



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

№2

Язык: Python 3.5 с пояснениями

def main():

u = int(input())

v = int(input())

res = []

for x in range(u, v+1):

if x%10 in [0, 1, 4, 5, 6, 9]:

res.append(x)

print(res)

# цикл - перебор  $x \in [u; v]$ 

# сравнение посл. цифры с 6-ю

# добавление в конец *данными*

main()

Объяснение: Вводятся целые числа  $u$  и  $v$ , объявляется массив  $res$ .  
 П.п.д.:  $0^3, 1^3, 4^3, 5^3, 6^3, 9^3 \equiv 0, 1, 4, 5, 6, 9 \pmod{10}$ , то числа, оканчивающиеся на эти цифры - триморфные. Достаточно проверить остаток от деления на 10 (в 6-й строке просто упрощенная запись  $if\ x\%10 == 0\ or\ x\%10 == 1\ or\ \dots$ ), добавляя само число в массив.

№3

def main():

s = int(input())

if s%3 != 0:

b = "ошибка"

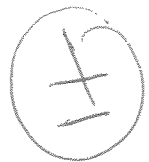
else:

b = s // 3

print(b)

# "!=" - не равно

# "//" - целочисленное деление



main()  $B-A=C-B \Rightarrow A+B+C=3B \Rightarrow B = \frac{s}{3}$

Пояснение: Проверка делимости на 3 и вывод рез-та.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

№4

```
def binpow(a, n):
    # реализация бинарного возведения в степень
    if n > 1:
        if n % 2 == 0:
            return binpow(a, n // 2) * binpow(a, n // 2) # рекурсивный вызов
        else:
            return binpow(a, n // 2) * binpow(a, n // 2) * a
    else:
        return a
```

```
def main():
    res = []
    s = str(binpow(2, 3078)) # м.к.  $64^{513} = 2^{3078}$ , str - переводим в строку
    l = len(s) # len - функция нахождения длины строки
    for i in range(l - 1, -1): # от l-го 0 с шагом -1
        if s[i] == "2":
            res.append(l - i) # м.к. разряды нумеруются с 1
    print(res)
```

main()

№5

```
def factorial(n):
    if n < 2:
        return 1
    return n * factorial(n - 1)

def is_factorion(a):
    t = 1
    for x in str(a):
        t *= factorial(int(x))
    return a == t

def main():
    u, v = map(int, input().split()) # блог u, v на одной строке
    res = []
    for x in range(u, v + 1):
        if is_factorion(x):
            res.append(x)
    print(res)
```

main()



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа



n/3!

```
def pow(a, n):
    i = 1
    while i < n:
        ax = a
        i += 1

def main():
    a, b = map(int, input().split())
    x, y = from_maya(a), from_maya(b)
    s = x + y
    s = into_maya(s)
    print(s)

main()
```



```
def from_maya(x):
    res = 0
    s = str(x)
    l = len(s)
    for k in range(l):
        if k == 2:
            res += 360 * int(s[l-k-1])
        else:
            res += pow(20, k) * int(s[l-k-1])
    return res
```

```
def into_maya(x):
    res = ""
    l = len(str(x))
    while x > 0:
        if l == 3:
            res += x % 18
            x //= 18
            l = len(str(x))
        else:
            res += x % 20
            x //= 20
            l = len(str(x))
    maya = ""
    for i in range(len(res)-1, -1, -1):
        maya += res[i]
    return int(maya)
```



