

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»

Виконала:

студентка групи КНТ – 132

Мілена Кочева

Прийняв:

Старший викладач

Сергій ЛЕОЩЕНКО

2023

Мета роботи: вивчити основні можливості та принципи роботи із довгою арифметикою.

Хід роботи:

Завдання №1

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

Код:

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

n = int(input("Введіть значення n: "))

if n <= 12:
    print("n повинно бути більше 12.")
else:
    result = factorial(n)
    print(f"{n}! = {result}")
```

Скріншот:

```
"C:\Program Files\Python3115\python.exe" D:\1_НУЗП\Спортивне\Lab_2\Task_1\main.py
Введіть значення n: 15
15! = 1307674368000

Process finished with exit code 0
```

Завдання №2

3.2.1 Обчислити $100! + 2^{100}$.

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

def power(base, exponent):
```

```
result = 1

for _ in range(exponent):
    result *= base

return result

factorial_100 = factorial(100)
power_100 = power(2, 100)

result = factorial_100 + power_100

print(f"Результат: {result}")
```

Скріншот:

```
"C:\Program Files\Python3115\python.exe" D:\1_НУЗП\Спортивне\Lab_2\Task_2\main.py
Результат: 93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185212184514600228229401496703205376
Process finished with exit code 0
```

Висновок: вивчила основні можливості та принципи роботи із довгою арифметикою.

Контрольні питання:

1) Які існують моделі представлення чисел довгої арифметики (це загальне питання)?

Рядкове представлення, масив цифр, бібліотеки довгої арифметики, бінарні формати

2) Чому в Python (на відміну від C++, наприклад) не виникає проблема довгої арифметики?

Через наявність вбудованих бібліотек, які надають підтримку для роботи з великими числами. Наприклад, `decimal`, яка надає точні обчислення з плаваючою комою, також модуль `math`, який має підтримку довгих цілих чисел. В Python числа з обмеженою точністю, як цілі числа або числа з плаваючою комою, автоматично перетворюються на числа довгої арифметики, якщо результат операції виходить за межі обмежень стандартних типів даних. Наприклад, в C++ результат додавання двох цілих чисел може бути обмежений розміром типу даних, наприклад, `int` або `long`. Якщо сума перевищує максимальне значення цього типу, виникає переповнення. У Python ж такі обмеження відсутні, і використовуються підтримувані бібліотеки для автоматичного переходу до довгої арифметики.

