МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Український державний університет імені Михайла Драгоманова

Факультет математики, інформатики та фізики

Кафедра інформаційних технологій та програмування

**Звіт**

з лабораторної роботи №2

«»

з дисципліни «Програмування»

Виконав:

студент ІІІ курсу групи 31І

Бритвич О.В.

Перевірила:

викладач Устименко О.Б.

Оцінка

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_

Київ – 2023

Зміст

Мета роботи 3

1 Постановка задачі 4

2 Основна частина 5

2.1 Опис вхідних та вихідних даних 6

2.1 Блок-схема 7

2.2 Опис вхідних та вихідних даних 11

2.2 Блок-схема 12

Висновки 14

Список літератури 15

Додатки 16

**Мета роботи**

**1 Постановка задачі**

1. Задача №1.

В університеті використовується наступна шкала для інтерпретації результатів тестування студентів: 90 балів і вище *(A)*, 80-89 *(B)*, 70-79 *(C)*, 60-69 *(D)*, нижче 60 *(F)*. Напишіть програму, яка дозволить студенту ввести тестовий бал, а потім відобразити оцінку для цього балу.

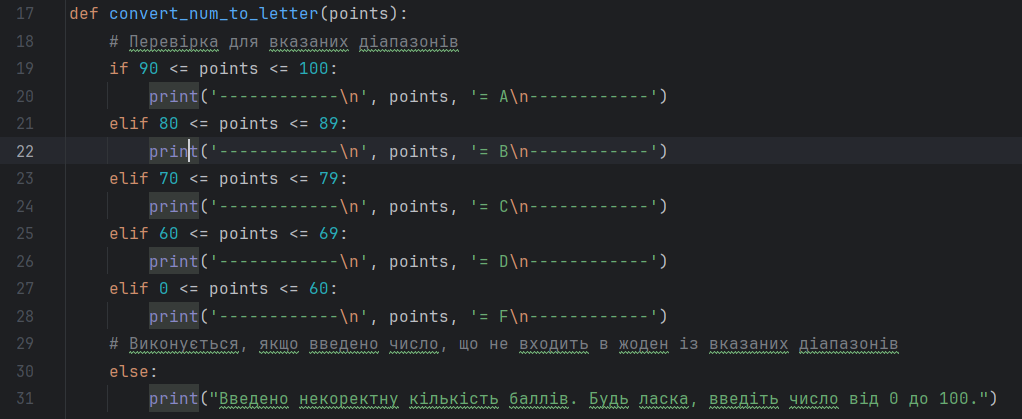
1. Задача №2.

Тетянка полюбляє їздити у громадському транспорті, а отримуючи квиток, відразу перевіряє, чи він щасливий. Квиток вважається щасливим, якщо сума перших трьох цифр збігається з сумою останніх трьох цифр номера квитка. Напишіть програму, яка перевірить рівність сум і виведе *Happy*, якщо суми збігаються, і *Ordinary*, якщо суми різні. На вхід програмі подається рядок із шести цифр

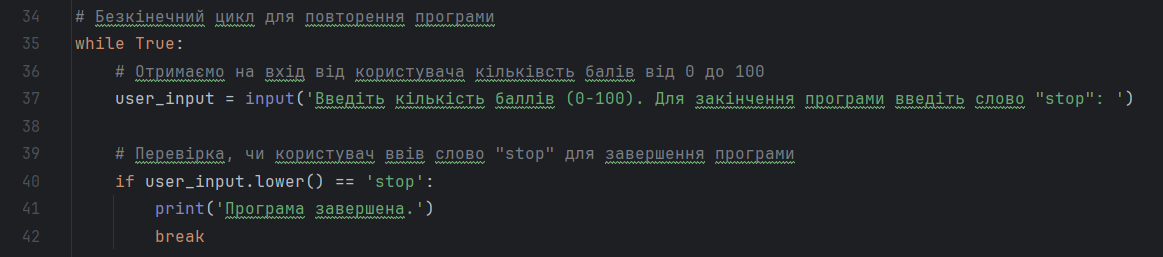
**2 Основна частина**

Задача №1

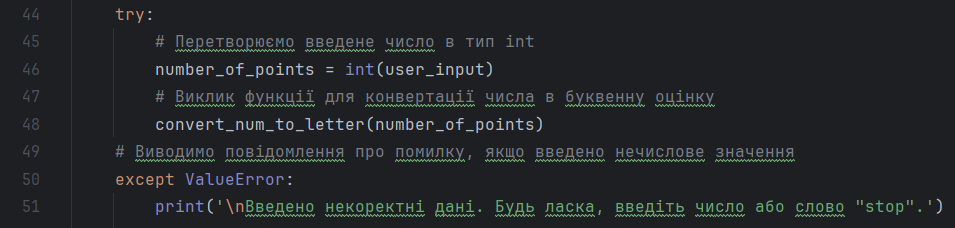
Наступна функція отримує на вхід кількість балів (points), та перевіряє, в якому діапазоні знаходиться введена кількість балів і виводить відповідну буквенну оцінку. Якщо число не входить в жоден з вказаних діапазонів, виводиться повідомлення про некоректну кількість балів:



Цикл працює безкінечно, доки користувач не введе слово «stop»:



Використовується конструкція *try-except* для перетворення введеного значення в ціле число. Якщо виводиться нечислове значення, виводиться повідомлення про помилку:



**2.1 Опис вхідних та вихідних даних**

Вхідні дані:

user\_input – змінна для збереження кількості балів.

