**Практична робота №2**

**Тема:** Робота на парі

**Мета:** Особливості організації умовних операторів і множинних розгалужень. Розробка програм. Особливості організації циклічної інструкції while. Особливості організації умовних операторів і множинних розгалужень. Розробка програм.

**Хід роботи**

**Завдання 1.**

Присвоїти двом змінним будь-які числові значення.

**Код:**

a = 10

b = 5

print(a,b)

**Результат:**



**Пояснення:**

* Ми визначаємо дві змінні a і b, які відповідають числам 10 і 5 відповідно. Потім виводимо значення цих змінних на екран, щоб показати їх користувачеві.

**Завдання 2.**

Складіть чотири складних логічних вирази за допомогою

оператора and, два з яких повинні давати істину, а два інших

- хибність.

**Код:**

a=10

b=10

result1 = (a > 0 and b > 0) #True

result2 = (a == 10 and b == 10) #True

result3 = (a < 0 and b > 0) #False

result4 = (a > 0 and b < 0) #False

print(result1,result2,result3,result4)

**Результат:**



**Пояснення:**

* result1 перевіряє, чи є обидва числа a і b більшими за 0, що є правдою.
* result2 перевіряє, чи обидва числа a і b рівні 10, що також є правдою.
* result3 перевіряє, чи a менше 0 і b більше 0, що є неправдою, оскільки обидва числа дорівнюють 10.
* result4 перевіряє, чи a більше 0 і b менше 0, що також є неправдою, оскільки обидва числа дорівнюють 10.

**Завдання 3.**

Аналогічно виконайте п. 2, використовуючи оператор or.

**Код:**

a=11

b=11

result5 = (a > 0 or b > 0)    # True

result6 = (a < 0 or b > 0)    # True

result7 = (a < 0 or b < 0)    # False

result8 = (a == 10 or b == 10) # False

print(result5,result6,result7,result8)

**Результат:**



**Пояснення:**

* result5 перевіряє, чи є хоча б одне з чисел a або b більшим за 0, що є правдою.
* result6 перевіряє, чи хоча б одне з чисел a або b менше 0, що також є правдою, оскільки ні a, ні b не менше 0.
* result7 перевіряє, чи обидва числа a і b менше 0, що є неправдою, оскільки ні a, ні b не менше 0.
* result8 перевіряє, чи хоча б одне з чисел a або b дорівнює 10, що є неправдою, оскільки ні a, ні b не дорівнюють 10.

**Завдання 4.**

Спробуйте використати в складних логічних виразах

роботу зі змінними рядкового типу.

**Код:**

str1 = "hello"

str2 = "world"

result9 = (str1 == "hello" and str2 == "world")  # True

result10 = (str1 == "hello" or str2 == "python") # True

result11 = (str1 == "hi" and str2 == "world")   # False

result12 = (str1 == "hi" or str2 == "python")   # False

print(result9,result10,result11,result12)

**Результат:**



**Пояснення:**

* **result9** перевіряє, чи обидва рядки рівні "hello" та "world", що правда.
* **result10** перевіряє, чи хоча б один із рядків дорівнює "hello" або "python", що також правда, оскільки **str1** рівне "hello".
* **result11** перевіряє, чи рядок **str1** дорівнює "hi" і рядок **str2** дорівнює "world", що є неправдою, оскільки **str1** не дорівнює "hi".
* **result12** перевіряє, чи хоча б один із рядків дорівнює "hi" або "python", що також є неправдою, оскільки ні **str1**, ні **str2** не дорівнюють "hi" або "python".

**Завдання 5.**

Напишіть програмний код, в якому в разі, якщо значення

якоїсь змінної більше 0, виводилося б спеціальне

повідомлення (використовуйте функцію print). Один раз

виконайте програму при значенні змінної більше 0, вдруге -

менше 0.

**Код:**

x = 5

if x > 0:

    print("Значення більше 0")

else:

    print("Значення менше 0")

**Результат:**



**Пояснення:**

* Якщо x більше за 0, то виконується перший блок коду, який виводить повідомлення "Значення більше 0".
* Якщо x менше або дорівнює 0, то виконується другий блок коду, який виводить повідомлення "Значення менше 0".
* У нашому випадку x дорівнює 5, що більше 0, тому виводиться перше повідомлення.

**Завдання 6.**

Вдоскональте попередній код за допомогою гілки else так,

щоб в залежності від значення змінної, виводилася або 1,

або -1.

