

20 (повышенный уровень, время – 5 мин)

Тема: Анализ программы, содержащей подпрограммы, циклы и ветвления.

Что нужно знать:

- операции целочисленного деления (`//`) и взятия остатка (`%`)
- как работают операторы присваивания, циклы и условные операторы в языке программирования

Пример задания:

P-10. Ниже приведён алгоритм. Укажите наименьшее число x , при вводе которого алгоритм напечатает сначала 2, потом – 5.

```
x = int(input())
a = 0
b = 1
while x > 0:
    if x % 2 > 0:
        a += 1
    else:
        b += (x % 7)
    x //= 7
print(a)
print(b)
```

Решение:

- 1) видим, что в конце программы на экран выводятся переменные **a** и **b**
- 2) переменная **a** в начале равна нулю, затем при выполнении условия $x \% 2 > 0$ увеличивается на 1, то есть **a** – счётчик
- 3) переменная **b** в начале равна 1, затем при нарушении условия $x \% 2 > 0$ к ней добавляется $(x \% 7)$ – последняя цифра записи числа **x** в системе счисления с основанием 7

- 4) в цикле

```
while x > 0:
    ...
    x //= 7
```

значение переменной **x** делится на 7, пока число не станет равно 0; это значит, что от его семеричной записи по очереди отсекаются цифры, начиная с последней

- 5) изменение переменных **a** и **b** выполняется в условном операторе

```
if x % 2 > 0:
    a += 1
else:
    b += (x % 7);
```

то есть если после очередного отсекаемого получилось нечётное число, увеличивается счётчик **a**, а если получилось чётное – к значению переменной **b** добавляется последняя цифра семеричной записи числа

- 6) поскольку фактически идёт работа с семеричной системой счисления, будем искать цифры нужного числа **x** в семеричной системе, а потом переведём его в десятичную систему
- 7) значение **a** – это количество нечётных чисел, полученных в процессе отсекаемого, а $b = 1 +$ сумма последних цифр семеричной записи чётных чисел, полученных в процессе отсекаемого

- 8) поскольку основание системы нечётное, чётность числа зависит от чётности цифр: если число нечётных цифр чётное (0, 2, 4, ...), то всё число чётное, а если в семеричной записи числа нечётное число нечётных цифр, то число нечётное
- 9) поскольку счётчик a увеличивается два раза им значение b увеличивается, по крайней мере, один раз, сначала будем искать минимальное число среди трёхзначных чисел (в семеричной системе)
- 10) нам нужно расставить чётные и нечётные цифры в семеричной записи числа так, чтобы из получаемых в процессе отсечения чисел два были нечётными, а одно – чётным
- 11) **маской чётности** числа назовём последовательность нулей и единиц, в которой ноль обозначает чётную цифру, а единица – нечётную; например, маска 101 обозначает последовательность «нечётная – чётная – нечётная»
- 12) при добавлении в конец семеричной записи числа новой нечётной цифры (1 в маске) чётность меняется, а при добавлении чётной (0 в маске) – нет, поэтому исходное число с маской 010 в ходе работы алгоритма как раз даст два нечётных числа (с масками 010 и 01) и одно чётное (с маской 0)
- 13) так как $b=5$, последняя цифра единственного чётного числа равна $b-1=4$, то есть первая цифра числа в семеричной системе – 4
- 14) продолжаем строить минимальное число по маске 010: вторая цифра должна быть нечётной, минимальная нечётная цифра – 1
- 15) третья цифра чётная, минимальное возможное значение – 0
- 16) таким образом, получаем число $410_7 = 203_{10}$
- 17) Ответ: **203**.

Ещё пример задания:

P-09. Ниже приведён алгоритм. Укажите наибольшее число x , при вводе которого алгоритм напечатает сначала 3, потом – 6.

```
x = int(input())
Q = 15
L = 0
while x >= Q:
    L += 1
    x -= Q
M = x
if M < L:
    M = L
    L = x
print(L)
print(M)
```

Решение:

1. видим, что в конце программы на экран выводятся переменные L и M
2. узнаём алгоритм в первой части программы:

```
L = 0
while x >= Q:
    L += 1
    x -= Q
```

Этот алгоритм вычисляет (с помощью последовательных вычитаний) частное L и остаток x от деления исходного значения x на Q .

3. Узнаём второй алгоритм:

```

M = x
if M < L:
    M = L
    L = x

```

Сначала то в **M** просто записывается значение **x** – остаток от деления исходного значения **x** на **Q**. Если **x < L**, то остаток и частное меняются местами так, чтобы в переменной **L** оказалось меньшее из двух значений, а в **M** – большее.

- 4.. таким образом, условию задачи при $Q=15$ соответствуют два числа – $15 \cdot 3 + 6 = 51$ и $15 \cdot 6 + 3 = 93$; наибольшее из них – 93.
5. Ответ: **93**.

Ещё пример задания:

P-08. Получив на вход число x , этот алгоритм печатает число M . Известно, что $x > 100$. Укажите наименьшее такое (т.е. большее 100) число x , при вводе которого алгоритм печатает 26.

```

x = int(input())
L = x
M = 65
if L % 2 == 0: M = 52
while L != M:
    if L > M:
        L -= M
    else:
        M -= L
print(M)

```

Решение:

- 1) видим, что в последней строке выводится на экран переменная **M**
- 2) **ключевой момент решения:** нужно узнать в строках программы, отмеченных знаком * в комментариях, **АЛГОРИТМ ЕВКЛИДА** для вычисления наибольшего общего делителя (НОД) чисел, записанный в переменные **M** и **L**
- 3) введённое значение **x** записывается в переменную **L** и участвует в поиске НОД
- 4) в переменную **M** до начала цикла записывается 65, но если было введено чётное ($L \% 2 = 0$) значение **x** (оно же **L**), значение **M** заменяется на 52
- 5) сначала предположим, что замены не было, и в **M** осталось значение 65; поскольку по условию алгоритм печатает 26, тогда получается, что **НОД(x, 65) = 26**; этого явно не может быть, потому что 65 не делится на 26
- 6) делаем вывод, что введено чётное значение **x** и произошла замена **M** на 52
- 7) итак, нужно найти чётное число **x**, большее 100, такое, что **НОД(x, 52) = 26**
- 8) первое число, большее 100, которое делится на 26 – это 104, но оно не подходит, потому что делится ещё и на 52, так что **НОД(x, 52) = 52**
- 9) поэтому берём следующее число, которое делится на 26: $104 + 26 = 130$
- 10) Ответ: **130**.

