1

Повторение (упражнение)

Напишите программу, которая считывает температуру в градусах Цельсия и выводит "ХОЛОДНО", если темпер
"ЖАРКО", если температура больше 28, и "НОРМАЛЬНО" в остальных случаях.

```
# код
# Считываем температуру в градусах Цельсия
температура = float(input("Введите температуру в градусах Цельсия: "))

# Проверяем условия и выводим результат
if температура < 15.5:
    print("ХОЛОДНО")
else:
    if температура > 28:
        print("ЖАРКО")
    else:
        print("НОРМАЛЬНО")
```

```
Введите температуру в градусах Цельсия: 12
ХОЛОДНО
```

2

Понимать код примера выше не требуется. Его мы разберем позже. Однако он позволит вам познакомиться с особенностями попеременного ввода-вывода.Вы ввели, он вывел, вы ввели... Ну, вы поняли.

Ваша задача - написать программу, на основе цикла `while`, задав условие выполнения цикла так, чтобы при введении нуля он прекращал свое выполнение.

```
# тут код
# Используем цикл while для ввода чисел
while True:
    number = int(input())  # Считываем число
    if number == 0:        # Если введен ноль, завершаем цикл
        break
    print(number)          # Выводим число, если оно не ноль
```

```
2
2
0
```

1*

Формат ввода

Вводятся строки одна за другой.

Формат вывода

Введенные строки до пустой

P.S. Напоминаем, что пустая строка - это `""` или `''`. Чтобы ее ввести, достаточно оставить поле ввода пустым и нажать `Enter`.

```
# тут код
# Используем цикл while для ввода строк
while True:
    строка = input()  # Считываем строку
    if строка == "":  # Если введена пустая строка, завершаем цикл
        break
    print(строка)     # Выводим строку, если она не пустая
```

```
ewe
ewe
```

Вводятся числа одно за другим, каждое на отдельной строке.

Формат вывода

Введеные числа, каждое на отдельной строке, пока введенные числа делятся на 10

P.S. %

```
# тут код
# Используем цикл while для ввода чисел
while True:
    число = int(input())  # Считываем число
    if число % 10 != 0:   # Если число не делится на 10, завершаем цикл
        break
    print(число)          # Выводим число, если оно делится на 10
```

```
10
10
10
10
30
30
11
```

2**

Несколько строк — результат проверки пар паролей.

```
while True:
    # Считываем пароль и его подтверждение
    пароль1 = input()
    пароль2 = input()

    # Проверка на длину пароля
    if len(пароль1) < 8:
        print("Короткий!")
        continue  # Переходим к следующей итерации цикла

    # Проверка на наличие "123" в пароле
    if "123" in пароль1:
        print("Простой!")
        continue  # Переходим к следующей итерации цикла

    # Проверка на совпадение паролей
    if пароль1 != пароль2:
        print("Различаются.")
        continue  # Переходим к следующей итерации цикла

    # Если все проверки пройдены, выводим "OK" и завершаем цикл
    print("OK")
    break
```

```
qwiejioqwej123
qwiejioqwej123
Простой!
qwiejioqwej12
qwiejioqwej12e
Различаются.
qwiejioqwej1222
qwiejioqwej1222
OK
```

3*

Иными словами, найдите первую цифру введённого числа при записи его в системе счисления с

Формат ввода

Одно целое число — изначальное количество монет у Учителя.

Формат вывода

Одно целое число — количество монет, которое останется у Учителя в конце.

```python
# тут код
# Считываем количество монет
монеты = int(input())

# Пока число монет больше или равно 8, делим его на 8
while монеты >= 8:
    монеты = монеты // 8

# Выводим оставшееся количество монет
print(монеты)
```

```
43
5
```

3

-1

Вывод

2295.35

```python
# тут код
# Инициализируем переменную для хранения общей суммы
общая_стоимость = 0.0

# Бесконечный цикл для ввода цен
while True:
    цена = float(input())  # Считываем цену товара
    if цена < 0:           # Если введено отрицательное число, завершаем цикл
        break
    if цена > 1000:        # Если цена больше 1000, применяем скидку 5%
        цена *= 0.95
    общая_стоимость += цена  # Добавляем цену к общей стоимости

# Выводим общую стоимость с учётом скидок
print(общая_стоимость)
```

```
1230
123
321
123
-102
1735.5
```

4

Две строки: количество кандидатур на первой, и минимальный и максимальный рост через пробел -

```python
#тут код
# Инициализируем список для хранения подходящих ростов
подходящие_росты = []

# Бесконечный цикл для ввода ростов
while True:
    ввод = input()  # Считываем ввод
    if ввод == "!":  # Если введен "!", завершаем цикл
        break
    рост = int(ввод)  # Преобразуем ввод в число
    if 150 <= рост <= 190:  # Проверяем, подходит ли рост
        подходящие_росты.append(рост)  # Добавляем в список подходящих

# Выводим количество подходящих кандидатов
print(len(подходящие_росты))

# Выводим минимальный и максимальный рост
print(min(подходящие_росты), max(подходящие_росты))
```

```
191
145
187
151
!
2
151 187
```

4**

Формат вывода

Выводится одно число — количество шагов, необходимое стартующей от n си
до 1.

```
# Считываем начальное число
n = int(input())

# Инициализируем счетчик шагов
шаги = 0

# Пока число не станет равным 1
while n != 1:
    if n % 2 == 0:  # Если число четное
        n = n // 2
    else:           # Если число нечетное
        n = 3 * n + 1
    шаги += 1  # Увеличиваем счетчик шагов

# Выводим количество шагов
print(шаги)
```

```
12
9
```

1

```
# тут код\
# Считываем символ-разделитель
разделитель = input()

# Считываем три слова
слово1 = input()
слово2 = input()
слово3 = input()

# Выводим слова через разделитель, используя параметр sep
print(слово1, слово2, слово3, sep=разделитель)
```

```
+
1
2
3
1+2+3
```

2

```python
# тут код
# Считываем изречение
изречение = input()

# Считываем количество повторений
количество = int(input())

# Выводим изречение нужное количество раз
for _ in range(количество):
    print(изречение)
```

```
ыыыыы
3
ыыыыы
ыыыыы
ыыыыы
```

3

Формат ввода

Вводится одно целое число `n≥0`.