**Код:**

x = 5

if x > 0:

    print(1)

else:

    print(-1)

**Результат:**



**Пояснення:**

* Тут ми маємо умовну конструкцію, яка перевіряє, чи є значення змінної x більшим за 0.
* Якщо умова x > 0 виконується (що правда для x = 5), то виконується перший блок коду, який виводить 1.
* Якщо умова не виконується, тобто x менше або дорівнює 0, то виконується другий блок коду, який виводить -1.
* Оскільки x дорівнює 5, що більше за 0, виведеться 1.

**Завдання 7.**

Напишіть програму за наступним описом:

a. двом змінним присвоюються числові значення;

b. якщо значення першої змінної більше другої, то

знайти різницю значень змінних (відняти від першої

другу), результат зв'язати з третьою змінною;

c. якщо перша змінна має менше значення, ніж друга, то

третю змінну зв'язати з результатом суми значень

двох перших змінних;

d. у всіх інших випадках, привласнити треті змінні

значення першої змінної;

e. вивести значення третьої змінної на екран.

**Код:**

a = 2

b = 1

if a > b:

    c = a - b

elif a < b:

    c = a + b

else:

    c = a

print(c)

**Результат:**



**Пояснення:**

* Тут ми маємо умовну конструкцію, яка порівнює значення змінних a і b.
* Якщо a більше за b, то виконується перший блок коду, де c дорівнює різниці між a і b.
* Якщо a менше за b, то виконується другий блок коду, де c дорівнює сумі a і b.
* Якщо ж жодна з цих умов не виконується, тобто a дорівнює b, то виконується блок else, де c приймає значення a.
* У нашому випадку a дорівнює 2 і b дорівнює 1, тому виконується перший блок коду, і c приймає значення 2 - 1, тобто 1. Тому на екран виводиться 1.

**Індивідуальні завдання**

**Завдання 1.**

Напишіть програму, яка в залежності від характеру вітру видає повідомлення про його швидкість: від 1 до 4 м / с - слабкий; від 5-10 м/c - помірний; від 9-18 м/c - сильний; більше 19 м / c – ураганний

**Код:**

def wind\_speed\_description(speed):

    if 1 <= speed <= 4:

        return "Слабкий вітер"

    elif 5 <= speed <= 10:

        return "Помірний вітер"

    elif 11 <= speed <= 18:

        return "Сильний вітер"

    elif speed >= 19:

        return "Ураганний вітер"

    else:

        return "Некоректна швидкість вітру"

try:

    speed = float(input("Введіть швидкість вітру: "))

    print(wind\_speed\_description(speed))

except ValueError:

    print("Будь ласка, введіть числове значення.")

**Результат:**



**Пояснення:**

* Функція wind\_speed\_description(speed) приймає швидкість вітру як вхідний параметр.
* Умовні конструкції перевіряють, в який діапазон потрапляє введена швидкість вітру.
* Якщо введена швидкість знаходиться в межах від 1 до 4 включно, функція повертає "Слабкий вітер".
* Якщо швидкість вітру знаходиться в межах від 5 до 10 включно, функція повертає "Помірний вітер".
* Якщо швидкість вітру знаходиться в межах від 11 до 18 включно, функція повертає "Сильний вітер".
* Якщо швидкість вітру 19 або більше, функція повертає "Ураганний вітер".
* Якщо введене значення не підходить під жодну з цих категорій, програма виводить повідомлення "Некоректна швидкість вітру".
* Програма використовує блок try-except, щоб перехопити помилки вводу, і виводить повідомлення про помилку, якщо введене значення не є числом.

**Завдання 2.**

Знайти суму цифр заданого натурального числа n.

**Код:**

def sum\_of\_numbers(n):

    sum\_num = 0

    while n > 0:

        # Додаємо останню цифру до суми

        sum\_num += n % 10

        # Видаляємо останню цифру з числа

        n = n // 10

    return sum\_num

number = int(input("Введіть натуральне число: "))

print("Сума цифр числа:", sum\_of\_numbers(number))

**Результат:**



**Пояснення:**

* У функції sum\_of\_numbers(n) створюється змінна sum\_num, що ініціалізується як 0.
* У циклі while перевіряється, чи число n є більшим за 0. Якщо так, програма продовжує виконуватися.
* У тілі циклу:
  1. Остання цифра числа додається до змінної sum\_num за допомогою оператора % 10.
  2. Потім остання цифра видаляється з числа, щоб перейти до наступної цифри, за допомогою операції n = n // 10.
* Коли всі цифри були додані до sum\_num, функція повертає значення sum\_num.
* Користувач вводить натуральне число.
* Потім викликається функція sum\_of\_numbers(number), де number - введене користувачем число.
* На екран виводиться сума цифр цього числа.