Вихідні дані:

points – змінна для збереження буквеного еквіваленту та оброблення вхідних даних.

**2.1 Блок-схема**

Початок

def convert\_num\_to\_letter(points)

Вивести еквівалент балів в діапазоні від 70 до 79

Вивести еквівалент балів в діапазоні від 60 до 69

elif 60 <= points <= 69

elif 80 <= points <= 89

Вивести еквівалент балів в діапазоні від 80 до 89

elif 70 <= points <= 79

Вивести еквівалент балів в діапазоні від 90 до 100

if 90 <= points <= 100

Вивести текст

Вихід з циклу break

Вивести текст

if user\_input.lower() == 'stop'

Увести user\_input

while True

Вивести попереджуючий текст

else

Вивести еквівалент балів в діапазоні від 0 до 60

elif 0 <= points <= 60

Кінець

Вивести текст

except ValueError

convert\_num\_to\_letter(number\_of\_points)

number\_of\_points = int(user\_input)

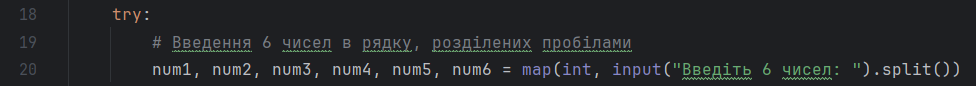
try

Задача №2

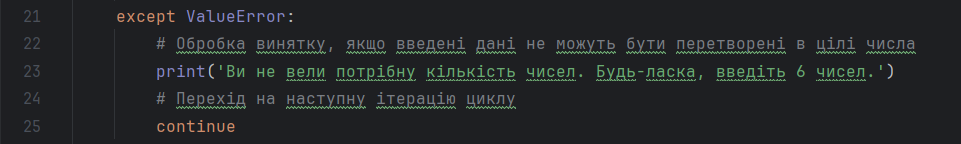
Безкінечний цикл (*while True*) буде виконуватися постійно, доки не буде вручну зупинений:



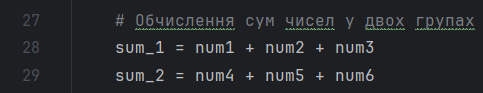
Введення змінних в блок *try* для обробки їх винятків:



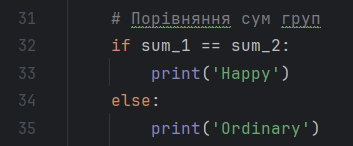
У блоці except обробляємо виняток *ValueError*, у разі якого виводиться повідомлення про помилку, і виконується *continue*, що переходить до наступної ітерації циклу:



Обчислюються суми *sum\_1* та *sum\_2* для двох груп по три числа в кожній:



Порівнюється сума першої групи (*sum\_1*) і сума другої групи (*sum\_2*). Якщо суми рівні, виводиться повідомлення «Happy», в іншому випадку – «Ordinary».



**2.2 Опис вхідних та вихідних даних**

Вхідні дані:

num1 – змінна для збереження першого введеного числа;

num2 – змінна для збереження другого введеного числа;

num3 – змінна для збереження третього введеного числа;

num4 – змінна для збереження четвертого введеного числа;

num5 – змінна для збереження п’ятого введеного числа;

num6 – змінна для збереження шостого введеного числа.

Вихідні дані:

Happy – текст, що виводиться при дійсній умові задачі (сума перших трьох чисел = сумі останніх трьох);

Ordinary – текст, що виводиться при порушенні умови задачі (сума перших трьох чисел ≠ сумі останніх трьох).

**2.2 Блок-схема**

Початок

while True

if sum\_1 == sum\_2

sum\_2 = num4 + num5 + num6

sum\_1 = num1 + num2 + num3

Введення num1, num2, num3, num4, num5, num6

except ValueError

Вивести попереджуючий текст

continue

try

Вивести *Happy*

Вивести *Ordinary*

else

**Висновки**

Під час роботи над лабораторною роботою були розроблені дві програми, призначені для вирішення конкретних завдань.

Перша програма дозволяє студентам вводити тестові бали і отримувати відповідні оцінки згідно з університетською шкалою. Реалізація програми дозволяє швидко та точно визначити оцінку, що полегшує процес оцінювання та допомагає уникнути помилок.

Друга програма призначена для перевірки квитка на шанс унікальності на основі сум перших та останніх трьох цифр номера. Простий та зрозумілий інтерфейс дозволяє використовувати програму з легкістю, а результат виводиться в зрозумілій формі – «Happy» або «Ordinary».

