# Міністерство освіти і науки України Державний університет "Житомирська політехніка"

Кафедра інженерії програмного забезпечення

Група: ВТ-21-1[1]

Програмування мовою Python

Лабораторна робота № 2

«Розгалуження та цикли»

Виконав: Бабушко А. С.

Прийняв: Морозов Д. С.

					«Житомирська політехніка».21. <mark>121.01</mark> .000–Лр2			000–Лр2
3мн.	Арк.	№ докум.	Підпис	Дата				,
Розр	<b>00</b> б.	Бабушко А.С.				Літ.	Арк.	Аркушів
Пере	евір.	Морозов Д. С.			Звіт з		1	00
Кері	вник				лабораторної роботи			
Н. кс	онтр.				лаоораторног росоти	ΦΙΚΤ	ФІКТ Гр. ВТ-21-	

Затверд.

**Mema роботи:** познайомитися із структурою розгалуження (if, ifelse, if-elif-else). Ознайомитися з циклічними конструкціями і їх використанням в мові Python. Навчитися працювати з числами і рядками використовуючи дані структури.

### Хід роботи:

### Завдання на лабораторну роботу:

Завдання 1. Дано три цілих числа. Вибрати з них ті, які належать інтервалу [1,3].

### Лістинг програми:

```
""" Lab 2. Python. Andrii Babushko. Repository: https://github.com/AndriiBabushko/Python
"""
import math

# task 1
def task_1_interval_int_list(some_int_list):
    some_result_list = []
    for index in range(len(some_int_list)):
        if 1 <= some_int_list[index] <= 3:
            some_result_list.append(some_int_list[index])

    return some_result_list

print('\nTASK 1!!!')
task_1_list = [1, 2, 5, 4, 10, 15, 3]
print(f'Task 1 list: {task_1_list}')
print(f'List of elems that in interval [1; 3]: {task_1_interval_int_list(task_1_list)}')</pre>
```

### Результат програми:

```
Run: main ×

C:\Users\A\PycharmProjects\pythonProject\Universi

TASK 1!!!

Task 1 list: [1, 2, 5, 4, 10, 15, 3]

List of elems that in interval [1; 3]: [1, 2, 3]
```

		Бабушко А.С.		
		Морозов Д.С.		
Змн.	Арк.	№ докум.	Підпис	Дата

Завдання 2. Дано номер року (позитивне ціле число). Визначити кількість днів в цьому році, враховуючи, що звичайний рік нараховує 365 днів, а високосний - 366 днів. Високосним вважається рік, ділиться на 4, за винятком тих років, які діляться на 100 і не діляться на 400 (наприклад, роки 300 1300 і 1900 не є високосними, а 1200 і 2000 - є).

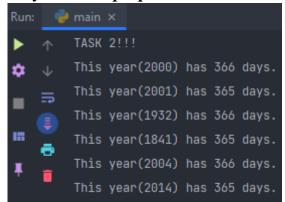
### Лістинг програми:

```
# task 2
def task_2_year_days(year):
    if is_leap_year(year):
        print(f'This year({year}) has 366 days.')
    else:
        print(f'This year({year}) has 365 days.')

def is_leap_year(year):
    if year % 400 == 0 or (year % 4 == 0 and year % 100 != 0):
        return True
    return False

print('\nTASK 2!!!')
task_2_year = [2000, 2001, 1932, 1841, 2004, 2014]
for i in range(len(task_2_year)):
    task_2_year_days(task_2_year[i])
```

#### Результат програми:



Завдання 3. Написати програму обчислення вартості покупки з урахуванням знижки. Знижка в 3% надається в тому випадку, якщо сума покупки більше 500 грн., В 5% - якщо сума більше 1000 грн.

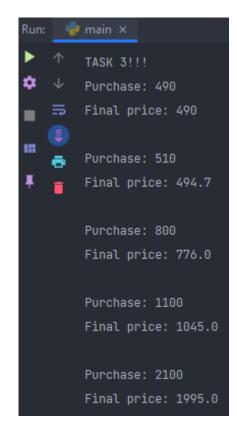
# Лістинг програми:

```
# task 3
def task_3_purchase_discount(purchase):
    if 500 <= purchase < 1000:
        return purchase - purchase * 0.03
    if purchase >= 1000:
        return purchase - purchase * 0.05
    return purchase

print('\nTASK 3!!!')
task_3_purchases = [490, 510, 800, 1100, 2100]
for i in range(len(task_3_purchases)):
    print(f'Purchase: {task_3_purchases[i]}\nFinal price:
{task_3_purchase_discount(task_3_purchases[i]}\n')
```

## Результат програми:

		Бабушко А.С.		
		Морозов Д.С.		
Змн.	Арк.	№ докум.	Підпис	Дата



Завдання 4. Знайти косинус мінімального з 4 заданих чисел.