Ещё пример задания:

P-07. Ниже записан алгоритм. Укажите минимальное число x , при вводе которого алгоритм печатает 26391.

```

x = int(input())
K = 1

```

```

A = 0
B = 0
while x>0 :
    if (x % 10) % 2 == 0 :
        A = A*10+x % 10
    else:
        K = K*10
        B = B*10 + x % 10
    x = x // 10
A = A*K + B
print(A)

```

Решение:

- 18) видим, что в последней строке выводится на экран переменная **A**, которая вычисляется в предыдущей строке по формуле $A = A*K+B$
- 19) определим, сколько раз выполняется цикл **while** условие его продолжения – $x > 0$, с переменной **x** выполняется единственная операция – деление на 10 нацело:
- ```

while x>0 :
 ...
 x = x // 10

```
- отсюда делаем вывод, что цикл выполняется столько раз, сколько цифр в десятичной записи введённого числа **x**
- 20) теперь посмотрим, что происходит внутри цикла: выбор варианта действия зависит от выполнения условия
- $$(x \% 10) \% 2 = 0$$
- здесь  $x \% 10$  – это последняя цифра  $x$ , в этом условии проверяется её чётность (делимость на 2)
- 21) итак, если последняя цифра числа чётная, выполняется оператор
- $$A = A*10+x \% 10$$
- то есть, предыдущее значение **A** умножается на 10 и к результату добавляется последняя цифра **x** таким образом переменная **A** составляется из чётных цифр числа **x**, причём в обратном порядке, потому что новая цифра добавляется в конец числа, а предыдущие (которые были ближе к концу в записи числа **x**) продвигаются влево, в старшие разряды
- 22) теперь смотрим, как строится **B**: здесь всё то же самое, только нечётные цифры собираются в обратном порядке например, если исходное число было 12345, после окончания цикла мы получим **A=42** и **B=531**
- 23) но есть ещё переменная **K**, её начальное значение – 1, и с каждой найденной нечётной цифрой она умножается на 10, то есть  $K=10$  в степени, равной количеству нечётных цифр! для числа 12345 получим  $K=1000$
- 24) в предпоследней строке по формуле  $A = A*K+B$  собирается итоговое значение **A** для нашего примера (12345) мы получим  $A = 42*1000+531=42531$ , то есть **K** служит для того, чтобы сдвинуть комбинацию чётных цифр в начало числа
- 25) итак, нам задано число 26391, поэтому в искомом числе есть чётные цифры (по порядку, слева направо) {6, 2} и нечётные цифры {1, 9, 3} (тоже по порядку)
- 26) как же расположить эти цифры, чтобы получилось минимальное число? для этого сравниваем первые числа в списках чётных и нечётных чисел, и записываем в ответ меньшее из них эту операцию повторяем, пока числа в обоих списках не кончатся помним, что менять порядок чётных и нечётных чисел нельзя!
- 27) в данном случае получается {1, 6, 2, 9, 3} = 16293.

28) Ответ: 16293.

### Ещё пример задания:

**P-06.** Ниже записан алгоритм. Укажите наименьшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 4, а потом 2.

```

a = 0
b = 0
x = int(input())
while x > 0 :
 y = x % 10
 if y > 3 :
 a = a + 1
 if y < 8 :
 b = b + 1
 x = x // 10
print(a)
print(b)

```

#### Решение:

- 1) видим, что в последней строке выводятся на экран переменные **a** и **b**, поэтому сначала нужно определить, что они обозначают в программе
- 2) перед началом цикла переменные **a** и **b** обнуляются
- 3) на каждом шаге цикла при выполнении некоторых условий переменные **a** и **b** увеличиваются на 1, то есть представляют собой счётчики
- 4) увеличение переменных зависит от значения  $y = x \% 10$ , то есть от последней цифры числа
- 5) если последняя цифра числа больше 3, увеличивается счётчик **a**, если меньше 8 – счётчик **b**
- 6) в конце каждого шага цикла операция  $x = x // 10$  отсекает последнюю цифру в десятичной записи числа
- 7) цикл заканчивается, когда перестаёт выполняться условие  $x > 0$ , то есть, когда все цифры исходного числа отброшены
- 8) таким образом, делаем вывод: после завершения цикла в переменной **a** находится количество цифр, больших 3, в десятичной записи числа, а в переменной **b** – количество цифр, меньших 8
- 9) если было выведено 4 и 2, то в числе 4 цифры больше 3 и 2 цифры меньше 8
- 10) так как число пятизначное, есть  $4 + 2 = 5$  = одна цифра, которая больше 3 и меньше 8 одновременно она должна быть минимальной, поэтому эта цифра **4**
- 11) для того чтобы число было минимальным, ещё одна цифра должна быть минимальной и меньшей 3 – это старшая **1**, и три цифры минимальные из цифр, больших или равных 8, то есть **три цифры 8**
- 12) ответ: 14888.

### Ещё пример задания:

**P-05.** Ниже записан алгоритм. Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 12?

```

x = int(input())
a = 0
b = 0
while x > 0 :

```

```

a = a + 1
b = b + (x % 10)
x = x // 10
print(a)
print(b)

```

**Решение:**

- 1) видим, что в последней строке выводятся на экран переменные **a** и **b**, поэтому сначала нужно определить, что они обозначают в программе
- 2) перед началом цикла переменные **a** и **b** обнуляются
- 3) на каждом шаге цикла при выполнении некоторого условия переменная **a** увеличивается на 1, а **b** увеличивается на  $x \% 10$ , то есть, на остаток от деления **x** на 10 – это последняя цифра десятичной записи числа **x**
- 4) в конце каждого шага цикла операция  $x = x // 10$  отсекает последнюю цифру в десятичной записи числа
- 5) цикл заканчивается, когда перестаёт выполняться условие  $x > 0$ , то есть, когда все цифры исходного числа отброшены
- 6) таким образом, делаем вывод: после завершения цикла в переменной **a** находится количество цифр в десятичной записи числа, а в переменной **b** – их сумма
- 7) если было выведено 2 и 12, то в числе 2 цифры, и их сумма равна 12 таким образом, нам нужно найти все двузначные числа, в котором сумма значений цифр равна 12
- 8) число 12 может быть разложено на два слагаемых, меньших 10, как  $12 = 3 + 9 = 4 + 8 = 5 + 7 = 6 + 6 = 7 + 5 = 8 + 4 = 9 + 3$ , нам подходят числа 39, 48, 57, 66, 75, 84 и 93
- 9) всего таких чисел - 7
- 10) ответ: **7**.

**Ещё пример задания:**

**P-04.** Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 15.

```

x = int(input())
a = 0
b = 1
while x > 0 :
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print(a)
print(b)

```

**Решение:**

- 1) видим, что в последней строке выводятся на экран переменные **a** и **b**, поэтому сначала нужно определить, что они обозначают в программе
- 2) перед началом цикла переменная **a** обнуляется, а переменная **b** равна 1
- 3) на каждом шаге цикла при выполнении некоторого условия переменная **a** увеличивается на 1, а **b** умножается на  $x \% 10$ , то есть, на остаток от деления **x** на 10 – это последняя цифра десятичной записи числа **x**
- 4) в конце каждого шага цикла операция  $x = x // 10$  отсекает последнюю цифру в десятичной записи числа

- 5) цикл заканчивается, когда перестаёт выполняться условие  $x > 0$ , то есть, когда все цифры исходного числа отброшены
- 6) таким образом, делаем вывод: после завершения цикла в переменной **a** находится количество цифр в десятичной записи числа, а в переменной **b** – их произведение
- 7) если было выведено 2 и 15, то в числа 2 цифры, и их произведение равно 15 таким образом, нам нужно найти минимальное двузначное число, в котором произведение значений цифр равно 15
- 8) поскольку число 15 может быть разложено на два сомножителя, меньших 10, только как 3·5, минимальное подходящее число – 35.
- 9) ответ: **35**.

