НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №2

із дисципліни «Основи програмування»

на тему

ПРОГРАМУВАННЯ ЦИКЛІЧНИХ АЛГОРИТМІВ

|  |  |
| --- | --- |
| Виконала: | Керівник: |
| студентка групи КМ-93 | ст.вик. Дрозденко О. М. |
| Довгаль Є. О. |  |

Київ — 2019

[1 ПОСТАНОВКА ЗАДАЧІ 3](#_Toc524472651)

[1.1 Мета роботи 3](#_Toc524472652)

[1.2 Завдання до лабораторної роботи 3](#_Toc524472653)

[2 ОПИС ПРОГРАМИ 4](#_Toc524472654)

[2.1 Змінні та функції 4](#_Toc524472655)

[Додаток А 6](#_Toc524472656)

[Текст програми 6](#_Toc524472657)

[Додаток Б 8](#_Toc524472658)

[Скріншоти тестувань програми 8](#_Toc524472659)

# 1 ПОСТАНОВКА ЗАДАЧІ

## Мета роботи

## Вивчення засобів реалізації циклічних конструкцій мови Python і отримання навичок складання програм з використанням операторів циклу.

## 1.2 Завдання до лабораторної роботи

1) Вивчити принципи побудови програм циклічної структури.

2) Розробити програми з операторами циклу відповідно до варіанта завдання.

3) Вхідні дані і результат роботи супроводжувати відповідною інформацією на екрані.

4) Показати розроблену програму викладачеві.

5) Письмово відповісти на Питання для самоперевірки.

6) Оформити звіт відповідно до вимог.

# 2 ОПИС ПРОГРАМИ

## 2.1 Змінні та функції

0.1) Перевірка: чи є значення типу int (цілим числом)

Функція get\_int використовується як допоміжний засіб при перевірки деяких введених користувачем чисел. Працює через цикл while True та конструкцію try-except (ValueError).

Використовується по аналогії з input: get\_int(faq: str) – викликається input(f"{faq}: ") та зберігає дані за умовою, якщо input – ціле число.

0.2) Установка слова «число» у правильній формі

Функція end отримує ціле число, потім трансформує його в string. Далі через if-else отримується остання цифра числа (останній елемент масива) і порівнюється з правилом. В решті ф-ція повертає одну з трьох форм.

1) Обчислити  (тренування for).

Використовується 4 змінні типу int – (index, result, x, n):

index – індекс сумування (=1), result – майбутній результат, який поверне це завдання (за замовчуванням = 0), x – змінна сумування (input), n – межа сумування (input).

Далі через цикл for виконується сумування (x+index)/index n разів. На виході отримується float число, яке округлюється до тисячних.

2) Тренування while

Використовується 3 змінні: 1 list, 1int, 1 float – (entered, num, average):

entered – масив типу list, який зберігає цілі числа з num; num – цілі числа, які вводить користувач; average – середнє арифметичне даних з entered.

Користувач за допомогою while True вводить цілі числа, які записуються у entered за допомогою ф-ції append(), але якщо введене число дорівнює нулю, тоді цикл переривається, АЛЕ якщо в entered немає жодного елементу, то користувач зобов’язан ввести як мінімум якесь інше число.

Далі average рахує середнє арифметичне введених даних, після чого програма виводить кількість введених чисел, самі числа, середнє арифметичне та їх суму.

Програма може виконуватися безкінечно, поки користувач відповідає на питання «Продовжити? y/n» символом «y». Все почнеться з самого початку.

# Додаток А

## Текст програми

Зовнішній вигляд збережено для більш зручного читання

def get\_int(faq: str):  
 while True:  
 try:  
 num = int(input(f'{faq}: '))  
 break  
 except ValueError:  
 print('ПОМИЛКА! Вводити потрібно лише цілі числа.')  
 return num  
  
  
# Вибирає закінчення для n числа  
def end(i: int):  
 ending = ['число', 'числа', 'чисел']  
 i = str(i)  
 if int(i) in range(11, 15): return ending[2]  
 elif int(i[-1]) == 1: return ending[0]  
 elif int(i[-1]) in range(2, 5): return ending[1]  
 else: return ending[2]  
  
  
def task1():  
 # Сумма (x+i)/i по n; i=1  
 print('\n<<< Виконання першого завдання >>>')  
 index, result = 1, 0  
 x, n = get\_int('Значення x'), get\_int('Межа сумування')  
  
