

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».21.121.01.000–Лр2						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Бабушко А.С.			Звіт з лабораторної роботи			Літ.	Арк.	Аркушів	
Перевір.		Морозов Д. С.							1	00	
Керівник								ФІКТ Гр. ВТ-21-1[1]			
Н. контр.											
Затверд.											

**Мета 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)}')
```

#### **Результат програми:**

		Бабушко А.С.			«Житомирська політехніка».21.121.01.000 – Лр2	Арк.
		Морозов Д.С.				2
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:

```
Run: main x
TASK 2!!!
This year(2000) has 366 days.
This year(2001) has 365 days.
This year(1932) has 366 days.
This year(1841) has 365 days.
This year(2004) has 366 days.
This year(2014) has 365 days.
```

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.			«Житомирська політехніка».21.121.01.000 – Лр2	Арк.
		Морозов Д.С.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```
Run: main x
TASK 3!!!
Purchase: 490
Final price: 490
Purchase: 510
Final price: 494.7
Purchase: 800
Final price: 776.0
Purchase: 1100
Final price: 1045.0
Purchase: 2100
Final price: 1995.0
```

Завдання 4. Знайти косинус мінімального з 4 заданих чисел.

*Лістинг програми:*

```
# task 4
def task_4_min_cosine(angles):
    return math.cos(min(angles))

print('\nTASK 4!!!')
angles_list = [math.radians(30 / math.pi), math.radians(60 / math.pi), math.radians(90 / math.pi),
               math.radians(120 / math.pi), math.radians(360 / math.pi)]
print(f'Angles list: {angles_list}')
print(f'Min cos in radians from angles list: {task_4_min_cosine(angles_list)}')
```

*Результат програми:*

```
Run: main x
TASK 4!!!
Angles list: [0.16666666666666669, 0.3333333333333337, 0.5, 0.6666666666666667, 2.0]
Min cos in radians from angles list: 0.986143231562925
```

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

*Лістинг програми:*

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

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

		Бабушко А.С.			«Житомирська політехніка».21.121.01.000 – Лр2	Арк.
		Морозов Д.С.				4
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)}')
```

### Результат програми:

```
Run: main x
TASK 5!!!
Angles list: [0.16666666666666669, 0.3333333333333337, 0.5, 0.6666666666666667, 2.0]
Max sin in radians from angles list: 0.9092974268256817
```

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 x
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:

```
# task 7
def task_7_english_month(month_number):
    months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
              'September', 'October', 'November', 'December']
    if 1 <= month_number <= 12:
        print(f'{month_number} month is {months[month_number - 1]}')
    else:
        print(f'{month_number} number of month isn't in range [1, 12]!')

print('\nTASK 7!!!')
```

		Бабушко А.С.			«Житомирська політехніка».21.121.01.000 – Лр2	Арк.
		Морозов Д.С.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```
for i in range(0, 14):
    task_7_english_month(i)
```

### Результат програми:

```
Run: main x
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:

```
Run: main x
TASK 8!!!
Range of numbers: [10, -10, -10.5, -143411, 12032, 120, 1, 0, -1]
Number of positive numbers in range: 4
```

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

### Лістинг програми:

		Бабушко А.С.			«Житомирська політехніка».21.121.01.000 – Лр2	Арк.
		Морозов Д.С.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```
# task 9
def task_9_sum_some_range(a, b):
    sum_some_range = 0
    for some_num in range(a, b + 1, 1):
        sum_some_range += some_num

    return sum_some_range

print('\nTASK 9!!!')
A = 5
B = 15
print(f'Sum of range from A({A}) to B({B}): {task_9_sum_some_range(A, B)}')
```

### Результат програми:

```
Run: main x
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  $A$  до  $B$  включно. Вирішити задачу використовуючи циклічну конструкцію `for`.

### Лістинг програми:

```
# task 10
def task_10_sum_squares_some_range(a, b):
    sum_squares_some_range = 0
    for some_num in range(a, b + 1, 1):
        sum_squares_some_range += math.pow(some_num, 2)

    return sum_squares_some_range

print('\nTASK 10!!!')
print(f'Sum of squares of range from A({A}) to B({B}): {task_10_sum_squares_some_range(A, B)}')
```

### Результат програми:

```
Run: main x
TASK 10!!!
Sum of squares of range from A(5) to B(15): 1210.0
```

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

### Лістинг програми:

```
# task 11
def task_11_arithmetic_mean(a, b):
    arithmetic_mean_sum = 0
    count_nums = 0
    for some_num in range(a, b + 1):
        arithmetic_mean_sum += some_num
        count_nums += 1

    return arithmetic_mean_sum / count_nums
```

		Бабушко А.С.			«Житомирська політехніка».21.121.01.000 – Лр2	Арк.
		Морозов Д.С.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```
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])}')
```