Формат вывода

Выводится строка целых чисел через пробел.

```python
# Считываем число n
n = int(input())

# Генерируем числа от 0 до n и выводим их через пробел
print(" ".join(map(str, range(n + 1))))
```

```
8
0 1 2 3 4 5 6 7 8
```

4

```python
# тут код
# Считываем число n
n = int(input())

# Генерируем и выводим фразы для каждого числа от 0 до n
for i in range(n + 1):
    print(f"Куб числа {i} равен {i ** 3}")
```

```
3
Куб числа 0 равен 0
Куб числа 1 равен 1
Куб числа 2 равен 8
Куб числа 3 равен 27
```

5

```python
# тут код
# Считываем число n
n = int(input())

# Инициализируем переменную для хранения факториала
факториал = 1

# Вычисляем факториал
for i in range(1, n + 1):
    факториал *= i

# Выводим результат
print(факториал)
```

```
0
1
```

6

```python
# тут код
# Инициализируем переменную для хранения произведения
произведение = 1

# Считываем 6 чисел
for _ in range(6):
    число = int(input())
    if число != 0:  # Игнорируем нули
        произведение *= число

# Выводим результат
print(произведение)4
```

```
1
2
3
4
5
6
720
```

1*

Для каждого из введенных чисел сообщение «ДА» или «НЕТ» на отдельной строчке. title

```python
# Перебираем числа от 0 до 16
for i in range(17):
    # Считываем число d
    d = int(input())

    # Проверяем, делится ли i на d
    if d == 0:
        print("НЕТ")  # На ноль делить нельзя
    elif i % d == 0:
        print("ДА")
    else:
        print("НЕТ")
```

```
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
1
ДА
```

2**

title

```python
# тут код
# Считываем количество чисел
n = int(input())

# Инициализируем сумму и флаг для знака
сумма = 0
знак = 1  # Начинаем с плюса

# Считываем числа и вычисляем знакочередующуюся сумму
for _ in range(n):
    число = int(input())
    сумма += знак * число
    знак *= -1  # Меняем знак на противоположный

# Выводим результат
print(сумма)
```

```
3
1
2
3
2
```

7

1

5

Вывод

1 6 11 16 21 26 31

```python
# код решения
# Считываем начальный день и шаг
n = int(input())
m = int(input())

# Генерируем и выводим дни с шагом m
print(" ".join(map(str, range(n, 32, m))))
```

```
1
5
1 6 11 16 21 26 31
```

3**

```python
# код решения
# Считываем количество тестируемых людей
n = int(input())

# Инициализируем список для хранения IQ
iq_list = []

# Перебираем всех людей
for _ in range(n):
    # Считываем IQ текущего человека
    iq = int(input())

    # Если список не пустой, вычисляем средний IQ предшественников
    if iq_list:
        средний_iq = sum(iq_list) / len(iq_list)
        if iq > средний_iq:
            print(">")
        elif iq < средний_iq:
            print("<")
        else:
            print("0")
    else:
        # Для первого человека выводим 0
        print("0")

    # Добавляем IQ текущего человека в список
    iq_list.append(iq)
```

```
4
100
0
101
>
80
<
99
>
```

Alpha

```python
[42] # код решения
# Считываем количество конфет
S = int(input())

# Перебираем возможные значения n (количество родственников)
n = 1
while True:
    # Вычисляем X по формуле
    numerator = 2 * S - n * (n - 1)
    denominator = 2 * n

    # Если числитель меньше нуля, дальше перебирать бессмысленно
    if numerator < 0:
        break

    # Если X целое и положительное, выводим его
    if numerator % denominator == 0:
        X = numerator // denominator
        if X > 0:
            print(X)
            break

    # Увеличиваем количество родственников
    n += 1
```

```
32
32
```

Omega

```python
# код решения
# Функция для нахождения НОД (алгоритм Евклида)
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

# Считываем количество дробинок
n = int(input())

# Инициализируем числитель и знаменатель суммы
числитель_суммы = 0
знаменатель_суммы = 1

# Перебираем все дробинки
for _ in range(n):
    # Считываем числитель и знаменатель текущей дробинки
    числитель = int(input())
    знаменатель = int(input())

    # Находим общий знаменатель
    общий_знаменатель = знаменатель_суммы * знаменатель
    новый_числитель = числитель_суммы * знаменатель + числитель * знаменатель_суммы

    # Упрощаем дробь
    общий_делитель = gcd(новый_числитель, общий_знаменатель)
    числитель_суммы = новый_числитель // общий_делитель
    знаменатель_суммы = общий_знаменатель // общий_делитель

# Выводим итоговую дробь
print(f"{числитель_суммы}/{знаменатель_суммы}")3
```

```
3
4
2
3
4
1
2
13/4
```