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

- ~1: 1) Создается массив по максимуму коммиссии, разрядов в большем числе + 1. Если  $wa - bo$  разрядов одинаково, то берем число разрядов меньше. Обозначим число разрядов как  $n$ . Получим массив  $[0..n]$
- 2) Складываем число в последнем разряде. Если полученное число больше 19, то из него вычитаем 20. Результат записывается в ячейку  $n$ . В ячейку  $n-1$  записываем число 1. Если полученное число меньше 20, то результат записывается в ячейку  $n$ .
- 3) Складываем число в предпоследнем разряде. Если в ячейке  $n-1$  стоит 1, то она прибавляется к данным числом. Если полученное число больше 17, то из него вычитаем 18. Результат записывается в ячейку  $n-1$ , в ячейку  $n-2$  записываем число 1. Если полученное число меньше 18, то результат записывается в ячейку  $n-1$ .
- 4) Если  $n \geq 3$  то открываем цикл, итерациями  $n-2$  раз:  
 Прибавим к  $i$  значение  $n-2$  при первом проходе цикла,  $n-3$  при втором проходе и т.д. пока  $i$  не станет равна 1.  
 Складываем число в разряде  $i$ . Если ~~полученное~~  $wa$  в ячейке  $i$  стоит 1, то прибавим  $n$  к  $e$ . Если ~~число~~ полученное число больше 19, то отнимаем от него 20, результат записываем в ячейку  $i$ , в ячейку  $i-1$  записываем 1.  
 Если полученное число меньше 20, то записываем его в ячейку  $i$ . Цикл закрываем.
- 5) Если ячейка  $[0]$  пуста, то записываем результат: число в виде массива с ячейками от 1 до  $n$ . Выводим массив.  
 Если ячейка  $[0]$  не пуста, то записываем результат: число в виде массива с ячейками от 0 до  $n$ . Выводим результат.

нч-нет

④



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

№2: 1) Вводим число  $n$  и  $v$ .  
 2) Открываем цикл, повторяющийся от  $n$  до  $v$   
 раз: Выводим переменную  $i$  значение  $n$  при первом проходе,  $n+1$  при втором проходе и т.д., пока  $i$  не достигнет значения  $v$ .  
 Возведем число  $i$  в  $v$  ст.  $\oplus$   
 Выводим переменную  $i$  значение  $i^3$ .  
 Выводим переменную  $i$  строкой без пробелов, обозначим как  $s$ . Узнаем длину  $s$ .  
 Возведем остаток от деления числа  $i$  на 10 в степень  $s$ . Если данный остаток равен числу  $i$ , то число  $i$  - палиндром.  
 Выводим его как один из результатов.  
 Закрываем цикл.

№3: 1) Вводим число  $s$ .  
 2) П.п. число  $s$  является суммой трех чисел  $A, B$  и  $C$  различающихся на какое-то определенное число ( $A+k=B, B+k=C$ ), то их сумме является число  $B$ , умноженным на 3  $\Rightarrow$  Если число  $s$  не делится на 3, вывести «ошибка», в ином  $y$  (остаток от деления числа  $s$  на 3 не равен 0), вывести «ошибка», в ином случае, вывести результат деления. ~~Закрываем цикл.~~  $\oplus$

№5: 1) Вводим число  $n$  и  $v$ .  
 2) Открываем цикл, повторяющийся от  $n$  до  $v$   
 раз: Выводим переменную  $i$  значение  $n$  при первом проходе,  $n+1$  при втором и т.д., пока значение  $i$  не станет равно  $v$ .  
 Переводим число  $i$  в строковый вид, выводим это значение переменной  $s$ .  
 Выводим переменную  $p$  значение 1.  
 Открываем цикл от  $l=1$  до длины  $s$ :  
 Выводим переменную  $k$  значение  $s[l]$ .  
 В строковом виде.  
 Открываем цикл: перемножаем число от  $l$  до  $k$ . Выводим переменную  $p$  значение произведения  $p$  на данный элемент строки. Закрываем цикл;  
 Выводим цикл;



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа



Если значение  $r$  равно значению  $i$ , то  
число  $i$  — факто риди первого  $10^i$ .  
Выводим данное число  $i$ .  
Защита числа.





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

№1) Так как мы работаем с 20-чной системой счисления, то для удобства представим числа от 10 до 19 в виде букв от A до J, то есть, получим в нашей системе следующий порядок:

I разряд:	II разряд:	III разряд:	IV разряд:
0	10	100	1000
1	11	101	1001
2	...	...	...
3	НН	JHJ	...
4	...	...	JJIH
5	НI	...	JJHI
6	HJ	...	JJHJ
7	...	...	...
8	...	...	...
9	...	...	...
A	...	...	...
B	...	...	...
C	...	...	...
D	...	...	...
E	...	...	...
F	...	...	...
G	...	...	...
H	...	...	...
I	...	...	...
J	...	...	...