Обидві програми виконують свої завдання швидко та ефективно. У майбутньому можна розглядати розширення шкали оцінювання для першої програми, а також додавання додаткових перевірок для другої. Для підвищення зручності користувача можна розглядати додаткову розробку графічного інтерфейсу програми.

**Список літератури**

1Вивчаємо Python, В двох томах. 5-е видання. Марк Лутц (комплект з двох книг) – [Режим доступу] : <https://leader-books.com.ua/komp-yuterna-literatura/vivchayemo-python-v-dvoh-tomah-5-e-vidannya-mark-lutc-komplekt-z-dvoh-knig?gclid=CjwKCAiAgeeqBhBAEiwAoDDhnxzTW_ZFWceIMmEQLdDM0XPtB2uG9tUdKMSjvmw6BX7zHEoS49h0gxoC_skQAvD_BwE>

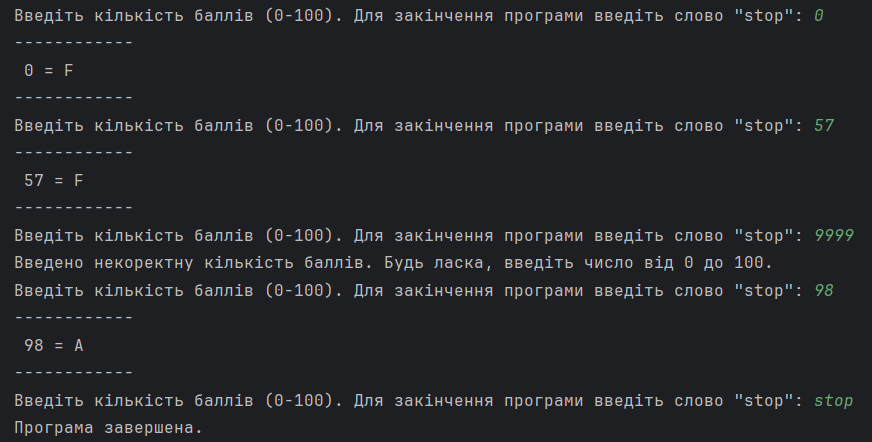
2 Форум з технічних питань – [Режим доступу] : <https://stackoverflow.com/>

**Додатки**

Текст програми №1

# Ініціалізація функції для конвертації числа у відповідну буквенну оцінку  
def convert\_num\_to\_letter(points):  
 # Перевірка для вказаних діапазонів  
 if 90 <= points <= 100:  
 print('------------\n', points, '= A\n------------')  
 elif 80 <= points <= 89:  
 print('------------\n', points, '= B\n------------')  
 elif 70 <= points <= 79:  
 print('------------\n', points, '= C\n------------')  
 elif 60 <= points <= 69:  
 print('------------\n', points, '= D\n------------')  
 elif 0 <= points <= 60:  
 print('------------\n', points, '= F\n------------')  
 # Виконується, якщо введено число, що не входить в жоден із вказаних діапазонів  
 else:  
 print("Введено некоректну кількість баллів. Будь ласка, введіть число від 0 до 100.")  
  
# Безкінечний цикл для повторення програми  
while True:  
 # Отримаємо на вхід від користувача кільківсть балів від 0 до 100  
 user\_input = input('Введіть кількість баллів (0-100). Для закінчення програми введіть слово "stop": ')  
  
 # Перевірка, чи користувач ввів слово "stop" для завершення програми  
 if user\_input.lower() == 'stop':  
 print('Програма завершена.')  
 break  
  
 try:  
 # Перетворюємо введене число в тип int  
 number\_of\_points = int(user\_input)  
 # Виклик функції для конвертації числа в буквенну оцінку  
 convert\_num\_to\_letter(number\_of\_points)  
 # Виводимо повідомлення про помилку, якщо введено нечислове значення  
 except ValueError:  
 print('\nВведено некоректні дані. Будь ласка, введіть число або слово "stop".')

Результат роботи програми №1



Текст програми №2

# Ініціалізація безкінечного циклу  
while True:  
  
 try:  
 # Введення 6 чисел в рядку, розділених пробілами  
 num1, num2, num3, num4, num5, num6 = map(int, input("Введіть 6 чисел: ").split())  
 except ValueError:  
 # Обробка винятку, якщо введені дані не можуть бути перетворені в цілі числа  
 print('Ви не ввели потрібну кількість чисел. Будь-ласка, введіть 6 чисел.')  
 # Перехід на наступну ітерацію циклу  
 continue  
  
 # Обчислення сум чисел у двох групах  
 sum\_1 = num1 + num2 + num3  
 sum\_2 = num4 + num5 + num6  
  
 # Порівняння сум груп  
 if sum\_1 == sum\_2:  
 print('Happy')  
 else:  
 print('Ordinary')

Результат роботи програми №2