 # Цикл выполнится ровно n раз  
 for i in range(n):  
 result += (x + index) / index  
 index += 1  
 return float(round(result, 3))  
  
  
def task2():  
 print('\n<<< Виконання другого завдання >>>')  
 """  
 Організувати безперервне введення чисел з клавіатури,  
 поки користувач не введе 0.  
 Після введення нуля, показати на екрані кількість чисел,  
 які були введені, їх загальну суму і середнє арифметичне.  
 """  
 entered = []  
 while True:  
 num = get\_int('Введіть число')  
 if num == 0:  
 if len(entered) == 0:  
 print('ПОМИЛКА! Треба ввести хоча б одне число.')  
 continue  
 break  
 entered.append(num)  
 average = round((float(sum(entered)) / len(entered)), 3)

print(f'Ви ввели {len(entered)} {end(len(entered))}:\n{entered}\n'  
 f'Их сумма: {sum(entered)}. Середнє арифметичне: {average}')  
  
  
while True:  
 print(f'Результат виконання першого завдання: {task1()}\n')  
 task2()  
 go = input('\nПродовжити? Y/N\n')  
 if go.lower() != 'y': break  
  
# Автор програми, номер лаб. роботи, варіант завдань  
print('\n>>> Лабораторну роботу №2 виконала\n'  
 '>>> студентка КМ-93 Довгаль Єва\n'  
 '>>> Варіанти кожного завдання: 4')

# Додаток Б

## Скріншоти тестувань програми

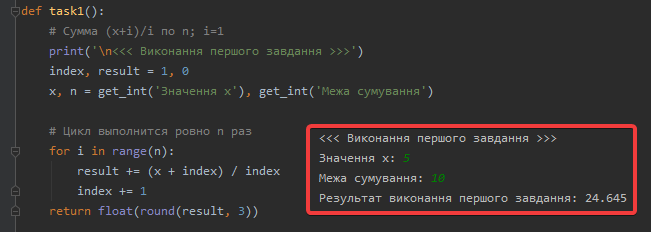


Рисунок 3.1 – Перше завдання

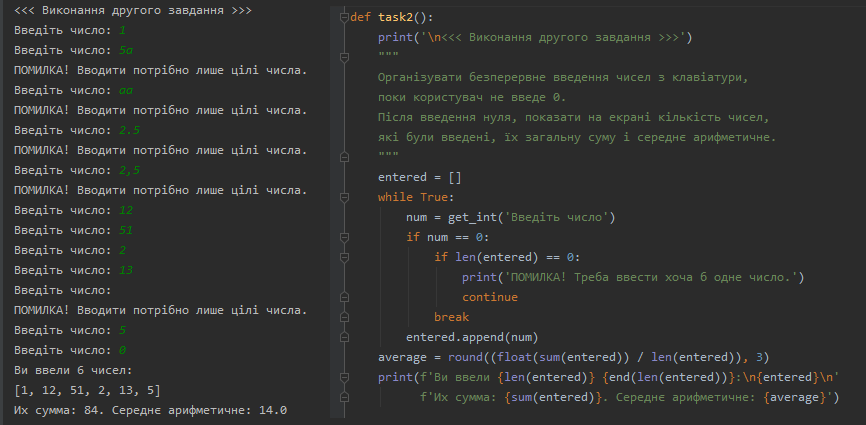


Рисунок 3.2 – Друге завдання

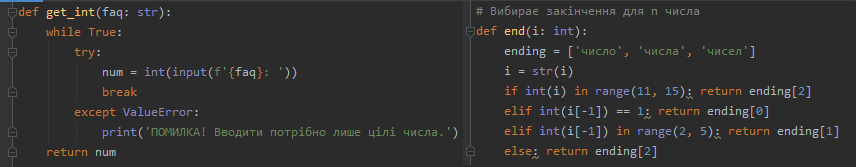


Рисунок 3.3 – ф-ції get\_int та end