### Ещё пример задания:

**P-03.** Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 2.

```
x = int(input())
a = 0
b = 0
while x > 0 :
 c = x % 2
 if c == 0 :
 a = a + 1
 else:
 b = b + 1
 x = x // 10
print(a)
print(b)
```

#### Решение:

- 1) видим, что в последних строках выводятся на экран переменные **a** и **b**, поэтому сначала нужно определить, что они обозначают в программе
- 2) перед началом цикла обе переменные обнуляются
- 3) на каждом шаге цикла при выполнении некоторого условия переменная **a** увеличивается на 1, а если это условие не выполняется, то на 1 увеличивается **b** таким образом, обе переменных – счётчики
- 4) теперь посмотрим на условие  $c = 0$ : в предыдущей строке в переменную **c** записывается остаток от деления числа **x** на 2, то есть, переменная **c** определяет четность числа или, что равносильно, **чётность его последней цифры**
- 5) если последняя цифра чётная, то увеличивается счётчик **a**, а если нечётная – увеличивается счётчик **b**
- 6) в конце каждого шага цикла операция  $x = x // 10$  отсекает последнюю цифру в десятичной записи числа
- 7) таким образом, делаем вывод: после завершения цикла в переменной **a** находится количество чётных цифр в десятичной записи числа, а в переменной **b** – количество нечётных цифр
- 8) если было выведено 3 и 2, то в числа 5 цифр, из них 3 чётных и 2 нечётных таким образом, нам нужно найти минимальное пятизначное число, в котором 3 чётные и 2 нечётные цифры
- 9) минимальная чётная цифра – это 0, минимальная нечётная – 1 0 не может стоять на первом месте, поэтому число начинается с 1
- 10) для получения минимального числа после 1 должны идти нули и последняя цифра – снова 1

11) ответ: 10001

### Ещё пример задания:

**P-02.** Ниже записан алгоритм. После выполнения алгоритма было напечатано 3 числа. Первые два напечатанных числа – это числа 9 и 81. Какое наибольшее число может быть напечатано третьим?

```
x, y = map(int, input().split())
if y > x :
 z = x
 x = y
 y = z
a = x
b = y
while b > 0 :
 r = a % b
 a = b
 b = r
print(a)
print(x)
print(y)
```

#### Решение:

- 1) сложность этой задачи состоит в том, чтобы разобраться в алгоритме
- 2) сначала вводятся два числа и переставляются так, чтобы в переменной **x** было наибольшее число, а в переменной **y** – наименьшее из двух:

```
if y > x :
 z = x
 x = y
 y = z
```

- 3) затем исходные значения копируются в переменные **a** и **b** и с ними выполняется следующий алгоритм

```
while b > 0 :
 r = a % b
 a = b
 b = r
```

его суть сводится к тому, что меньшее из двух чисел, **a** и **b**, каждый раз заменяется на остаток от деления большего на меньшее до тех пор, пока этот остаток не станет равен нулю

- 4) делаем вывод, что это классический [Алгоритм Евклида](#), который служит для вычисления наибольшего общего делителя (НОД) двух чисел это делитель в результате оказывается в переменной **a**
- 5) смотрим, что выводится на экран: сначала значение переменной **a** (наибольший общий делитель исходных чисел, НОД(x,y)), затем значение **x** (большее из исходных чисел) и значение **y** (меньшее из исходных чисел)
- 6) по условию первое число – 9, второе – 81, поэтому третье число должно быть меньше, чем 81, и  $\text{НОД}(81, y) = 9$
- 7) наибольшее число, которое меньше 81 и делится на 9, равно 72 (обратите внимание, что исходные числа не могут быть равны, потому что в этом случае их НОД был бы равен 81)
- 8) ответ: 72

**Ещё пример задания:**

**P-01.** Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0
M = 0
while x > 0 :
 L = L+1
 if M < (x % 10) :
 M = x % 10
 x = x // 10
print(L)
print(M)
```

**Решение:**

- 1) для решения задачи необходимо понять, что делает эта программа
- 2) если это не видно сразу, можно выполнить ручную прокрутку для какого-то простого числа, например, для числа 251:

| оператор                             | условие            | x   | L | M |
|--------------------------------------|--------------------|-----|---|---|
| <code>x = int(input())</code>        |                    | 251 | ? | ? |
| <code>L = 0 M = 0</code>             |                    |     | 0 | 0 |
| <code>while x &gt; 0 :...</code>     | 251 > 0? да        |     |   |   |
| <code>L = L+1</code>                 |                    |     | 1 |   |
| <code>if M &lt; (x % 10) :...</code> | M < (251 % 10)? да |     |   |   |
| <code>M = x % 10</code>              |                    |     |   | 1 |
| <code>x = x // 10</code>             |                    | 25  |   |   |
| <code>while x &gt; 0 :...</code>     | 25 > 0? да         |     |   |   |
| <code>L = L+1</code>                 |                    |     | 2 |   |
| <code>if M &lt; (x % 10) :...</code> | M < (25 % 10)? да  |     |   |   |
| <code>M = x % 10</code>              |                    |     |   | 5 |
| <code>x = x // 10</code>             |                    | 2   |   |   |
| <code>while x &gt; 0 :...</code>     | 2 > 0? да          |     |   |   |
| <code>L = L+1</code>                 |                    |     | 3 |   |
| <code>if M &lt; (x % 10) :...</code> | M < (2 % 10)? нет  |     |   |   |
| <code>x = x // 10</code>             |                    | 0   |   |   |
| <code>while x &gt; 0 :...</code>     | 0 > 0? нет         |     |   |   |
| <code>print(L) print(M)</code>       |                    |     | 3 | 5 |

- 3) можно догадаться, что в результате работы программы в переменной L окажется число цифр числа, а в переменной M – наибольшая цифра, но это предположение нужно постараться доказать
- 4) нужно вспомнить (и запомнить), что для целого числа  $x$  остаток от деления на 10 ( $x \% 10$ ) – это последняя цифра в десятичной записи числа, а целочисленное деление ( $x // 10$ ) отсекает последнюю цифру, то есть из 123 получается 12
- 5) рассмотрим цикл, число шагов которого зависит от изменения переменной  $x$ :

```
while x > 0 :
 ...
 x = x // 10 { отсечение последней цифры }
```

здесь оставлены только те операторы, которые влияют на значение  $x$

6) из приведенного цикла видно, что на каждом шаге от десятичной записи **x** отсекается последняя цифра до тех пор, пока все цифры не будут отсечены, то есть **x** не станет равно 0 поэтому **цикл выполняется столько раз, сколько цифр в десятичной записи введенного числа**

7) на каждом шаге цикла переменная **L** увеличивается на 1:

