

Міністерство освіти і науки України  
Національний університет «Запорізька політехніка»

кафедра програмних засобів

Звіт  
з лабораторної роботи №1  
з дисципліни «Спортивне програмування»  
на тему : «**Рекурентні послідовності**»

**Виконала:**

студентка групи КНТ – 132

Мілена Кочева

**Прийняв:**

Старший викладач

Сергій ЛЕОЩЕНКО

2023

**Мета роботи:** вивчити основні можливості та принципи роботи з мовою рекурентні послідовності та співвідношення.

**Хід роботи:**

*Завдання №1*

3.2.2 У покупця є  $N$  монет гідністю  $H(1), \dots, H(n)$ . У продавця є  $m$  монет номіналом  $B(1), \dots, B(n)$ . Чи може купити покупець річ вартості  $S$  так, щоб у продавця знайшлася точна сума (якщо вона необхідна).

**Код:**

```
def hasExactChange(coinsBuyer, coinsSeller, targetAmount):  
    if targetAmount == 0:  
        return True  
  
    if targetAmount < 0 or not coinsBuyer:  
        return False  
  
    currentCoin = coinsBuyer[0]  
    remainingCoins = coinsBuyer[1:]  
  
    if hasExactChange(coinsSeller, coinsSeller, targetAmount -  
currentCoin):  
        return True  
  
    if hasExactChange(remainingCoins, coinsSeller, targetAmount):  
        return True  
  
    return False  
  
buyerCoins = [1, 2, 5]  
sellerCoins = [1, 2, 5]  
targetPrice = 3
```

```
if hasExactChange(buyerCoins, sellerCoins, targetPrice):  
    print("Покупець може купити річ з точною задачею.")  
else:  
    print("Покупець не може купити річ з точною задачею.")
```

## Скріншот:

```
"C:\Program Files\Python3115\python.exe" D:\1_НУЗП\Спортивне\Lab_1\main.py  
Покупець може купити річ з точною задачею.  
  
Process finished with exit code 0
```

---

### Завдання №2

3.2.7 У заданій послідовності цілих чисел знайти максимально довгу підпослідовність чисел таку, що кожен наступний елемент підпослідовності ділився націло на попередній.

### Код:

```
def findLongestDivisibleSubsequence(sequence):  
    n = len(sequence)  
    dp = [1] * n  
    maxLength = 1  
  
    for i in range(1, n):  
        for j in range(i):  
            if sequence[i] % sequence[j] == 0 and dp[j] + 1 > dp[i]:  
                dp[i] = dp[j] + 1  
                maxLength = max(maxLength, dp[i])  
  
    subsequence = []  
    lastElement = -1  
  
    for i in range(n - 1, -1, -1):
```

```

        if dp[i] == maxLength and (lastElement == -1 or lastElement %
sequence[i] == 0):

            subsequence.append(sequence[i])

            maxLength -= 1

            lastElement = sequence[i]

subsequence.reverse()

return subsequence

sequence = [2, 3, 4, 5, 8, 9, 16, 32, 64, 128, 256, 512]
result = findLongestDivisibleSubsequence(sequence)
print("Максимально довга підпоследовність:", result)

```

### Скріншот:

```

"C:\Program Files\Python3115\python.exe" D:\1_НУЗП\Спортивне\Lab_1\Task_2\main.py
Максимально довга підпоследовність: [2, 4, 8, 16, 32, 64, 128, 256, 512]

Process finished with exit code 0

```

### Завдання №3

3.2.8 Звести число  $a$  в натуральну ступінь  $n$  за якомога меншу кількість множень.

### Код:

```

def fastPower(a, n):

    if n == 0:

        return 1

    elif n % 2 == 0:

        halfPower = fastPower(a, n // 2)

        return halfPower * halfPower

    else:

        halfPower = fastPower(a, (n - 1) // 2)

        return halfPower * halfPower * a

```

```
a = 3
n = 10
result = fastPower(a, n)
print(f"{a} в ступені {n} дорівнює {result}")
```

### Скріншот:

```
"C:\Program Files\Python3115\python.exe" D:\1_НУЗП\Спортивне\Lab_1\Task_3\main.py
3 в ступені 10 дорівнює 59049

Process finished with exit code 0
```

---

**Висновки:** вивчила основні можливості та принципи роботи з мовою рекурентні послідовності та співвідношення.

### Контрольні питання:

1. Які механізми мови програмування та підходи Ви використали для розв'язання задач?

У першому завданні проблема розбивається на менші підзадачі, що обробляються рекурсивно. Кожна підзадача розв'язується окремо, а результати зворотно об'єднуються для визначення загального результату.

У другому завданні використовується підхід динамічного програмування, де результати попередніх обчислень зберігаються в масиві, що дозволяє уникнути повторних обчислень і покращити ефективність алгоритму.

У третьому завданні основна ідея алгоритму полягає у використанні рекурсивного підходу та властивостей ступеневої функції

2. Як у Python вирішується проблема динамічної пам'яті?

За допомогою автоматичного управління пам'яттю. Python використовує механізм підрахунку посилань та зборки сміття для автоматичного визначення, коли об'єкт вже не використовується і може бути звільнений з пам'яті