### Результат програми:

```
Run: main x
TASK 11!!!
Enter A for task 11: 5
Enter B for task 11: 10
Arithmetic mean from A(5) to B(10):7.5
```

Завдання 12. Знайти суму всіх цілих чисел від  $a$  до  $b$  (значення  $a$  і  $b$  вводяться з клавіатури;  $b \geq a$ ). 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)}')
```

### Результат програми:

```
Run: main x
TASK 12!!!
Sum of range from A(5) to B(15): 110
```

Завдання 13. Знайти суму квадратів всіх цілих чисел від  $a$  до 50 (значення  $a$  вводиться з клавіатури;  $0 \leq a \leq 50$ ). Вирішити задачу використовуючи циклічну конструкцію `for`.

### Лістинг програми:

```
# 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.			«Житомирська політехніка».21.121.01.000 – Лр2	Арк.
		Морозов Д.С.				8
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)}')

```

### Результат програми:

```

Run: main x
TASK 13!!!
Sum of squares of range from A(48) to 50: 7205.0

```

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:

```

TASK 14!!!
Enter N: 5
Some K from N(5): 1

```

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

### Лістинг програми:

```

# 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.			«Житомирська політехніка».21.121.01.000 – Лр2	Арк.
		Морозов Д.С.				9
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)}')

```

### Результат програми:

```

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)}')

```

### Результат програми:

```

TASK 16!!!

Enter N: 10

The first number that greater than n: 10.0

```

### Увесь 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)}')

# task 2

```

		Бабушко А.С.			«Житомирська політехніка».21.121.01.000 – Лр2	Арк.
		Морозов Д.С.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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])

# 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')

# task 4
def task_4_min_cosine(angles): return math.cos(min(angles))

print('\nTASK 4!!!')
angles_list = [math.radians(30 / math.pi), math.radians(60 / math.pi), math.radians(90 / math.pi),
               math.radians(120 / math.pi), math.radians(360 / math.pi)]
print(f'Angles list: {angles_list}')
print(f'Min cos in radians from angles list: {task_4_min_cosine(angles_list)}')

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

print('\nTASK 5!!!')
print(f'Angles list: {angles_list}')
print(f'Max sin in radians from angles list: {task_5_max_sin(angles_list)}')

# 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

```

		Бабушко А.С.			«Житомирська політехніка».21.121.01.000 – Лр2	Арк.
		Морозов Д.С.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

for x in range(1, 13, 3):
    task_6_isosceles_triangle([left_right_sides + x, base_side + x])

# task 7
def task_7_english_month(month_number):
    months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
              'September', 'October',
              'November', 'December']
    if 1 <= month_number <= 12:
        print(f'{month_number} month is {months[month_number - 1]}')
    else:
        print(f'{month_number} number of month isn\'t in range [1, 12]!!')

print('\nTASK 7!!!')
for i in range(0, 14):
    task_7_english_month(i)

# 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)}')

# task 9
def task_9_sum_some_range(a, b):
    sum_some_range = 0
    for some_num in range(a, b + 1, 1):
        sum_some_range += some_num

    return sum_some_range

print('\nTASK 9!!!')
A = 5
B = 15
print(f'Sum of range from A({A}) to B({B}): {task_9_sum_some_range(A, B)}')

# task 10
def task_10_sum_squares_some_range(a, b):
    sum_squares_some_range = 0
    for some_num in range(a, b + 1, 1):
        sum_squares_some_range += math.pow(some_num, 2)

    return sum_squares_some_range

print('\nTASK 10!!!')
print(f'Sum of squares of range from A({A}) to B({B}): {task_10_sum_squares_some_range(A, B)}')

# task 11
def task_11_arithmetic_mean(a, b):
    arithmetic_mean_sum = 0
    count_nums = 0
    for some_num in range(a, b + 1):

```

		Бабушко А.С.			«Житомирська політехніка».21.121.01.000 – Лр2	Арк.
		Морозов Д.С.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        arithmetic_mean_sum += some_num
        count_nums += 1

    return arithmetic_mean_sum / count_nums

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])}')

# 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)}')

# 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)

    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)}')

# 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

```

		Бабушко А.С.			«Житомирська політехніка».21.121.01.000 – Лр2	Арк.
		Морозов Д.С.				13
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)}')

```

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

		Бабушко А.С.			«Житомирська політехніка».21.121.01.000 – Лр2	Арк.
		Морозов Д.С.				14
Змн.	Арк.	№ докум.	Підпис	Дата		