```
L = L+1
```

других операторов, меняющих значение **L**, в программе нет поэтому после завершения цикла **в переменной L действительно находится количество цифр**

8) теперь разберемся с переменной **M**, которая сначала равна 0 оператор, в котором она меняется, выглядит так:

```
if M < (x % 10) :
 M = x % 10
```

учитывая, что **x % 10** – это последняя цифра десятичной записи числа, получается что если эта цифра больше, чем значение **M**, она записывается в переменную **M**

9) этот оператор выполняется в цикле, причем выражение **x % 10** по очереди принимает значения всех цифр исходного числа поэтому после завершения цикла **в переменной M окажется наибольшая из всех цифр**, то есть наша догадка подтверждается

10) итак, по условию задачи фактически требуется найти наибольшее трехзначное число, в котором наибольшая цифра – 7 очевидно, что это 777.

11) ответ: **777**.

#### Возможные ловушки и проблемы:

- это очень неплохая задача на понимание, тут достаточно сложно «вызубрить» метод решения, можно только освоить последовательность (системность) анализа
- ручной прокрутки в такой задаче недостаточно, по её результатам можно угадать алгоритм, но можно и не угадать в критическом случае можно сделать ручную прокрутку для нескольких чисел им попытаться понять закономерность

### Ещё пример задания:

**P-00.** Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 120.

```
x = int(input())
L = 0
M = 1
while x > 0 :
 L = L+1
 M = M*(x % 8)
 x = x // 8
print(L)
print(M)
```

#### Решение:

- 1) для решения задачи необходимо понять, что делает эта программа повторяя рассуждения из предыдущего примера, выясняем, что
  - a) переменная **L** с каждым шагом цикла увеличивается на 1
  - b) переменная **x** на каждом шаге цикла делится на 8 и остаток отбрасывается

поэтому можно сделать вывод, что в конце цикла переменная **L** будет равна **количеству цифр** введенного числа, записанного в *восьмеричной* системе счисления таким образом, восьмеричная запись числа содержит ровно 3 цифры

- 2) выражение  $x \% 8$  – это последняя цифра восьмеричной записи числа на каждом шаге цикла переменная **M** умножается на эту величину, поэтому в результате в **M** будет записано **произведение всех цифр** восьмеричной записи введенного числа
- 3) по условию это произведение равно 120, то есть  $120 = a \cdot b \cdot c$ , где  $a, b$  и  $c$  – числа от 0 до 7 (которые в восьмеричной системе счисления записываются одной цифрой)
- 4) поскольку нам нужно наибольшее число, перебираем делители числа 120, начиная со старшей цифры – 7 видим, что 120 на 7 не делится, поэтому такой цифры в восьмеричной записи числа нет
- 5) но 120 делится на 6, поэтому старшей цифрой может быть 6 – только в том случае, когда второй сомножитель можно представить в виде произведения двух чисел в интервале 1..6
- 6) делим 120 на 6, получаем 20 это число представляется как произведение 5 и 4, каждое из этих чисел записывается в виде одной восьмеричной цифры, то есть, они нам подходят
- 7) вспомним, что нас интересует максимальное число, поэтому цифры нужно выстроить в порядке убывания:  $654_8$
- 8) заметим, что мы получили число в восьмеричной системе, а ответ нужно дать в десятичной переводим:  $654_8 = 6 \cdot 8^2 + 5 \cdot 8^1 + 4 \cdot 8^0 = 428$ .
- 9) ответ: **428**.

**Возможные ловушки и проблемы:**

- поскольку в цикле идет деление на 8, мы получаем цифры числа в восьмеричной системе каждая из них должна быть в интервале 0..7 (не может быть 8 и 9)
- на последнем шаге нужно не забыть перевести число из восьмеричной системы в десятичную

**Задачи для тренировки<sup>1</sup>:**

- 1) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))
```

- 2) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 0:
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))
```

- 3) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 0.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 0:
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))
```

- 4) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 1:
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))
```

<sup>1</sup> Источники заданий:

1. Авторские разработки.
2. Тренировочные и диагностические работы МИОО.
3. Путимцева Ю.С. Информатика. Диагностические работы в формате ЕГЭ 2012. М.: МЦНМО, 2012.
4. Евич Л.Н., Кулабухов С.Ю. Информатика и ИКТ. Подготовка к ЕГЭ-2014. — Ростов-на-Дону: Легион, 2013.

- 5) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 0:
 M = M + (x % 10) // 2
 x = x // 10
print("%d\n%d" % (L, M))
```

- 6) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 1:
 M = M + (x % 10) // 2
 x = x // 10
print("%d\n%d" % (L, M))
```

- 7) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if M < x:
 M = x % 10
 x = x // 10
print("%d\n%d" % (L, M))
```

- 8) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if (M < x) and (x % 2 == 0):
 M = x % 10
 x = x // 10
print("%d\n%d" % (L, M))
```

- 9) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 10.

```
x = int(input())
L = 0; M = 0
while x > 0:
```

```

L = L + 1
if (M < x) and (x % 2 == 1):
 M = (x % 10) * 2
 x = x // 10
print("%d\n%d" % (L, M))

```

- 10) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 10.

```

x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if (M < x):
 M = (x % 10) * 2
 x = x // 10
print("%d\n%d" % (L, M))

```

- 11) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 72.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 12) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 14.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 13) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 14) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 0.

```

x = int(input())

```

```

a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 15) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))

```

- 16) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```

x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 0:
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))

```

- 17) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 0.

```

x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 0:
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))

```

- 18) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```

x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 1:
 M = M + (x % 10)
 x = x // 10
print("%d\n%d" % (L, M))

```

- 19) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 0:
 M = M + (x % 10) // 2
 x = x // 10
print("%d\n%d" % (L, M))
```

- 20) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if x % 2 == 1:
 M = M + (x % 10) // 2
 x = x // 10
print("%d\n%d" % (L, M))
```

- 21) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if M < x:
 M = x % 10
 x = x // 10
print("%d\n%d" % (L, M))
```

- 22) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 8.

```
x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if (M < x) and (x % 2 == 0):
 M = x % 10
 x = x // 10
print("%d\n%d" % (L, M))
```

- 23) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 10.

```
x = int(input())
L = 0; M = 0
while x > 0:
```

```

L = L + 1
if (M < x) and (x % 2 == 1):
 M = (x % 10) * 2
 x = x // 10
print("%d\n%d" % (L, M))

```

- 24) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 28.

```

x = int(input())
L = 0; M = 0
while x > 0:
 L = L + 1
 if (M < x):
 M = M + (x % 10) * 2
 x = x // 10
print("%d\n%d" % (L, M))

```

- 25) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 72.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 26) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 14.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 27) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 28) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 36.

```

x = int(input())

```

```

a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 29) (Д.Ю. Мельникова, г. Саратов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 24.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 8)
 x = x // 8
print("%d\n%d" % (a, b))

```

- 30) (Д.Ю. Мельникова, г. Саратов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 10.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 8)
 x = x // 8
print("%d\n%d" % (a, b))

```

- 31) (Д.Ю. Мельникова, г. Саратов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 6.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 6)
 x = x // 6
print("%d\n%d" % (a, b))

```

- 32) (Д.Ю. Мельникова, г. Саратов) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 9.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 5)
 x = x // 5
print("%d\n%d" % (a, b))

```

- 33) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 21.

