

Раздел 1. Поиск максимума/минимума с ограничениями. Заданное количество чисел/ввод до условного значения.

1.1. Напишите программу, которая в последовательности натуральных чисел определяет **максимальное число, кратное 5**. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 5. Количество чисел не превышает 1000. Введенные числа не превышают 30 000. Программа должна вывести одно число – максимальное число, кратное 5.

Пример работы программы:

Входные данные	Выходные данные
3 10 25 12	25

Решение

```
n = int(input())
maxi = 0
for i in range(n):
    a = int(input())
    if a % 5 == 0 and a > maxi:
        maxi = a
print(maxi)
```

1.2 Напишите программу, которая в последовательности натуральных чисел определяет **максимальное число, оканчивающееся на 3**. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 3. Количество чисел не превышает 1000. Введенные числа не превышают 30 000. Программа должна вывести одно число – максимальное число, оканчивающееся на 3.

Пример работы программы:

Входные данные	Выходные данные
3 13 23 3	23

Решение

```
n = int(input())
maxi = 3
for i in range(n):
    a = int(input())
    if a % 10 == 3 and a > maxi:
        maxi = a
print(maxi)
```

1.3. Напишите программу, которая в последовательности натуральных чисел определяет **минимальное число, оканчивающееся на 6**. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 6. Количество чисел не превышает 1000. Введенные числа не превышают 30 000. Программа должна вывести одно число – минимальное число, оканчивающееся на 6.

Пример работы программы:

Входные данные	Выходные данные
3 26 16 36	16

Решение

```
n = int(input())
mini = 30000
for i in range(n):
    a = int(input())
    if a % 10 == 6 and a < mini:
        mini = a
print(mini)
```

1.4. Напишите программу, которая в последовательности натуральных чисел определяет **минимальное двузначное число**. Программа получает на вход количество чисел в последовательности, а затем сами числа. Если в последовательности нет ни одного двузначного числа, вывести «Двузначные числа не обнаружены». Количество чисел не превышает 1000. Введенные числа не превышают 30 000. Программа должна вывести одно число – минимальное двузначное число или строку «Двузначные числа не обнаружены».

Пример работы программы:

Входные данные	Выходные данные
5 3 26 16 36 200	16
2 3 200	Двузначные числа не обнаружены

Решение

```
n = int(input())
mini = 100
for i in range(n):
    x = int(input())
    if x >= 10 and x < mini:
        mini = x
if mini == 100:
    print('Двузначные числа не обнаружены')
else:
    print(mini)
```

1.5. Напишите программу, которая в последовательности натуральных чисел определяет **«второй максимум»** (число, которое было бы максимальным, если бы максимальное число отсутствовало). Программа вводит числа до тех пор, пока не будет введен ноль. Гарантируется, что до ноля будет не менее двух натуральных чисел.

Пример работы программы:

Входные данные	Выходные данные
2 3 0	2
3 4 5 6 0	5

Решение

```
f = int(input())
s = int(input())
if s > f:
    f, s = s, f
n = int(input())
while n != 0:
    if n >= f:
        f, s = n, f
    elif n > s:
        s = n
    n = int(input())
print(s)
```

Раздел 2. Поиск сумм / количеств с ограничениями в заданном количестве чисел

2.1. Напишите программу, которая в последовательности натуральных чисел определяет **сумму чисел, кратных 6**. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 6. Количество чисел не превышает 100. Введенные числа не превышают 300. Программа должна вывести одно число – сумму чисел, кратных 6.

Пример работы программы:

Входные данные	Выходные данные
3 12 25 6	18

Решение

```
n = int(input())
sumi = 0
for i in range(n):
    a = int(input())
    if a % 6 == 0:
        sumi += a
print(sumi)
```

2.2. Напишите программу, которая в последовательности натуральных чисел определяет **количество чисел, кратных 6**. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 6.

Пример работы программы:

Входные данные	Выходные данные
3 18 26 24	2

Решение

```
n = int(input())
kol = 0
for i in range(n):
    a = int(input())
    if a % 6 == 0:
        kol += 1
print(kol)
```

2.3. Напишите программу, которая в последовательности натуральных чисел определяет **сумму чисел, оканчивающихся на 4**. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 4. Количество чисел не превышает 1000. Введенные числа не превышают 30 000. Программа должна вывести одно число – сумму чисел, оканчивающихся на 4.

Пример работы программы:

Входные данные	Выходные данные
3 14 25 24	38

Решение

```
n = int(input())
sumi = 0
for i in range(n):
    a = int(input())
    if a % 10 == 4:
        sumi += a
print(sumi)
```

2.4. Напишите программу, которая в последовательности натуральных чисел определяет **количество чисел, оканчивающихся на 3**. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 3. Количество чисел не превышает 1000. Введенные числа не превышают 30 000. Программа должна вывести одно число – количество чисел, оканчивающихся на 3.

