#1

is\_next = None

num = int(input("Enter the number of points: "))

if num >= 83:

is\_next = True

else:

is\_next = False

Реалізуйте оператор контролю виконання так, щоб він привласнив логічній змінній is\_next значення True, якщо кількість набраних балів буде більшою або дорівнює 83. В іншому випадку значення змінної дорівнює False.

#2

Приведіть змінні is\_active, is\_admin та is\_permission до булевого вигляду.

Надайте змінній access, значення, яке покаже, чи є доступ у користувача. Використовуйте логічні оператори.

Адміністратор завжди має доступ, незалежно від значень змінних is\_permission та is\_active.

Користувач має доступ, тільки якщо is\_permission дорівнює True та is\_active також дорівнює True

is\_active = bool(input("Is the user active? "))

is\_admin = bool(input("Is the user administrator? "))

is\_permission = bool(input("Does the user have access? "))

if is\_admin:

access = True

elif is\_active == True and is\_permission == True:

access = True

else:

access = False

#3

Є змінна work\_experience, що визначає стаж роботи програміста. Залежно від стажу роботи, присвоїти змінній developer\_type значення "Junior", "Middle" або "Senior". Використовуйте, якщо необхідно, булеві оператори **or** та **and** під час перевірок.

work\_experience = int(input("Enter your full work experience in years: "))

if 1 < work\_experience <= 5:

developer\_type = "Middle"

elif work\_experience <= 1:

developer\_type = "Junior"

else:

developer\_type = "Senior"

#4

змінна result повинна містити одне з чотирьох значень:

* "Positive even number"
* "Positive odd number"
* "Negative number"
* "It is zero"

Підказка: перевірка на парність виконується порівнянням залишку від поділу на 2 з 0 або 1. Нагадаємо, залишок від ділення можна отримати після операції %

num = int(input("Enter a number: "))

if num > 0:

if num%2 == 1:

result = "Positive odd number"

else:

result = "Positive even number"

elif num < 0:

result = "Negative number"

else:

result = "It is zero"

#5

Необхідно обчислити коріння квадратного рівняння.

a · x2 + b · x + c = 0

Дискримінант рівняння помістіть у змінну D

D = b2 - 4 · a · c

Коріння рівняння помістіть у змінні x1 та x2

x1 = (-b + D0.5) / (2 · a)

x2 = (-b - D0.5) / (2 · a)

Минулого разу ми просто вказали значення коефіцієнтів a, b, c. Тепер, коли ми вже знаємо про розгалуження, ми можемо перевіряти значення дискримінанта і, в залежності від того додатне чи від'ємне, провести розрахунок коренів. Виконайте розрахунок коренів для довільних значень коефіцієнтів a, b, c.

import math

a = int(input("Enter coefficient a: "))

b = int(input("Enter coefficient b: "))

c = int(input("Enter coefficient c: "))

D = b \*\* 2 - 4 \* a \* c

if D > 0:

x1 = (-b + math.sqrt(D)) / (2 \* a)

x2 = (-b - math.sqrt(D)) / (2 \* a)

else:

x = (-b ) / (2 \* a)

#6

Виконайте завдання, щоб визначити парне число чи ні, за допомогою тернарного оператора.

Програма встановлює значення змінної is\_even у True або False, залежно від того, чи є число в змінній num парним або непарним.

num = int(input("Enter an integer number: "))

is\_even = True if num%2 == 0 else False

#7

Користувач вводить число від 0 до 100. Підрахуйте, використовуючи цикл while, суму всіх чисел від 1 до введеного числа включно num. Результат помістити в змінну sum.

Тести будуть:

Поміщати тестові значення для змінної num: 20, 10, 5, 100

І чекати суми в змінній sum: 210, 55, 15, 5050

num = int(input("Enter the integer (0 to 100): "))

sum = 0

i = 0

while i < num + 1:

sum = i + sum

i += 1

print(sum)

#8

Рядок — це об'єкт, що ітерується, а, значить, ми можемо використовувати його в циклі for. Підрахуйте в заданому рядку message кількість входжень символу зі змінної search. Результат помістіть у змінну result.

message = "Never argue with stupid people, they will drag you down to their level and then beat you with experience."

search = "r"

result = 0

for letter in message:

if letter == search:

result += 1

#9

розрахуйте найбільший спільний дільник (НД) двох додатних чисел. НСД — це найбільше число, яке без залишку діляться на обидва числа.

Є кілька алгоритмів знаходження НСД, але ми розберемо циклічний алгоритм

1. Хай є два початкових числа first та second.
2. Виберемо менше з них та привласнимо значення змінній gcd.
3. Поки first або second не діляться на gcd без залишку, треба виконувати цикл, в якому зменшуємо змінну gcd на одиницю.
4. Коли цикл закінчиться в змінній gcd буде НСД для чисел first та second

Напишіть програму, яка для двох додатних цілих чисел знаходить НСД.

