ПРОГРАММИРОВАНИЕ НА РҮТНОМ

ДОМАШНЕЕ ЗАДАНИЕ №2

TEMA: УСЛОВНЫЕ КОНСТРУКЦИИ IF-ELIF-ELSE

ЗАДАЧА 1

Напишите программу, которая определяет, является число чётным или нечётным.

Пример ввода:

10

Пример вывода:

Чётное

ЗАДАЧА 2

Напишите программу, которая по введённому возрасту пользователя сообщает, к какой возрастной группе он относится:

- до 13 включительно детство;
- от 14 до 24 молодость;
- от 25 до 59 зрелость;
- от 6о старость.

Пример ввода:

91

Пример вывода:

старость

ЗАДАЧА 3

Вводятся два целых числа не равных нулю. Напишите программу, которая проверяет делится ли первое число на второе нацело. Программа выводит сообщение об этом, а также частное и остаток, если он есть.

Для вывода используйте f-строку.

Пример ввода 1:

8

2

Пример вывода 1:

8 делится на 2 нацело

Частное: 4

Пример ввода 2:

10

3

Пример вывода 2:

10 не делится на 3 нацело

Частное: 3 Остаток: 1

ЗАДАЧА 4

Напишите программу, которая считывает три числа и подсчитывает сумму только положительных чисел.

Пример ввода:

4

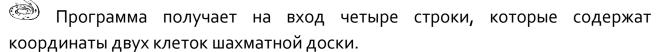
-22

1

Пример вывода:

5

ЗАДАЧА 5



По горизонтали клетка кодируется латинскими буквами от 'a' до 'h'.

По вертикали клетка кодируется цифрами от 1 до 8.

Если две заданные клетки покрашены в один цвет, то программа выводит слово "ДА". Если клетки покрашены в разные цвета, то должно быть напечатано слово "HET".

Пример ввода:

a

1

b

2

Пример вывода:

ДА

Сохраните код к каждой задаче в отдельный файл с расширением .py.
Упакуйте эти файлы в архив и прикрепите его в MyStat в качестве ответа.

подсказки

1. Проверка типа данных в условной конструкции должна выполняться с использованием ключевого слова is:

```
>>> t = type("τeκcτ")
>>> if t is str:
... print("Y")
...
```

2. Внутри условия можно использовать вызовы функций:

```
>>> if type(12) is int:
... print("число")
...
число
```

Однако, обратите внимание, что результат такого вызова функции (возвращаемое значение) никуда не сохраняется.

3. Если в блоке условной конструкции у вас только одна команда и нет других вложенных блоков, то эту команду можно писать в той же строке, где и условие, с пробелом после двоеточия:

```
>>> if input() == "abc": print("ABC")
...
abc
ABC
```

4. Блоки if могут вкладываться друг в друга — это необходимо для выполнения дополнительной проверки после прохождения первой:

```
>>> a = input("Введите число: ")
Введите число: 0
>>> if a.isdigit():
... if int(a) != 0: print(1/int(a))
... else: print("На ноль делить нельзя")
... else: print("Строка содержит не только цифры")
...
На ноль делить нельзя
```

isdigit() это метод проверки символов строки на то, все ли они являются цифрами. Возвращает True, если в строке содержатся только символы цифр.

5. Напоминалка про то, как применяются f-строки:

```
>>> n, o = "четыре", "две"
>>> print(f"Сегодня здесь {n} новых и {o} старых подсказки")
Сегодня здесь четыре новых и две старых подсказки
```

6. У функции print есть два скрытых аргумента, которые можно изменить. Первый называется sep и отвечает за разделитель, который функция вставляет при выводе между несколькими аргументами – по умолчанию это пробел:

```
>>> print(1, 2)
1 2
>>> print(1, 2, sep='!')
1!2
>>> print(1, 2, sep='\n')
1
2
```

```
>>> print("\nПустые строки сверху и снизу", end='\n\n')
Пустые строки сверху и снизу
>>> print("Без переноса строки", end=' ')
Без переноса строки >>>
```