Пример работы программы:

Входные данные	Выходные данные
3 13 23 24	2

Решение

```
n = int(input())
kol = 0
for i in range(n):
    a = int(input())
    if a % 10 == 3:
        kol += 1
print(kol)
```

Раздел 3. Поиск суммы/количества чисел по условию с вводом чисел до заданного значения (с неизвестным заранее количеством)

3.1. Напишите программу, которая в последовательности натуральных чисел определяет **сумму всех чисел, кратных 6 и оканчивающихся на 4**. Программа получает на вход натуральные числа, количество введенных чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность). Количество чисел не превышает 1000. Введенные числа не превышают 30 000. Программа должна вывести одно число: сумму всех чисел, кратных 6 и оканчивающихся на 4.

Пример работы программы:

Входные данные	Выходные данные
14 24 144 22 12 0	168

Решение

```
sumi = 0
a = 1
while a != 0:
    a = int(input())
    if a % 6 == 0 and a % 10 == 4:
        sumi += a
print(sumi)
```

3.2. Напишите программу, которая в последовательности целых чисел определяет **количество нечетных чисел, кратных 3**. Программа получает на вход целые числа, количество введенных чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность). Количество чисел не превышает 1000. Введенные числа по модулю не превышают 30 000. Программа должна вывести два числа: длину последовательности (завершающий 0 не учитывается) и количество нечетных чисел, кратных 3.

Пример работы программы:

Входные данные	Выходные данные
4	
6	
15	4
180	1
0	

Решение

```
n = 0
num = 0
a = int(input())
while a != 0:
    num += 1
    if (a % 2 != 0) and (a % 3 == 0):
        n += 1
    a = int(input())
print(num)
print(n)
```

3.3. Напишите программу, которая в последовательности натуральных чисел определяет количество **трехзначных чисел, кратных 4**. Программа получает на вход натуральные числа, количество введенных чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введенные числа не превышают 30 000. Программа должна вывести одно число: количество трехзначных чисел, кратных 4.

Пример работы программы:

Входные данные	Выходные данные
120	
9	
365	1
4	
0	

Решение

```
count = 0
a = 1
while a != 0:
    a = int(input())
    if a == 0:
        break
    if a % 4 == 0 and 99 < a < 1000:
        count += 1
print(count)
```

3.4. Напишите программу, которая в последовательности целых чисел определяет **количество чисел, кратных 5 или 9**. Программа получает на вход целые числа, количество введенных чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность). Количество чисел не превышает 1000. Введенные числа по модулю не превышают 30 000. Программа должна вывести одно число: количество чисел, кратных 5 или 9.

Пример работы программы:

Входные данные	Выходные данные
15 91 90 0	2

Решение

```
count = 0
a = 1
while a != 0:
    a = int(input())
    if a == 0:
        break
    if a % 5 == 0 or a % 9 == 0:
        count += 1
print(count)
```

Раздел 4. Поиск суммы/количества чисел / ввод чисел (фильтрация ввода) по условию до достижения заданного условия (с неизвестным заранее количеством)

4.1 Вводить целые числа до тех пор, пока очередное число не окажется неотрицательным. Сообщить о достижении неотрицательного числа.

Пример работы программы:

Входные данные	Выходные данные
-2 -55 1	Неотрицательное число

Решение

```
x = int(input())
while x < 0:
    x = int(input())
else:
    print('Неотрицательное число')
```

4.2. Вводить целые числа до тех пор, пока очередное число не окажется меньше предыдущего. Сообщить **количество введенных чисел**. Гарантируется, что будет введено не менее двух чисел и среди них будет число, меньшее предыдущего.

Пример работы программы:

Входные данные	Выходные данные
2 4 4 3	<i>Введено чисел 4</i>

Решение

```
f = int(input())
s = int(input())
k = 2
while f <= s:
    f, s = s, int(input())
    k += 1
print('Введено чисел', k)
```

4.3. Вводить целые числа до тех пор, пока не накопится **сумма не менее 100**. Сообщить значение накопленной суммы.

Пример работы программы:

Входные данные	Выходные данные
1 2 40 60	<i>Накопилось 103</i>
105	<i>Накопилось 105</i>

Решение

```
x = int(input())
s = x
while s < 100:
    print('Недостаточно')
    x = int(input())
    s += x
else:
    print('Накопилось', s)
```

Раздел 5. Работа с цифрами числа в произвольной системе счисления

5.1. Найдите количество четных цифр в восьмеричной записи числа, введенного с клавиатуры.

