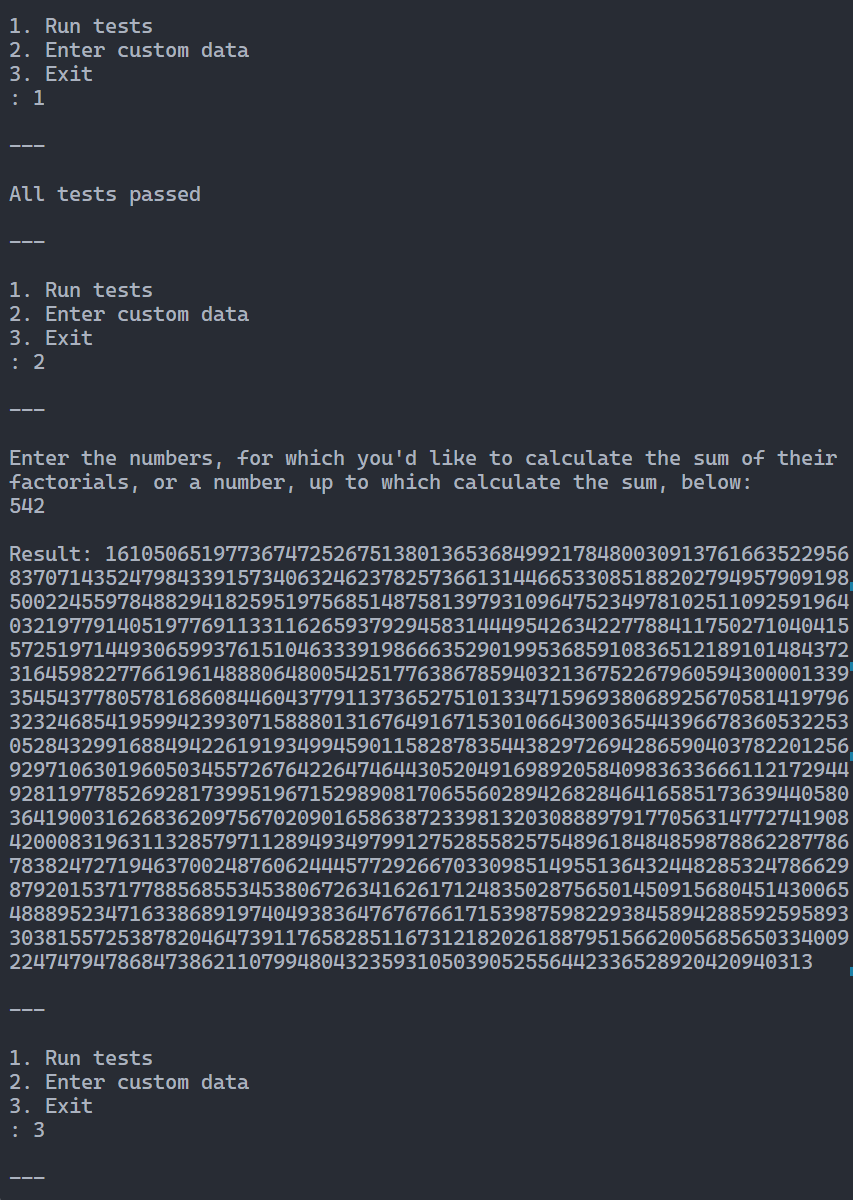
Результати виконання програми 4



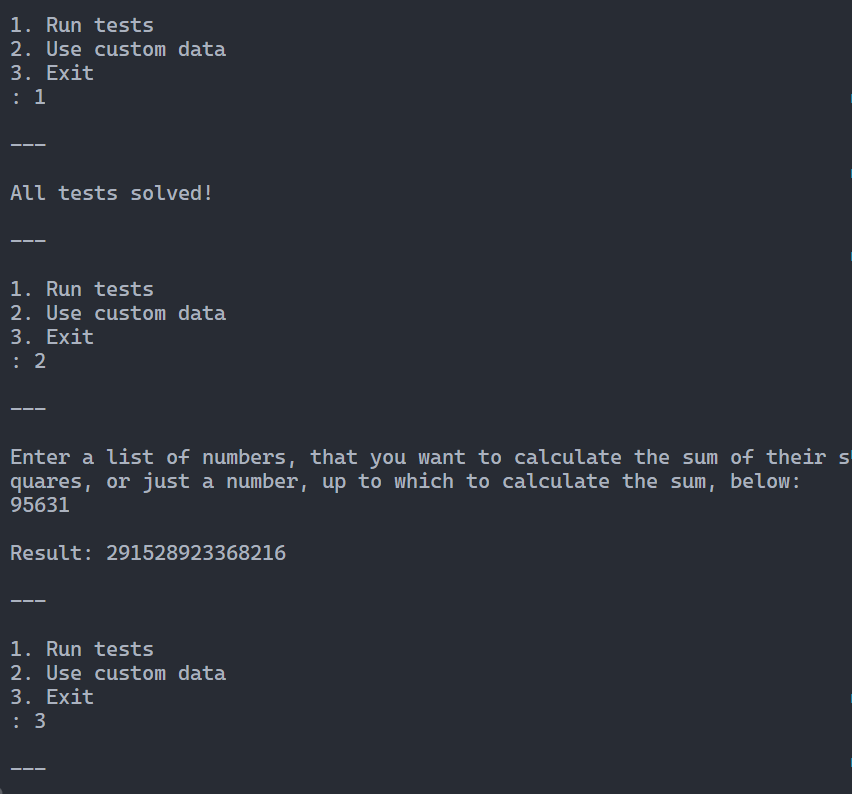
Результати виконання програми 5



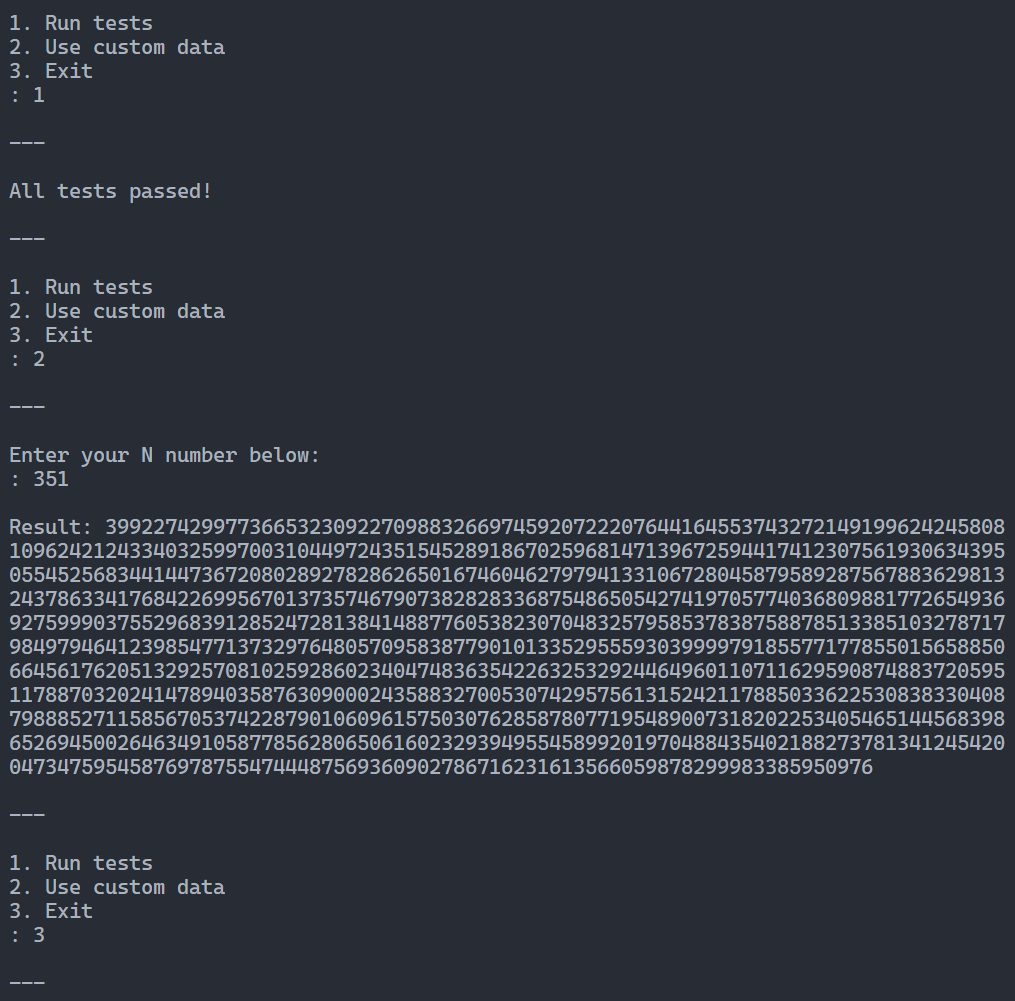
Результати виконання програми 6



Результати виконання програми 7



Результати виконання програми 8



Результати виконання програми 9

**Програмний код**

"""

Скласти програму обчислення точного значення 𝑛!, де 𝑛>12

"""

import math

def factorial(*n*):

    return 1 if *n* == 1 else *n* \* factorial(*n* - 1)

def tests():

    assert factorial(13) == math.factorial(13), "Test 1 failed"

    assert factorial(26) == math.factorial(26), "Test 2 failed"

    assert factorial(61) == math.factorial(61), "Test 3 failed"

    assert factorial(256) == math.factorial(256), "Test 4 failed"

    assert factorial(128) == math.factorial(128), "Test 5 failed"

    assert factorial(912) == math.factorial(912), "Test 6 failed"

    print("All tests passed")

def main():

    print()

    while True:

        print("1. Run tests")

        print("2. Enter custom data")

        print("3. Exit")

        choice = input("Enter your choice: ")

        print("\n---\n")

        if choice == "1":

            tests()

        elif choice == "2":

            print(

