Міністерство освіти і науки України

Державний університет ,,Житомирська політехніка”

Група: ІПЗ-21-1

Програмування мовою Python

Лабораторна робота № 2

«РОЗГАЛУЖЕННЯ ТА ЦИКЛИ»

Виконав: Беккер М.М

Прийняв: Морозов Д. С.

**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.

Виконання роботи:

Завдання 1. Дано три цілих числа. Вибрати з них ті, які належать інтервалу [1,3].

def get\_interval\_number(a, b, c):  
 arr = [a, b, c] # запис даних в лист  
 return list(filter(lambda x: x >= 1 and x <= 3, arr)) # фільтрація масиву для заданого діапазону  
  
print(get\_interval\_number(4, 2, 3)) # результат виконання функції

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).

def check\_year(year):  
 if year % 400 == 0 or (year % 4 == 0 and year % 100 != 0): #перевірка на високосний рік та повернення 366  
 return 366  
 else:  
 return 365 #якщо рік не вискосний повертаемо 355  
  
print(check\_year(2022))

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

def get\_discount(suma):  
 if 500 < suma < 1000: #якщо сума в діапазоні від 500 до 1000  
 suma = suma - (suma \* 3 / 100) #рахуемо знижку 3 відсотка  
 print(f"Discount 3%! Your final sum: {suma}")  
 elif suma > 1000: #якщо сума більше 1000  
 suma = suma - (suma \* 5 / 100) # рахуємо знижку 5 відсотків  
 print(f"Discount 5%! Your final sum: {suma}")  
  
  
get\_discount(1001)

Завдання 4. Знайти косинус мінімального з 4 заданих чисел.

def calculate\_min\_cos(a, b, c, d):  
 arr = [a, b, c, d] # заносимо дані в лист  
 return math.cos(min(arr)) # за допомогою функції мін шукаємо мінімальний елемент та рахуемо його косинус  
  
  
print(calculate\_min\_cos(12, 4, 8, 5))

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

def get\_max\_sin(a, b, c):  
 arr = [a, b, c] # заносимо дані в лист  
 return math.sin(max(arr)) # за допомогою функції мах шукаємо максимальний елемент та рахуемо його сінус  
  
  
print(get\_max\_sin(4, 12, 3))

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!»

def calculate\_s(height, sides):  
 s = 1 / 2 \* float(sides) \* float(height) # рахуємо площу  
 if s % 2 == 0: # якщо площа ділиться на 2, повертаємо результат  
 return s / 2  
 else:  
 return "I can't divide this" #інакше виводимо повідомлення, що я не можу ділити це  
  
  
print(calculate\_s(4, 2))

Завдання 7. Скласти програму, яка по даному числу (1-12) виводить назву відповідного йому місяця англійською мовою.

def get\_year\_name\_by\_number(number):  
 # словник місяців  
 month = {1: "January", 2: "February", 3: "March", 4: "April", 5: "May", 6: "June", 7: "July", 8: "August",  
 9: "September", 10: "October", 11: "November", 12: "December"}  
 if not number in month: # якщо в словнику немає такого ключа, пишемо, зо місяця нема  
 return "month not exist"  
  
 return F"You month is {month[int(number)]}"

Завдання 8. Дано три числа. Знайти кількість позитивних чисел серед них;

def get\_positive\_number(a, b, c):  
 arr = [a, b, c] # заносимо дані в масив  
 return len(list(filter(lambda x: x > 0, arr))) #фільтрація масиву якщо елемет більше 0,тобто додатній  
 # поветаємо довжину, тому що в списку залишаться тільки ті елементи які прошли умову  
  
  
print(get\_positive\_number(2, -10, 4))

Завдання 9.

def get\_sum\_integer(a, b):  
 sum = 0  
 if b < a: # якшо б менше, повертаємо помилку  
 return "invalid value b"  
 for i in range(a, b + 1): #якщо все ок, проходимо по елементам від а до б  
 print(i)  
 sum += i # послідовно додаємо  
  
 return sum  
  
  
print(get\_sum\_integer(3, 10))

Завдання 10.

def get\_quadrat\_integer(a, b):  
 sum = 0  
 if b < a:  
 return "invalid value b" #теж саме, що в 9 завданні,тільки в суму записуємо квадрат чисел  
 for i in range(a, b + 1):  
 sum += math.pow(i, 2)  
  
 return sum  
  
  
print(get\_quadrat\_integer(1, 3))

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

def get\_average():  
 sum = 0  
 a = input("Enter a:") #зчитуємо а  
 for i in range(int(a), 201): # від а до 200  
 print(i)  
 sum += int(a) # сума числе  
  
 return F"result:{sum / 200}" #середнє занчення  
  
  
print(get\_average())

Завдання 12. Знайти суму всіх цілих чисел від a до b (значення a і b вводяться з клавіатури; b ≥a). Вирішити задачу використовуючи циклічну конструкцію while.

def get\_sum\_for\_range(a, b):  
 sum = 0  
 if b < a:  
 return "invalid value b"  
 while a <= b: # поки а більше рівне б  
 sum += a # сумуємо елементи  
 a += 1 # інкрементумо лічильник  
 return sum  
  
  
print(get\_sum\_for\_range(2, 5))

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

def get\_quadratic(a):  
 sum = 0  
 if (a < 0 or a > 50):  
 return "error invalid value a" # якщо а не впадає в діапазон повертаємо помилку  
 for i in range(a, 51): # якщо все ок  
 print(i)  
 sum += math.pow(i, 2) # рахуємо суму квадратів  
 return sum  
  
  
c = int(input("Enter value:"))  
print(get\_quadratic(c))

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.

def get\_min\_number\_k(n):  
 i = 0  
 if n < 1:  
 return "error invalid value" #перевірка що Н більше 1  
  
 while math.pow(5, i) <= n: # поки 5 в K менше Н інкрементуємо і  
 i += 1  
  
 return i  
  
  
print(get\_min\_number\_k(12))

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

def get\_number\_for\_n(n):  
 l = 1  
 sum = 0  
 for i in iter(int, 1): #нескільнченний цикл  
 sum +=l #сума послідовності  
 l += 2 #інкрементація змінної л на 2  
 if sum > n: # якщо сума більше н, виходимо с цикла  
 break  
 return sum #повертаємо суму  
  
  
print(get\_number\_for\_n(1))

Завдання 16. Серед чисел 1 5 10, 17, 26, ... знайти перше число, більше n. Умовний оператор не використовувати. Вирішити задачу використовуючи циклічну конструкцію while.

def get\_number\_while\_n(n):  
 i = 1  
 sum = 0  
 while n >= sum: #поки н більше рівне сумі  
 sum += i  
 if (i == 1): # для 1 операції інкрементуємо і на 4  
 i += 4  
 i += 2 # для всіх інших на 2  
  
  
 return sum # повертаємо суму  
  
  
print(get\_number\_while\_n(12))