### Лістинг програми:

### Результат програми:

Завдання 5. Вивести на екран синус максимального з 3 заданих чисел.

```
# task 5
def task_5_max_sin(angles):
    return math.sin(max(angles))

print('\nTASK 5!!!')
```

		Бабушко А.С.		
		Морозов Д.С.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
print(f'Angles list: {angles_list}')
print(f'Max sin in radians from angles list: {task_5_max_sin(angles_list)}')
```

Завдання 6. Складіть програму підрахунку площі рівнобедреного трикутника. Якщо площа трикутника парна, розділити її на 2, в іншому випадку вивести повідомлення «Не можу ділити на 2!»

### Лістинг програми:

```
# task 6
def task_6_isosceles_triangle(sides):
    s = (sides[0] * sides[1]) / 2
    if s % 2 == 0:
        s /= 2
        print(f'The area of the triangle: {s}')
    else:
        print(f"The S({s}), I can\'t divide by 2!")

print('\nTASK 6!!!')
left_right_sides = 1
base_side = 2
for x in range(1, 13, 3):
    task_6_isosceles_triangle([left_right_sides + x, base_side + x])
```

### Результат програми:

```
Run: main ×

TASK 6!!!

The S(3.0), I can't divide by 2!

The S(15.0), I can't divide by 2!

The area of the triangle: 18.0

The area of the triangle: 33.0
```

Завдання 7. Скласти програму, яка по даному числу (1-12) виводить назву відповідного йому місяця англійською мовою.

# Лістинг програми:

		Бабушко А.С.		
		Морозов Д.С.		
Змн.	Арк.	№ докум.	Підпис	Дата

Арк.

```
for i in range(0, 14):
Результат програми:
        TASK 7!!!
        0 number of month isn't in range [1, 12]!
       1 month is January
        2 month is February
         3 month is March
        4 month is April
         5 month is May
         6 month is June
         7 month is July
         8 month is August
         9 month is September
         10 month is October
         11 month is November
         12 month is December
         13 number of month isn't in range [1, 12]!
```

Завдання 8. Дано три числа. Знайти кількість позитивних чисел серед них;

### Лістинг програми:

```
# task 8
def task_8_positive_numbers(numbers):
    positive_counter = 0
    for number in numbers:
        if number >= 1:
            positive_counter += 1
        return positive_counter

print('\nTASK 8!!!')
some_numbers = [10, -10, -10.5, -143411, 12032, +120, 1, 0, -1]
print(f'Range of numbers: {some_numbers}')
print(f'Number of positive numbers in range: {task 8 positive numbers(some numbers)}')
```

Результат програми:

Завдання 9. Дано два числа A і B (A <B). Знайти суму всіх цілих чисел від A до В включно. Вирішити задачу використовуючи циклічну конструкцію for.

		Бабушко А.С.				Арк.
		Морозов Д.С.			«Житомирська політехніка».21. <mark>121.01</mark> .000 – Лр2	6
Змн.	Арк.	№ докум.	Підпис	Дата		U

```
B = 15
```

```
TASK 9!!!
Sum of range from A(5) to B(15): 110
```

Завдання 10. Дано два числа A і B (A <B). Знайти суму квадратів всіх цілих чисел від А до В включно. Вирішити задачу використовуючи циклічну конструкцію for.

### Лістинг програми:

# Результат програми:



Завдання 11. Знайти середнє арифметичне всіх цілих чисел від а до 200 (значення a і b вводяться з клавіатури; a≤ 200). Вирішити задачу використовуючи циклічну конструкцію for.

# Лістинг програми:

		Бабушко А.С.		
		Морозов Д.С.		
Змн.	Арк.	№ докум.	Підпис	Дата

Арк.

```
def enter_A_and_B():
    while True:
        a = int(input('Enter A for task 11: '))
        b = int(input('Enter B for task 11: '))
        if a < b:
            break

    return [a, b]

print('\nTASK 11!!!')
some_range = enter_A_and_B()
print(f'Arithmetic mean from A({some_range[0]}) to B({some_range[1]}):'
    f'{task_11_arithmetic_mean(some_range[0], some_range[1])}')</pre>
```

Завдання 12. Знайти суму всіх цілих чисел від а до b (значення а і b вводяться з клавіатури; b ≥a). Вирішити задачу використовуючи циклічну конструкцію while.

### Лістинг програми:

```
# task 12
def task_12_sum_some_range(a, b):
    sum_some_range = 0
    while a <= b:
        sum_some_range += a
        a += 1

    return sum_some_range

print('\nTASK 12!!!')
print(f'Sum of range from A({A}) to B({B}): {task_12_sum_some_range(A, B)}')</pre>
```

### Результат програми:

Завдання 13. Знайти суму квадратів всіх цілих чисел від а до 50 (значення а вводиться з клавіатури; 0 ≤а≤50). Вирішити задачу використовуючи циклічну конструкцію for.