                f"Enter the numbers, that you would like to calculate factorial for, below:"

            )

            num = int(input())

            res = factorial(num)

            print(f"\nResult: {res}")

        else:

            break

        print("\n---\n")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

"""

Скласти програму для обчислення точного значення n\*\*k, де n>10

"""

def pow(*n*, *power*):

    res = 1

    for \_ in range(*power*):

        res \*= *n*

    return res

def tests():

    assert pow(13, 13) == 13\*\*13, "Test 1 failed"

    assert pow(51, 51) == 51\*\*51, "Test 2 failed"

    assert pow(318, 318) == 318\*\*318, "Test 3 failed"

    assert pow(916, 916) == 916\*\*916, "Test 4 failed"

    print("All tests passed")

def main():

    print()

    while True:

        print("1. Run tests")

        print("2. Enter custom data")

        print("3. Exit")

        choice = input("Enter your choice: ")

        print("\n---\n")

        if choice == "1":

            tests()

        elif choice == "2":

            print(f"Enter two numbers, a base for the power and an exponent, separated by a space, below:")

            num, power = list(map(int, input().split()))

            res = pow(num, power)

            print(f"\nResult: {res}")

        else:

            break

        print("\n---\n")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

"""

Обчислити 100! + 2^100

"""

import math

def solve(*factorialNumber*=100, *powerBase*=2, *powerExponent*=100):

    return factorial(*factorialNumber*) + pow(*powerBase*, *powerExponent*)

def factorial(*n*):

    return 1 if *n* == 1 else *n* \* factorial(*n* - 1)

def pow(*n*, *power*):

    return 1 if *power* == 0 else *n* \* pow(*n*, *power* - 1)

def tests():

    assert solve() == math.factorial(100) + 2\*\*100

    print("All tests passed")

def main():

    print()

    while True:

        print("1. Run tests")

        print("2. Use custom data")

        print("3. Exit")

        choice = int(input(": "))

        print(f"\n---\n")

        if choice == 1:

            tests()

        elif choice == 2:

            factorialNumber = int(input(f"Enter a number for factorial: "))

            print(f"Enter two numbers, a base and an exponent, for the power below:")

            powerBase, powerExponent = list(map(int, input().split()))

            res = solve(factorialNumber, powerBase, powerExponent)

            print(f"\nResult: {res}")

        else:

            break

        print("\n---\n")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

"""