*Примітка*: Для умови циклу в пункті 3 необхідно пам'ятати, що цикл while виконується за умови True, а наш цикл повинен закінчитися, тільки якщо gcd поділив обидва числа без залишку.

first = int(input("Enter the first integer: "))

second = int(input("Enter the second integer: "))

gcd = min(first, second)

while first % gcd != 0 or second % gcd != 0:

gcd = gcd - 1

print(gcd)

#10

Напишіть два цикли, вкладені один в один. У першому циклі while ми постійно запитуємо ціле число, а у другому з допомогою циклу for складаємо суму чисел від 0 до введеного числа включно і додаємо до змінної sum. Змінна sum накопичує суми, що утворюються при кожному введенні числа. Вихід з першого циклу здійснюємо, якщо ми ввели число 0.

Тести використовують дві тестові послідовності чисел:

* 10, 13, 73, 0 і чекають на суму 2847
* 1, 2, 3, 4, 0 і чекають на суму 20

num = int(input("Enter integer (0 for output): "))

sum = 0

while num != 0:

for i in range(num + 1):

sum = sum + i

num = int(input("Enter integer (0 for output): "))

print(sum)

#11

Перепишіть попередній приклад, але використовуючи оператор break.

Напишіть два подвійні цикли. У першому циклі while ми постійно запитуємо ціле число, а у другому за допомогою циклу for обчислюємо суму чисел від 0 до введеного числа. Вихід з першого циклу здійснюємо, якщо ввели число 0, за допомогою оператора break.

sum = 0

while True:

num = int(input("Enter integer (0 for output): "))

if num == 0:

break

for i in range(num + 1):

sum = sum + i

#12

Напишіть два подвійні цикли. У першому циклі while ми постійно запитуємо ціле число, а у другому за допомогою циклу for обчислюємо суму **парних** чисел від 0 до введеного числа. Вихід з першого циклу здійснюємо, якщо ввели число 0 за допомогою оператора break.

Тести використовують дві тестові послідовності чисел:

* 10, 13, 73, 0 і чекають на суму 1404
* 1, 2, 3, 4, 0 і чекають на суму 10

sum = 0

while True:

num = int(input("Enter integer (0 for output): "))

if num == 0:

break

for i in range(num + 1):

if i % 2 == 1:

continue

sum = sum + i

#13

**КОД ЦЕЗАРЯ**

Ідея шифрування полягає у циклічному зміщенні букв на задану кількість. Наприклад, якщо зміщення на три позиції, то літера A стає літерою D, B – E тощо. Останні три літери алфавіту зациклюються та переносяться на початок. Літера X стає A, Y – B, а Z – C. Цифри, пробіли та інші символи не шифруються.

У програмі користувач вводить фразу та число для зсуву, після чого треба вирахувати нове закодоване повідомлення.

Програма шифруватиме як малі (a-z), так і великі літери (A-Z).

Тести перевіряють та кодують наступні рядки:

* "Hello my little friends!", offset = 37,
* "Hello world!", offset = 7

message = input("Введите сообщение: ")

offset = int(input("Введите сдвиг: "))

encoded\_message = ""

for ch in message:

    if 32 <= ord(ch) < 63:

        #    pos = ord(ch) - ord(' ')

        #    pos = (pos + offset) % (32)

        #    new\_char = chr(pos + ord(' '))

        #    encoded\_message = encoded\_message + new\_char

        new\_char = ch

        encoded\_message = encoded\_message + new\_char

    elif 64 <= ord(ch) < 95:

        pos = ord(ch) - ord('A')

        pos = (pos + offset) % 26

        new\_char = chr(pos + ord('A'))

        encoded\_message = encoded\_message + new\_char

    else:

        pos = ord(ch) - ord('a')

        pos = (pos + offset) % 26

        new\_char = chr(pos + ord('a'))

        encoded\_message = encoded\_message + new\_char

print(encoded\_message)

#14

Ситуація проста, вам необхідно вирахувати кількість SMS, які треба надсилати в одному пакеті розсилки потенційним покупцям. Всього на день виділяється 1000 платних SMS для кампанії маркетингу pool=1000. Співробітник відділу маркетингу вводить кількість розсилок quantity, і ви обчислюєте розмір пакета chunk = pool // quantity. Опрацюйте помилку поділу на нуль.

pool = 1000

quantity = int(input("Enter the number of mailings: "))

try:

chunk = pool // quantity

except:

print('Divide by zero completed!')

#15