```

x = int(input())

```

```

a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 34) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 35.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 35) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 4.

```

x = int(input())
L = 0; M = 9
while x > 5:
 L = L + 1
 if M > (x % 10):
 M = x % 10
 x = x // 10
print("%d\n%d" % (L, M))

```

- 36) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

x = int(input())
L = 0; M = 0
while x > 5:
 L = L + 1
 if M < (x % 10):
 M = x % 10
 x = x // 10
print("%d\n%d" % (L, M))

```

- 37) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 6, а потом 5.

```

x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 2
 b = b + (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 38) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 8, а потом 19.

```
x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 2
 b = b + (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))
```

- 39) Ниже записан алгоритм. После выполнения алгоритма было напечатано 3 числа. Первые два напечатанных числа – это числа 7 и 42. Какое наибольшее число может быть напечатано третьим?

```
x = int(input())
y = int(input())
if (y > x):
 z = x
 x = y
 y = z
a = x; b = y
while b > 0:
 r = a % b
 a = b
 b = r
print("%d\n%d\n%d" % (a, x, y))
```

- 40) Ниже записан алгоритм. После выполнения алгоритма было напечатано 3 числа. Первые два напечатанных числа – это числа 13 и 65. Какое наибольшее число может быть напечатано третьим?

```
x = int(input())
y = int(input())
if (y > x):
 z = x
 x = y
 y = z
a = x; b = y
while b > 0:
 r = a % b
 a = b
 b = r
print("%d\n%d\n%d" % (a, x, y))
```

- 41) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $N$ , при вводе которых алгоритм напечатает 17.

```
N = int(input())
for i in range(1, N):
 if N % i == 0:
 q = i
print(q)
```

- 42) (<http://ege.yandex.ru>) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите, сколько есть таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 12.

```
x = int(input())
```

```

a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))

```

- 43) (<http://ege.yandex.ru>) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 8.

```

x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 1
 b = b + (x % 100)
 x = x // 100
print("%d\n%d" % (a, b))

```

- 44) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 8.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 100)
 x = x // 100
print("%d\n%d" % (a, b))

```

- 45) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наименьшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 14, а потом 8.

```

x = int(input())
a = 0; b = 10
while x > 0:
 c = x % 10
 a = a + c
 if c < b:
 b = c
 x = x // 10
print("%d\n%d" % (a, b))

```

- 46) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ . Укажите наибольшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 15, а потом 5.

```

x = int(input())
a = 0; b = 10
while x > 0:
 c = x % 10
 a = a + c
 if c < b:
 b = c
 x = x // 10
print("%d\n%d" % (a, b))

```

- 47) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ .  
Укажите наибольшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 13, а потом 3.

```
x = int(input())
a = 0; b = 10
while x > 0:
 c = x % 10
 a = a + c
 if c < b:
 b = c
 x = x // 10
print("%d\n%d" % (a, b))
```

- 48) Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $a$  и  $b$ .  
Укажите наименьшее из чисел  $x$ , при вводе которых алгоритм печатает сначала 19, а потом 4.

```
x = int(input())
a = 0; b = 10
while x > 0:
 c = x % 10
 a = a + c
 if c < b:
 b = c
 x = x // 10
print("%d\n%d" % (a, b))
```

- 49) Ниже записан алгоритм. Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 2.

```
x = int(input())
a = 0; b = 0
while x > 0:
 c = x % 2
 if c == 0:
 a = a + 1
 else:
 b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))
```

- 50) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 0.

```
x = int(input())
a = 0; b = 0
while x > 0:
 c = x % 2
 if c == 0:
 a = a + 1
 else:
 b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))
```

- 51) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 2.

```
x = int(input())
a = 0; b = 0
```

```

while x > 0:
 c = x % 2
 if c == 0:
 a = a + 1
 else:
 b = b + 1
 x = x // 8
print("%d\n%d" % (a, b))

```

- 52) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 0.

```

x = int(input())
a = 0; b = 0
while x > 0:
 c = x % 2
 if c == 0:
 a = a + 1
 else:
 b = b + 1
 x = x // 6
print("%d\n%d" % (a, b))

```

- 53) Получив на вход число  $x$ , этот алгоритм печатает два числа  $K$  и  $R$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 3.

```

x = int(input())
K = 0; R = 9
y = x % 10
while x > 0:
 K = K + 1
 if R > (x % 10):
 R = x % 10
 x = x // 10
R = y - R
print("%d\n%d" % (K, R))

```

- 54) Получив на вход число  $x$ , этот алгоритм печатает два числа  $K$  и  $R$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```

x = int(input())
K = 0; R = 9
y = x % 10
while x > 0:
 K = K + 1
 if R > (x % 10):
 R = x % 10
 x = x // 10
R = y - R
print("%d\n%d" % (K, R))

```

- 55) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 13.

```

x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 1
 b = b + (x % 100)

```

```
x = x // 100
print("%d\n%d" % (a, b))
```

- 56) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 15.

```
x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 1
 b = b + (x % 100)
 x = x // 100
print("%d\n%d" % (a, b))
```

- 57) Ниже записан алгоритм. Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 15?

```
x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 1
 b = b + (x % 10)
 x = x // 10
print("%d\n%d" % (a, b))
```

- 58) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает числа:  $a$  и  $b$ . Укажите наименьшее положительное пятизначное число  $x$ , при котором после выполнения алгоритма будет напечатано сначала 5, а потом 2.

```
x = int(input())
a = 0; b = 10
while x > 0:
 y = x % 10
 x = x // 10
 if (y > a):
 a = y
 if (y < b):
 b = y
print("%d\n%d" % (a, b))
```

- 59) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает числа:  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которого после выполнения алгоритма будет напечатано сначала 2, а потом 22.

```
x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 1
 b = b + (x % 100)
 x = x // 100
print("%d\n%d" % (a, b))
```

- 60) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 5.

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 100)
```

```

x = x // 100
print("%d\n%d" % (a, b))

```

- 61) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 18.

```

x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 100)
 x = x // 100
print("%d\n%d" % (a, b))

```

- 62) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наименьшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 5, а потом 3.

```

x = int(input())
a = 0; b = 0
while x > 0:
 y = x % 10
 if y > 3: a = a + 1
 if y < 8: b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))

```

- 63) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 4, а потом 3.

```

x = int(input())
a = 0; b = 0
while x > 0:
 y = x % 10
 if y > 3: a = a + 1
 if y < 8: b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))

```

- 64) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 5, а потом 4.

```

x = int(input())
a = 0; b = 0
while x > 0:
 y = x % 10
 if y > 4: a = a + 1
 if y < 6: b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))

```

- 65) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наименьшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 3, а потом 4.