Обчислити 100! − 2^100

"""

import math

def solve(*factorialNumber*=100, *powerBase*=2, *powerExponent*=100):

    return factorial(*factorialNumber*) - pow(*powerBase*, *powerExponent*)

def factorial(*n*):

    return 1 if *n* == 1 else *n* \* factorial(*n* - 1)

def pow(*base*, *exponent*):

    return 1 if *exponent* == 0 else *base* \* pow(*base*, *exponent* - 1)

def tests():

    assert solve() == math.factorial(100) - 2\*\*100

    print("All tests passed")

def main():

    print()

    while True:

        print("1. Run tests")

        print("2. Use custom data")

        print("3. Exit")

        choice = int(input(": "))

        print("\n---\n")

        if choice == 1:

            tests()

        elif choice == 2:

            factorialNumber = int(input(f"Enter a number for the factorial: "))

            print(

                "Below enter two numbers separated by space: power base and the exponent: "

            )

            powerBase, powerExponent = list(map(int, input().split()))

            res = solve(factorialNumber, powerBase, powerExponent)

            print(f"\nResult: {res}")

        else:

            break

        print("\n---\n")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

"""

Обчислити 7^123

"""

def solve(*powerBase*=7, *powerExponent*=123):

    return pow(*powerBase*, *powerExponent*)

def pow(*base*, *exponent*):

    return 1 if *exponent* == 0 else *base* \* pow(*base*, *exponent* - 1)

def tests():

    assert solve() == 7\*\*123

    print("All tests passed")

def main():

    print()

    while True:

        print("1. Run tests")

        print("2. Use custom data")

        print("3. Exit")

        choice = int(input(": "))

        print("\n---\n")

        if choice == 1:

            tests()

        elif choice == 2:

            print(f"Enter power base and power exponent below: ")

            powerBase, powerExponent = list(map(int, input().split()))

            res = solve(powerBase, powerExponent)

            print(f"\nResult: {res}")

        else:

            break

        print("\n---\n")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

"""

З'ясувати яке з чисел a\*\*m чи b\*\*n більше і на скільки. За умови, що a,b <= 40000, а m,n <= 10

"""

def solve(*a*, *b*):

    biggerOne = *a* if *a* > *b* else *b*

    difference = *b* - *a* if *b* > *a* else *a* - *b*

    return biggerOne, difference

def pow(*base*, *exponent*):

    return 1 if *exponent* == 0 else *base* \* pow(*base*, *exponent* - 1)

def tests():

    assert solve(pow(3, 2), pow(5, 3)) == (125, 116)

    assert solve(pow(7, 9), pow(9, 8)) == (43046721, 2693114)

    assert solve(pow(12, 3), pow(9, 16)) == (1853020188851841, 1853020188850113)

    assert solve(pow(24, 15), pow(17, 59)) == (

        3948992976476546055807962117305548095339102740462421587418915544041816753,

        3948992976476546055807962117305548095339102740462421082561632587995710129,

    )

    print("All tests passed!")

def main():

    print()

    while True:

        print("1. Run tests")

        print("2. Use custom data")

        print("3. Exit")

        choice = int(input(": "))

        print("\n---\n")

        if choice == 1:

            tests()

        elif choice == 2:

            print(

                "Enter two numbers for the first element, one for the base and one for the exponent of the power, separated by a space below: "

            )

            oneBase, oneExponent = list(map(int, input().split()))

            one = pow(oneBase, oneExponent)

            print(

                "Enter two numbers for the second element, one for the base of the power, another for the exponent of the power, separated by a space below:"

            )

            twoBase, twoExponent = list(map(int, input().split()))

            two = pow(twoBase, twoExponent)

            bigger, difference = solve(one, two)

            print(f"\nResult: Larger one - {bigger}, difference - {difference}")

        else:

            break

        print("\n---\n")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

"""

Скласти програму обчислення точного значення суми 1! + 2! + 3! + ⋯ + 𝑛! за умовою, що 𝑛 > 10

"""

import math

def solve(*nums*: [int]) -> int:

    res = 0

    for num in *nums*:

        calculatedFactorial = factorial(num)

        res += calculatedFactorial

    return res

def factorial(*n*):

    return 1 if *n* == 1 else *n* \* factorial(*n* - 1)

def tests():

    assert solve([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]) == math.factorial(