```
# task 13
def task_13_sum_some_square_range(a):
    sum_some_squares = 0
    for sum_num in range(a, 51, 1):
        sum_some_squares += math.pow(sum_num, 2)
```

		Бабушко А.С.		
		Морозов Д.С.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
return sum_some_squares

print('\nTASK 13!!!')
A = 48
print(f'Sum of squares of range from A({A}) to 50: {task_13_sum_some_square_range(A)}')
```

Завдання 14. Дано ціле число N (> 1). Знайти найменше ціле число K, при якому виконується нерівність 5 ^ K> N. Вирішити задачу використовуючи циклічну конструкцію while.

### Лістинг програми:

```
# task 14
def task_14_find_K(N_number):
    K = N_number
    K_array = [K]
    while math.pow(5, K) > N_number:
        K -= 1
        K_array.append(K)

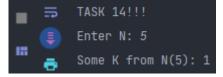
    return min(K_array)

def enter_N():
    while True:
        some_N = int(input('Enter N: '))
        if some_N > 1:
            break

    return some_N

print('\nTASK 14!!!')
N = enter_N()
print(f'Some K from N({N}): {task_14_find_K(N)}')
```

#### Результат програми:



Завдання 15. Серед чисел 1, 4, 9, 16, 25, ... знайти перше число, більше п. Вирішити задачу використовуючи циклічну конструкцію for.

```
# task 15
def task_15_find_number_greater_n(n):
    for number in range(1, n, 1):
        some_number = math.pow(number, 2)
        if some_number > n:
            return some_number
        else:
        continue
```

		Бабушко А.С.		
		Морозов Д.С.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
return n
print('\nTASK 15!!!')
N = enter_N()
print(f'The first number that greater than n: {task_15_find_number_greater_n(N)}')
```

```
TASK 15!!!
Enter N: 10
The first number that greater than n: 16.0
```

Завдання 16. Серед чисел 1 5 10, 17, 26, ... знайти перше число, більше n. Умовний оператор не використовувати. Вирішити задачу використовуючи циклічну конструкцію while.

### Лістинг програми:

```
# task 16
def task_16_find_number_greater_n(n):
    some_number = 1
    iterator = 2
    while some_number < n:
        some_number = math.pow(iterator, 2) + 1
        iterator += 1

    return some_number

print('\nTASK 16!!!')
N = enter_N()
print(f'The first number that greater than n: {task_16_find_number_greater_n(N)}')</pre>
```

### Результат програми:

```
TASK 16!!!
Enter N: 10
The first number that greater than n: 10.0
```

# Увесь лістинг програми:

		Бабушко А.С.		
		Морозов Д.С.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
def task 2 year days(year):
def is_leap_year(year):
task_2_year = [2000, 2001, 1932, 1841, 2004, 2014]
for i in range(len(task_2_year)):
    task_2_year_days(task_2_year[i])
task_3_purchases = [490, 510, 800, 1100, 2100]
for i in range(len(task_3_purchases)):
def task 4 min cosine(angles): return math.cos(min(angles))
angles_list = [math.radians(30 / math.pi), math.radians(60 / math.pi), math.radians(90 /
math.pi),
base_side = 2
```

		Бабушко А.С.		
		Морозов Д.С.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
some_numbers = [10, -10, -10.5, -143411, 12032, +120, 1, 0, -1]
print(f'Range of numbers: {some_numbers}')
A = 5
B = 15
B) } ')
def task 11_arithmetic_mean(a, b):
```

		Бабушко А.С.		
		Морозов Д.С.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
arithmetic mean sum += some num
some range = enter A and B()
```

		Бабушко А.С.		
		Морозов Д.С.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
print('\nTASK 14!!!')
N = enter N()
print(f'Some K from N({N}): (task_14_find_K(N))')

# task 15
def task_15_find_number_greater_n(n):
    for number in range(1, n, 1):
        some_number = math.pow(number, 2)
        if some_number > n:
            return some_number
        else:
            continue

return n

print('\nTASK 15!!!')
N = enter N()
print(f'The first number that greater than n: (task_15_find_number_greater_n(N))')

# task 15
def task_16_find_number_greater_n(n):
        some_number = 1
        iterator = 2
        while some_number < n:
            some_number = math.pow(iterator, 2) + 1
        iterator += 1

        return some_number

print('\nTASK 16!!!')
N = enter_N()
print(f'The first number that greater than n: (task_16_find_number_greater_n(N))')</pre>
```

**Висновок:** під час виконання лабораторної роботи було отримано навички з написання простих скриптів з використанням циклів for та while, перевірок умови іf на мові Python.

		Бабушко А.С.		
		Морозов Д.С.		
Змн.	Арк.	№ докум.	Підпис	Дата