```

x = int(input())
a = 0; b = 0
while x > 0:
 y = x % 10
 if y > 4: a = a + 1
 if y < 6: b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))

```

- 66) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наименьшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 2, а потом 4.

```
x = int(input())
a = 0; b = 0
while x > 0:
 y = x % 10
 if y > 3: a = a + 1
 if y < 7: b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))
```

- 67) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 2, а потом 4.

```
x = int(input())
a = 0; b = 0
while x > 0:
 y = x % 10
 if y > 4: a = a + 1
 if y < 7: b = b + 1
 x = x // 10
print("%d\n%d" % (a, b))
```

- 68) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 6.

```
x = int(input())
L = 0; M = 0
while x > 0:
 M = M + 1
 if x % 2 != 0:
 L = L + 1
 x = x // 2
print("%d\n%d" % (L, M))
```

- 69) Ниже записан алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 11.

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b * (x % 1000)
 x = x // 1000
print("%d\n%d" % (a, b))
```

- 70) Ниже записан алгоритм. Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 13.

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 2
 b = b * (x % 1000)
 x = x // 1000
print("%d\n%d" % (a, b))
```

- 71) Ниже записан алгоритм. Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 12?

```
x = int(input())
a = 0; b = 0
while x > 0:
 a = a + 1
 b = b + (x % 100)
 x = x // 100
print("%d\n%d" % (a, b))
```

- 72) Получив на вход число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите наибольшее число  $x$ , при вводе которого алгоритм печатает сначала 5, а потом 8.

```
x = int(input())
L = 0
M = 0
while x > 0:
 M = M + 1
 if x % 2 != 0:
 L = L + 1
 x = x // 2
print(L)
print(M)
```

- 73) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 15.

```
x = int(input())
L = x - 30
M = x + 30
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 74) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 30.

```
x = int(input())
L = x - 30
M = x + 30
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 75) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 18.

```
x = int(input())
a = 0
b = 0
while x > 0:
 a = a + 1
 if x % 2 == 0:
```

```

 b = b + x % 10
 x = x // 10
print(a)
print(b)

```

- 76) Получив на вход число **x**, этот алгоритм печатает два числа **a** и **b**. Укажите наименьшее из таких чисел **x**, при вводе которых алгоритм печатает сначала 45, а потом 5.

```

x = int(input())
a = 1
b = 10
while x > 0:
 c = x % 10
 a = a*c
 if c < b:
 b = c
 x = x // 10
print(a)
print(b)

```

- 77) Получив на вход число **x**, этот алгоритм печатает число **M**. Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число **x**, при вводе которого алгоритм печатает 2.

```

x = int(input())
L = x - 12
M = x + 12
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)

```

- 78) Получив на вход число **x**, этот алгоритм печатает число **M**. Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число **x**, при вводе которого алгоритм печатает 11.

```

x = int(input())
L = x - 21
M = x + 12
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)

```

- 79) Получив на вход число **x**, этот алгоритм печатает число **M**. Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число **x**, при вводе которого алгоритм печатает 35.

```

x = int(input())
L = x - 15
M = x + 20
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L

```

```
print(M)
```

- 80) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 9.

```
x = int(input())
L = x - 18
M = x + 36
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 81) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 35.

```
x = int(input())
L = x - 20
M = x + 15
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 82) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 4.

```
x = int(input())
L = x - 16
M = x + 32
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 83) Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т. е. большее 100) число  $x$ , при вводе которого алгоритм печатает 16.

```
x = int(input())
L = x - 16
M = x + 16
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)
```

- 84) (Д.Ф. Муфаззалов) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 2.

```
x = int(input())
```

```

a = 0
b = 0
i = 0
while x > 0:
 i = i + 1
 c = x % 10
 if i % 2 == 0:
 a = a + c
 else:
 b = b + c
 x = x // 10
print(a)
print(b)

```

- 85) (Д.Ф. Муфаззалов) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 5.

```

x = int(input())
b = 0
i = 0
while x > 0:
 if i % 2 > 0:
 b = b + x % 10
 x = x // 10
 i = i + 1
print(i)
print(b)

```

- 86) (Д.Ф. Муфаззалов) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 5.

```

x = int(input())
b = 0
i = 0
while x > 0:
 b = b + x % 10
 x = x // 10
 i = i + 1
print(i)
print(b)

```

- 87) (Д.Ф. Муфаззалов) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 2.

```

x = int(input())
a = 0
b = 0
i = 0
c = 0
while x > 0:
 i = i + 1
 if i % 2 == 0:
 a = a + c
 else:
 b = b + c
 c = x % 10

```

```

 x = x // 10
print(a)
print(b)

```

- 88) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 15.

```

x = int(input())
a = 0; b = 0; d = 0
while x > 0:
 if d % 2 == 0:
 a += x % 10
 else:
 b += x % 10
 x = x // 10
 d += 1
print(a, b)

```

- 89) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 14, а потом 12.

```

x = int(input())
a = 0; b = 0; d = 0
while x > 0:
 if d % 2 == 0:
 a += x % 10
 else:
 b += x % 10
 x = x // 10
 d += 1
print(a, b)

```

- 90) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , большее, чем 200, при вводе которого алгоритм напечатает 70.

```

x = int(input())
L = 2*x-30
M = 2*x+40
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)

```

- 91) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , большее, чем 200, при вводе которого алгоритм напечатает 50.

```

x = int(input())
L = 2*x-20
M = 2*x+30
while L != M:
 if L > M:
 L = L - M
 else:
 M = M - L
print(M)

```

- 92) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которого алгоритм напечатает четырёхзначное число.

```
x = int(input())
x0 = x
N = 0
while x>0:
 d = x % 2
 N = 10*N + d
 x = x // 2
N += x0
print(N)
```

- 93) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которого алгоритм напечатает пятизначное число.

```
x = int(input())
x0 = x
N = 0
while x>0:
 d = x % 2
 N = 10*N + d
 x = x // 2
N += x0
print(N)
```

- 94) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которого алгоритм напечатает четырёхзначное число.

```
x = int(input())
x0 = x
N = 0
while x>0:
 d = x % 3
 N = 10*N + d
 x = x // 3
N += x0
print(N)
```

- 95) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которого алгоритм напечатает пятизначное число.

```
x = int(input())
x0 = x
N = 0
while x>0:
 d = x % 3
 N = 10*N + d
 x = x // 3
N += x0
print(N)
```

- 96) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , при вводе которого алгоритм напечатает шестизначное число.

```
x = int(input())
x0 = x
N = 0
while x>0:
 d = x % 3
```

```

N = 10*N + d
x = x // 3
N += x0
print(N)

```

- 97) (А.Н. Носкин) Ниже приведён алгоритм. Укажите наименьшее число  $x$ , при вводе которого алгоритм напечатает число, большее 10200.

```

x = int(input())
x0 = x
N = 0
while x > 0:
 d = x % 5
 N = 10*N + d
 x = x // 5
N += x0
print(N)