        1

    ) + math.factorial(2) + math.factorial(3) + math.factorial(4) + math.factorial(

        5

    ) + math.factorial(

        6

    ) + math.factorial(

        7

    ) + math.factorial(

        8

    ) + math.factorial(

        9

    ) + math.factorial(

        10

    ) + math.factorial(

        11

    ), "Test 1 failed"

    assert solve([12, 13, 14, 15]) == math.factorial(12) + math.factorial(

        13

    ) + math.factorial(14) + math.factorial(15), "Test 2 failed"

    assert solve([16, 17, 18, 19, 20]) == math.factorial(16) + math.factorial(

        17

    ) + math.factorial(18) + math.factorial(19) + math.factorial(20), "Test 3 failed"

    assert solve([2, 3, 9, 5]) == math.factorial(2) + math.factorial(

        3

    ) + math.factorial(9) + math.factorial(5)

    print("All tests passed")

def main():

    print()

    while True:

        print("1. Run tests")

        print("2. Enter custom data")

        print("3. Exit")

        choice = input(": ")

        print("\n---\n")

        if choice == "1":

            tests()

        elif choice == "2":

            print(

                f"Enter the numbers, for which you'd like to calculate the sum of their factorials, or a number, up to which calculate the sum, below:"

            )

            nums = list(map(int, input().split()))

            res = solve(nums) if len(nums) != 1 else solve(range(1, nums[0] + 1))

            print(f"\nResult: {res}")

        else:

            break

        print("\n---\n")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

"""

Обчислити точне значення суми 1\*\*2 + 2\*\*2 + 3\*\*2 + ... + n\*\*2 при умові, що n >= 400

"""

def solve(*nums*: [int]) -> int:

    res = 0

    for num in *nums*:

        calculatedSquare = num \* num

        res += calculatedSquare

    return res

def tests():

    assert solve([1, 2, 3, 4, 5]) == 1\*\*2 + 2\*\*2 + 3\*\*2 + 4\*\*2 + 5\*\*2

    assert solve([3, 9, 5, 4, 6]) == 3\*\*2 + 9\*\*2 + 5\*\*2 + 4\*\*2 + 6\*\*2

    assert solve([11, 64, 32, 16]) == 11\*\*2 + 64\*\*2 + 32\*\*2 + 16\*\*2

    print("All tests solved!")

def main():

    print()

    while True:

        print("1. Run tests")

        print("2. Use custom data")

        print("3. Exit")

        choice = int(input(": "))

        print("\n---\n")

        if choice == 1:

            tests()

        elif choice == 2:

            print(

                "Enter a list of numbers, that you want to calculate the sum of their squares, or just a number, up to which to calculate the sum, below:"

            )

            nums = list(map(int, input().split()))

            res = solve(nums) if len(nums) != 1 else solve(range(1, nums[0] + 1))

            print(f"\nResult: {res}")

        else:

            break

        print("\n---\n")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

"""

Обчислити точне значення суми 1\*\*n + 2\*\*n + 3\*\*n + ... + n\*\*n за умови, що n >= 10

"""

def solve(*n*: int) -> int:

    res = 0

    for i in range(1, *n* + 1):

        tmp = int(pow(i, *n*))

        res += tmp

    return res

def pow(*base*, *exponent*):

    return 1 if *exponent* == 0 else *base* \* pow(*base*, *exponent* - 1)

def tests():

    assert solve(5) == 1\*\*5 + 2\*\*5 + 3\*\*5 + 4\*\*5 + 5\*\*5

    assert solve(3) == 1\*\*3 + 2\*\*3 + 3\*\*3

    assert (

        solve(15)

        == 1\*\*15

        + 2\*\*15

        + 3\*\*15

        + 4\*\*15

        + 5\*\*15

        + 6\*\*15

        + 7\*\*15

        + 8\*\*15

        + 9\*\*15

        + 10\*\*15

        + 11\*\*15

        + 12\*\*15

        + 13\*\*15

        + 14\*\*15

        + 15\*\*15

    )

    print("All tests passed!")

def main():

    print()

    while True:

        print("1. Run tests")

        print("2. Use custom data")

        print("3. Exit")

        choice = int(input(": "))

        print("\n---\n")

        if choice == 1:

            tests()

        elif choice == 2:

            print("Enter your N number below:")

            n = int(input(": "))

            res = solve(n)

            print(f"\nResult: {res}")

        else:

            break

        print("\n---\n")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Висновки**

Таким чином, ми вивчили основні можливості та принципи роботи із довгою арифметикою.