День 1 - Начальные задачи по Python (ветвление и циклы)

Задача 1: Определение положительного, отрицательного или нулевого числа

```
num = float(input("Введите число: "))

if num > 0:
    print(f"{num} является положительным числом.")
elif num < 0:
    print(f"{num} является отрицательным числом.")
else:
    print(f"{num} равен нулю.")
```

Задача 2: Определение наибольшего из трех чисел

Методические указания:

1. Попросите пользователя ввести три числа.

2. Используйте ряд операторов `if-elif-else` для сравнения трёх чисел и определения наибольшего.

3. Выведите наибольшее число на консоль.

Задача 3: Подсчитать цифры в числе

Методические указания:

1. Попросите пользователя ввести число.

2. Инициализируйте переменную для отслеживания количества цифр.

3. Используйте `while`-цикл для последовательного извлечения последней цифры числа и увеличения счётчика цифр.

4. Выведите итоговое количество цифр на консоль.

День 2 - Промежуточные задачи по Python (ветвление и циклы)

Задача 1: Определение високосного года

```
year = int(input("Введите год: "))
if year % 4 == 0:
    if year % 100 == 0:
        if year % 400 == 0:
            print(f"{year} является високосным годом.")
        else:
            print(f"{year} не является високосным годом.")
    else:
        print(f"{year} является високосным годом.")
else:
    print(f"{year} не является високосным годом.")
```

Задача 2: Вывод последовательности Фибоначчи до заданного числа

Методические указания:

1. Попросите пользователя ввести количество членов последовательности Фибоначчи, которые нужно вывести.

2. Инициализируйте первые два члена последовательности (0 и 1).

3. Используйте `for`-цикл для генерации оставшихся членов последовательности, выводя каждый из них.

4. Следующий член последовательности равен сумме двух предыдущих.

Задача 3: Вычисление факториала числа

Методические указания:

1. Попросите пользователя ввести число.

2. Инициализируйте переменную для хранения факториала.

3. Используйте `for`-цикл для умножения переменной факториала на каждое число от 1 до введённого пользователем.

4. Выведите итоговое значение факториала на консоль.

День 3 - Промежуточные задачи по Python (ветвление и циклы)

Задача 1: Обращение строки

input_string = input("Введите строку: ")

reversed_string = input_string[::-1]

print(f"Обращённая строка: {reversed_string}")

Задача 2: Нахождение наибольшего общего делителя (НОД) двух чисел

Методические указания:

1. Попросите пользователя ввести два числа.

2. Используйте `while`-цикл для нахождения НОД двух чисел.

3. НОД - это наибольшее положительное целое число, которое делит оба числа без остатка.

4. Выведите НОД на консоль.

Задача 3: Проверка на простоту числа

Методические указания:

1. Попросите пользователя ввести число.

2. Используйте `for`-цикл для проверки, делится ли число на любое число от 2 до квадратного корня из этого числа.

3. Если число делится только на 1 и само на себя, то оно является простым.

4. Выведите, является ли число простым или нет.

День 4 - Промежуточные задачи по Python (ветвление и циклы)

Задача 1: Конвертация градусов Цельсия в Фаренгейт

celsius = float(input("Введите температуру в градусах Цельсия: "))

fahrenheit = (celsius * 9/5) + 32

print(f"{celsius}°C равно {fahrenheit}°F.")

Задача 2: Нахождение суммы цифр в числе

Методические указания:

1. Попросите пользователя ввести число.

2. Инициализируйте переменную для хранения суммы цифр.

3. Используйте `while`-цикл для последовательного извлечения последней цифры числа и добавления её к сумме.

4. Выведите итоговую сумму цифр на консоль.

Задача 3: Проверка числа на принадлежность к числам Армстронга

Методические указания:

1. Попросите пользователя ввести число.

2. Вычислите количество цифр в числе.

3. Инициализируйте переменную для хранения суммы цифр, возведённых в степень, равную количеству цифр.

4. Используйте `while`-цикл для последовательного извлечения цифр числа и добавления их кубов к сумме.

5. Проверьте, равна ли сумма оригинальному числу, и выведите, является ли число числом Армстронга.

День 5 - Продвинутые задачи по Python (ветвление и циклы)

Задача 1: Реализация калькулятора

```
def calculator():
    print("Добро пожаловать в Python-калькулятор!")
    print("Доступные операции: +, -, *, /, %")
    while True:
        operator = input("Введите операцию (+, -, *, /, %): ")
        if operator in ['+', '-', '*', '/', '%']:
            num1 = float(input("Введите первое число: "))
            num2 = float(input("Введите второе число: "))
            if operator == '+':
                result = num1 + num2
            elif operator == '-':
                result = num1 - num2
            elif operator == '*':
                result = num1 * num2
            elif operator == '/':
                if num2 == 0:
                    print("Ошибка: деление на ноль")
                    continue
                result = num1 / num2
            elif operator == '%':
                result = num1 % num2
            print(f"Результат: {result}")
        else:
            print("Неверный оператор. Попробуйте ещё раз.")
```

calculator()

Задача 2: Генерация таблицы умножения для числа

Методические указания:

1. Попросите пользователя ввести число.

2. Используйте `for`-цикл для итерации от 1 до 10.

3. На каждой итерации умножьте число, введённое пользователем, на текущее число итерации и выведите результат.

4. Это сгенерирует таблицу умножения для заданного числа.

Задача 3: Реализация игры "Угадай число"

Методические указания:

1. Сгенерируйте случайное число в определённом диапазоне (например, от 1 до 100).

2. Попросите пользователя угадать число.

3. Используйте `while`-цикл, чтобы продолжать игру, пока пользователь не угадает число.

4. Внутри цикла сравните предположение пользователя со случайным числом и предоставьте обратную связь (слишком высоко, слишком низко или правильно).

5. Ведите учёт количества попыток и выведите итоговый результат, когда пользователь угадает правильно.

День 6 - Продвинутые задачи по Python (ветвление и циклы)

Задача 1: Реализация гипотезы Коллатца

```
def collatz(n):
    steps = 0
    while n != 1:
        if n % 2 == 0:
            n = n // 2
        else:
            n = 3 * n + 1
        steps += 1
    return steps
num = int(input("Введите число: "))
result = collatz(num)
print(f"Количество шагов для достижения 1: {result}")
```

Задача 2: Реализация Ханойской башни

Методические указания:

1. Попросите пользователя ввести количество дисков.

2. Определите функцию, которая рекурсивно решает задачу Ханойской башни.

3. Функция должна перемещать диски с исходного стержня на целевой стержень, используя вспомогательный стержень.

4. Выведите шаги, необходимые для решения задачи.

Задача 3: Реализация решета Эратосфена

Методические указания:

1. Попросите пользователя ввести верхний предел для диапазона простых чисел.

2. Создайте булевый список для отметки всех чисел как потенциально простых.

3. Используйте `for`-цикл для итерации по числам и пометки их кратных как непростых.

4. Выведите все найденные простые числа в диапазоне.