```

- 98) (А.Н. Носкин) Ниже приведён алгоритм. Укажите наименьшее число  $x$ , при вводе которого алгоритм напечатает число, большее, чем 10300.

```

x = int(input())
x0 = x
N = 0
while x > 0:
 d = x % 4
 N = 10*N + d
 x = x // 4
N += x0
print(N)

```

- 99) Ниже приведён алгоритм. Укажите наибольшее число  $x$ , при вводе которого алгоритм напечатает сначала 3, потом – 5.

```

x = int(input())
Q = 6
L = 0
while x >= Q:
 L = L + 1
 x = x - Q
M = x
if M < L:
 M = L
 L = x
print(L)
print(M)

```

- 100) Ниже приведён алгоритм. Укажите наибольшее число  $x$ , при вводе которого алгоритм напечатает сначала 8, потом – 11.

```

x = int(input())
Q = 16
L = 0
while x >= Q:
 L = L + 1
 x = x - Q
M = x
if M < L:
 M = L

```

```
L = x
print(L)
print(M)
```

- 101) (Д.В. Богданов) Укажите наименьшее натуральное число  $x$ , при вводе которого будет два раза напечатано число 3.

```
x = int(input())
a = 0
b = 1
while x > 0:
 if x % 3 > 0:
 a += 1
 if x % 3 > 1:
 b += 1
 x = x // 10
print(a)
print(b)
```

- 102) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 24?

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b*(x%10)
 x = x//10
print(a, "\n", b)
```

- 103) Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 0?

```
x = int(input())
a = 0; b = 1
while x > 0:
 a = a + 1
 b = b*(x%10)
 x = x//10
print(a, "\n", b)
```

- 104) (Д.Ф. Муфаззалов) Укажите наибольшее двузначное натуральное число, при вводе которого приведенная ниже программа напечатает число 0.

```
n = int(input())
i = 0
while n > 0:
 i = i + n % 16
 n = n // 16
printf(i % 15)
```

- 105) (Д.Ф. Муфаззалов) Укажите наименьшее трехзначное натуральное число, при вводе которого приведенная ниже программа напечатает число 0.

```
n = int(input())
i = 0
while n > 0:
 i = i + n % 20
 n = n // 20
printf(i % 19)
```

- 106) (Д.Ф. Муфаззалов) Укажите количество двузначных натуральных чисел, при вводе которых приведенная ниже программа напечатает число 0.

```
n = int(input())
i = 0
while n > 0:
 i = i + n % 8
 n = n // 8
printf(i % 7)
```

- 107) (Д.Ф. Муфаззалов) Укажите количество двузначных натуральных чисел, при вводе которых приведенная ниже программа напечатает число, отличное от 0.

```
n = int(input())
i = 0
while n > 0:
 i = i + n % 9
 n = n // 9
printf(i % 8)
```

- 108) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 1, потом – 4.

```
x = int(input())
a=0; b=0
while x > 0:
 if x%2 == 0:
 a += 1
 else:
 b += x%6
 x = x//6
print(a, b)
```

- 109) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 2, потом – 6.

```
x = int(input())
a=0; b=0
while x > 0:
 if x%2 == 0:
 a += 1
 else:
 b += x%6
 x = x//6
print(a, b)
```

- 110) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 1, потом – 5.

```
x = int(input())
a=0; b=0
while x > 0:
 if x%2 == 0:
 a += 1
 else:
 b += x%4
 x = x//4
print(a, b)
```

- 111) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 2, потом – 7.

```
x = int(input())
a=0; b=0
while x > 0:
 if x%2 == 0:
 a += 1
 else:
 b += x%4
 x = x//4
print(a, b)
```

- 112) Укажите наибольшее натуральное число, при вводе которого эта программа напечатает сначала 3, потом – 6.

```
x = int(input())
L = 0
M = 0
while x > 0:
 L = L + 1
 if (x % 8) != 0:
 M = M + x % 8
 x = x // 8
print(L)
print(M)
```

- 113) Укажите наибольшее натуральное число, при вводе которого эта программа напечатает сначала 14, потом – 3.

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

- 114) Укажите наибольшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом – 8.

```
x = int(input())
a=0; b=1
while x > 0:
 if x%2 > 0:
 a += x%8
 else:
 b *= x%8
 x = x//8
print(a, b)
```

- 115) Укажите наименьшее натуральное число, при вводе которого эта программа напечатает сначала 2, потом – 24.

```
x = int(input())
a=0; b=1
while x > 0:
```

```

if x%2 > 0:
 a += x%8
else:
 b *= x%8
x = x//8
print(a, b)

```

- 116) Укажите наибольшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом – 7.

```

x = int(input())
a=0; b=1
while x > 0:
 if x%2 > 0:
 a += x%6
 else:
 b += x%6
 x = x//6
print(a, b)

```

- 117) Укажите наименьшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом – 9.

```

x = int(input())
a=0; b=1
while x > 0:
 if x%2 > 0:
 a += x%6
 else:
 b += x%6
 x = x//6
print(a, b)

```

- 118) Укажите наибольшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом – 6.

```

x = int(input())
a=0; b=1
while x > 0:
 if x%2 > 0:
 a += 1
 else:
 b += x%5
 x = x//5
print(a, b)

```

- 119) Укажите наименьшее трёхзначное натуральное число, при вводе которого эта программа напечатает сначала 2, потом – 9.

```

x = int(input())
a=0; b=1
while x > 0:
 if x%2 > 0:
 a += 1
 else:
 b += x%5
 x = x//5
print(a, b)

```

- 120) (Д.Ф. Муфаззалов, Уфа) Укажите наименьшее четырехзначное натуральное число, при вводе которого приведенная ниже программа напечатает число 4.

```
n = int(input())
a = -1
while n > 9 and a != n % 10:
 a = n % 10
 n //= 10
print(n % 10)
```

- 121) (Д.Ф. Муфаззалов, Уфа) Укажите минимальное натуральное число, при вводе которого приведенная ниже программа напечатает число 4, а потом число 7.

```
n = int(input())
a = -1
k = 0
while n > 9 and a != n % 10:
 a = n % 10
 n //= 10
 k += 1
print(k)
print(a)
```

- 122) (Д.Ф. Муфаззалов, Уфа) Укажите минимальное трехзначное натуральное число, при вводе которого приведенная ниже программа напечатает число 5.

```
n = int(input())
a = -1
while n > 7 and a != n % 8:
 a = n % 8
 n //= 8
if a == n % 8:
 print(a)
else:
 print(n)
```

- 123) (Д.Ф. Муфаззалов, Уфа) Укажите наибольшее трехзначное натуральное число, при вводе которого приведенная ниже программа напечатает число 5.

```
n = int(input())
a = -1
while n > 9 and a != n % 10:
 a = n % 10
 n //= 10
if a == n % 10:
 print(a)
else:
 print(n)
```

- 124) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 4, а потом 7.

```
x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 > 0:
 a = a + x % 13
 else:
```

```

 b = b * (x % 13)
 x = x // 13
print(a)
print(b)

```

- 125) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 5, а потом 2.