Пример работы программы:

Входные данные	Выходные данные
225	1 <i>примечание:</i> $225 = 341(8)$

Решение

```
x = int(input())
k = 0
if x == 0:
    k = 1
while x > 0:
    if x % 8 % 2 == 0:
        k += 1
    x //= 8
print(k)
```

5.2. Найдите сумму цифр в записи натурального числа n , введенного с клавиатуры в n -ричной системе счисления, $1 < n < 20$.

Пример работы программы:

Входные данные	Выходные данные
226 8	9 <i>примечание:</i> $226 = 342(8)$
15 16	15

Решение

```
x = int(input())
n = int(input())
s = 0
while x > 0:
    s += x % n
    x //= n
print(s)
```

5.3. Найдите запись натурального числа x в n -ричной системе счисления, $1 < n < 10$. Значения x , n вводятся с клавиатуры.

Пример работы программы:

Входные данные	Выходные данные
225 8	341

Решение

```
x = int(input())
n = int(input())
s = ''
while x > 0:
    print(x % n)
    s = str(x % n) + s
    x //= n
print(s)
```

Раздел 6. Поиск делителей числа, НОД, НОК

6.1. Найдите количество нетривиальных делителей натурального числа, введенного с клавиатуры (нетривиальными называются делители, отличные от 1 и от самого числа).

Пример работы программы:

Входные данные	Выходные данные
16	3 <i>примечание: 2 4 8</i>
24	4

Решение

```
x = int(input())
k = 0
b = int(x ** 0.5)
if x == b ** 2:
    k = 1
for d in range(2, b):
    if x % d == 0:
        k += 2
print(k)
```

6.2. Найдите все простые делители натурального числа, введенного с клавиатуры.

Пример работы программы:

Входные данные	Выходные данные
12	2 2 3
2	2

Решение

```
x = int(input())
d = 2
while x > 1:
    if x % d == 0:
        print(d, end=' ')
        x //= d
    else:
        d += 1
```

6.3. Найдите наибольший простой делитель натурального числа $x > 1$, введенного с клавиатуры.

Пример работы программы:

Входные данные	Выходные данные
12	3
2	2

Решение

```
x = int(input())
d = 2
md = 2
while x > 1:
    if x % d == 0:
        md = d
        x //= d
    else:
        d += 1
print(md)
```

6.4. Найдите **наибольший общий делитель** двух натуральных чисел, введенных с клавиатуры. **Алгоритм Евклида.**

Пример работы программы:

Входные данные	Выходные данные
12 15	3
12 4	4
12 17	1

Решение

```
a = int(input())
b = int(input())
while a != b:
    if a > b:
        a -= b
    else:
        b -= a
print(a)
```

Решение сокращенным алгоритмом

```
a = int(input())
b = int(input())
while a * b != 0:
    a, b = b, a % b
print(a + b)
```

6.5. Найдите **наименьшее общее кратное** двух натуральных чисел, введенных с клавиатуры.

Пример работы программы:

Входные данные	Выходные данные
12 15	60
12 4	12
12 17	204

Решение

```
a = int(input())
b = int(input())
nok = a * b
while a != b:
    if a > b:
        a -= b
    else:
        b -= a
print(nok // a)
```

Раздел 7. Вывод табличных данных, числовых последовательностей

7.1. Для заданного числа x выведите таблицу умножения от 1 до 9 в удобном для чтения виде (см. пример работы программы).

Пример работы программы:

Входные данные	Выходные данные
7	7 * 1 = 7 7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49 7 * 8 = 56 7 * 9 = 63

Решение

```
x = int(input())
for i in range(1, 10):
    print(x, '*', i, '=', x * i)
```

7.2. Написать программу, которая вводит число, на которое приходится первый понедельник марта и выводит даты всех понедельников марта в одну строку через пробел.

Пример работы программы:

Входные данные	Выходные данные
2	Даты понедельников в марте: 2 9 16 23 30

Решение

```
x = int(input())
print('Даты понедельников в марте:')
for d in range(x, 32, 7):
    print(d, end=' ')
```

7.3. Написать программу, которая вводит для натуральных числа a и b ($0 < a < b$) и выводит все числа, кратные 3 в диапазоне от a до b включительно.

Пример работы программы:

Входные данные	Выходные данные
2 12	3 6 9 12

Решение

```
a = int(input())
b = int(input())
if a % 3 != 0:
    a += (3 - a % 3)
for x in range(a, b + 1, 3):
    print(x, end=' ')
```