```

x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 == 0:
 a = a + x % 13
 else:
 b = b * (x % 13)
 x = x // 13
print(a)
print(b)

```

- 126) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 2, а потом 9.

```

x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 > 0:
 a = a + x % 11
 else:
 b = b * (x % 11)
 x = x // 11
print(a)
print(b)

```

- 127) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее из таких чисел  $x$ , при вводе которого алгоритм печатает сначала 1, а потом 8.

```

x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 > 0:
 a = a + x % 11
 else:
 b = b * (x % 11)
 x = x // 11
print(a)
print(b)

```

- 128) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наибольшее трёхзначное натуральное число, при вводе которого алгоритм печатает сначала 1, а потом 8.

```

x = int(input())
a = 0

```

```
b = 1
while x > 0:
 if x % 2 > 0:
 a = a + x % 11
 else:
 b = b * (x % 11)
 x = x // 11
print(a)
print(b)
```

- 129) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 1, а потом 9.

```
x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 == 0:
 a = a + x % 9
 else:
 b = b * (x % 9)
 x = x // 9
print(a)
print(b)
```

- 130) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 3, а потом тоже - 3.

```
x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 == 0:
 a = a + x % 7
 else:
 b = b * (x % 7)
 x = x // 7
print(a)
print(b)
```

- 131) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее натуральное число, при вводе которого алгоритм печатает сначала 3, а потом 12.

```
x = int(input())
a = 0
b = 1
while x > 0:
 if x % 2 > 0:
 a = a + x % 13
 else:
 b = b * (x % 13)
 x = x // 13
print(a)
print(b)
```

- 132) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наименьшее четырёхзначное натуральное число, при вводе которого алгоритм печатает сначала 4, а потом тоже - 4.

```
x = int(input())
a = 0
b = 0
while x > 0:
 if x % 2 > 0:
 a = a + x % 9
 else:
 b = b + x % 9
 x = x // 9
print(a)
print(b)
```

- 133) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наибольшее трёхзначное натуральное число, при вводе которого алгоритм печатает сначала 4, а потом тоже - 4.

```
x = int(input())
a = 0
b = 0
while x > 0:
 if x % 2 == 0:
 a = a + x % 5
 else:
 b = b + x % 5
 x = x // 5
print(a)
print(b)
```

- 134) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 0?

```
x = int(input())
a = 0
b = 1
while x > 0:
 a = a + 1
 b = b * (x % 10)
 x = x // 10
print(a)
print(b)
```

- 135) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 4, а потом 5?

```
x = int(input())
a = 0
while x > 0:
 a = a + 1
 b = x % 6
 x = x // 6
print(a)
print(b)
```

- 136) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 5, а потом 7?

```
x = int(input())
a = 0
while x > 0:
 a = a + 1
 b = x % 9
 x = x // 9
print(a)
print(b)
```

- 137) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 24?

```
x = int(input())
a = 0
b = 1
while x > 0:
 a = a + 1
 if x % 8 != 1:
 b = b * (x % 8)
 x = x // 8
print(a)
print(b)
```

- 138) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 24?

```
x = int(input())
a = 0
b = 1
while x > 0:
 a = a + 1
 if x % 8 != 0:
 b = b * (x % 8)
 x = x // 8
print(a)
print(b)
```

- 139) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 12?

```
x = int(input())
a = 0
b = 1
while x > 0:
 a = a + 1
 if x % 14 != 0:
 b = b * (x % 14)
 x = x // 14
print(a)
print(b)
```

- 140) (С.С. Поляков, Саратов) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 10?

```
x = int(input())
a = 0
b = 1
while x > 0:
 a = a + 1
 if x % 12 != 0:
 b = b * (x % 12)
 x = x // 12
print(a)
print(b)
```

- 141) (Д. Ф. Муфаззалов, Уфа) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число, сумма цифр которого равна 15.

```
x = int(input())
L = 0
M = 1
while x > 0:
 L = x % 10 * M + L
 x = x // 10
 M = M * 10
print(L)
```

- 142) (Д. Ф. Муфаззалов, Уфа) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число, сумма цифр которого равна 15.

```
x = int(input())
L = 0
M = 1
while x > 0:
 L = x % 8 * M + L
 x = x // 8
 M = M * 10
print(L)
```

- 143) (Д. Ф. Муфаззалов, Уфа) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число 110.

```
x = int(input())
L = 0
M = 1
while x > 0:
 L = x % 16 * M + L
 x = x // 16
 M = M * 10
print(L)
```

- 144) Укажите минимальное двузначное натуральное число, при вводе которого этот алгоритм напечатает число 8.

```
x = int(input())
a = 0; b = 10
while x > 0:
 d = x % 6
 if d > a: a = d
 if d < b: b = d
```

```
x = x // 6
print(a+b)
```

- 145) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число 11.

```
x = int(input())
a = 0; b = 10
while x > 0:
 d = x % 7
 if d > a: a = d
 if d < b: b = d
 x = x // 7
print(a+b)
```

- 146) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает сначала 7, а потом – 12.

```
x = int(input())
a = 0; b = 1
while x > 0:
 if x % 2 > 0:
 a += x % 12
 else:
 b *= x % 12
 x = x // 12
print(a)
print(b)
```

- 147) Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число 18.

```
x = int(input())
a = 0; b = 10
while x > 0:
 d = x % 9
 if d > a: a = d
 if d < b: b = d
 x = x // 9
print(a*b)
```

- 148) (Е. Дзюбс) Ниже приведён алгоритм. Укажите наименьшее из таких чисел  $x$ , большее, чем 100, при вводе которого алгоритм напечатает 21.

```
x = int(input())
L = 3*x - 6
M = 3*x + 99
while L != M:
 if L > M:
 L = L - M
 else: M = M - L
print(M)
```

- 149) (Е. Дзюбс) Ниже записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа  $a$  и  $b$ . Укажите наибольшее пятизначное число  $x$ , при вводе которого алгоритм печатает сначала 2, а потом 1.

```
x = int(input())
a, b = 0, 0
while x > 0:
 y = x % 10
```

```

if y > 3:
 a = a + 1
else:
 b = b - 1
if y < 8:
 b = b + 1
x = x // 10
print(a)
print(b)

```

- 150) (Е. Джобс) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Укажите наибольшее натуральное число, при вводе которого алгоритм печатает сначала 13, а потом 3.

```

x = int(input())
a = 0
b = 0
while x > 0:
 x = x // 9
 if x % 2 > 0:
 a = a + x % 9
 b = b + 1
print(a)
print(b)

```

- 151) (Досрочный ЕГЭ-2020) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите наибольшее натуральное число, при вводе которого алгоритм печатает сначала 2, а потом 3.

```

x = int(input())
L = 0
M = 0
while x > 0:
 M = M + 1
 if x % 2 != 0:
 L = L + x % 8
 x = x // 8
print(L)
print(M)

```

- 152) (Досрочный ЕГЭ-2020) Получив на вход натуральное число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите наибольшее натуральное число, при вводе которого алгоритм печатает сначала 12, а потом 3.

```

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

```