Можем заметить, что во втором разряде любого числа максимальное значение H ( $H=(18)$ ). Также, если допустить существование чисел I и J во втором разряде (то есть от IO до JJ - всего 40 чисел), то можно заметить, что данная система счисления не будет отличаться от десятичной  $\Rightarrow \Rightarrow$  (допустим, что мы можем использовать во втором разряде 20 цифровых, тогда сложение происходит попарное (как в десятичной) и без формирования).

Если мы в сумме при сложении получим во втором и первом разрядах число больше HJ (это по системе Майя не может быть), то прибавим это число к HJ (то есть из этого большего числа вычитаем HJ) и получим какое-то число на которое данное число отстает от числа в системе Майя  $\Rightarrow$  (то есть вместо большего числа на месте 1 и 2 разряда ставим HJ и прибавим разность большего числа к сумме)  $\Rightarrow$  получаем итоговую сумму в системе счисления Майя.

Пример:

$$\begin{array}{r} 8BHJ \\ + A12 \\ \hline \end{array}$$

$$91J1 \Rightarrow J1 > HJ \Rightarrow J1 - HJ = 22 \Rightarrow$$

$$\Rightarrow 91HJ + 22 \Rightarrow 9200 + 21 \Rightarrow 9221.$$



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Тогда получаем следующий алгоритм на вход:

- 1) Складываем 2 числа без учета того, что на 2-ом разряде может быть только 18 или 20.
- 2) Сравниваем последние 2 разряда у складываемых ⇒ Если при их сложении во втором разряде может оказаться число больше ИТ (17+19) (или даже ненулевой 3-х разрядное число), тогда вычитаем из полученного числа (то которое > ИТ) число ИТ и прибавляем его к нашей основной сумме (из последних 2 разрядов вычитаем перед этим из ИТ); Если при сложении последних 2-х разрядов число получается число  $\leq$  ИТ, то ничего не вычитаем и записываем такую же сумму.
- 3) Пишем ответ.

№2) 1) Получаем на вход число. (в диапазоне от  $U$  до  $V$ ) (read ( $U, V$ )).

2) Возводим данное число в куб ( $U^3$  или  $V^3$ ).

3) Вычитаем из куба числа само число (получим некое число  $X$ ).

4) Если у числа  $X$  последние разряды (последние в кол-ве разрядов у входного числа; то есть, если у числа  $U$  - 4 разряда, то и у числа  $X$  должно быть в последних 4-х разрядах нули). Если в кол-ве разрядов у самого входного числа  $\Rightarrow$  это число триморфное.

Если же не проходит по 1-му данному условию, то число не триморфное.

5) Проанализируем (переберем) все числа в диапазоне от  $U$  до  $V$

6) Выводим все триморфные числа. (если их нет выводим «нет таковых»)

№3) 1)  $S = A + B + C$  |  $\Rightarrow$

2)  $B - A = C - B$  |  $\Rightarrow$

$\Rightarrow S$  кратно 3  $\Rightarrow$

$\Rightarrow$

$2B = A + C \Rightarrow S = 2B + B \Rightarrow S = 3B \Rightarrow$

```

var S, B: integer.
begin
  read (S)
  if (S mod 3) = 0 then B := S/3 and write (B)
  if (S mod 3) <> 0 then write ('ошибка')
end.

```





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

⇒ Алгоритм следующий:

- 1) считываем число  $S$  (read  $S$ )
- 2) Если при делении  $S$  на 3 число не дает остатка (то есть  $S:3$  без остатка) ⇒  $B$  присваиваем  $S:3$  ( $B := S/3$ ) ⇒ ответ:  $B = \frac{S}{3}$  (write ( $B$ )).
- Если же не делится без остатка ( $S \bmod 3 \neq 0$ ) ⇒  $\text{False}$
- ⇒ в ответе пишем «ошибка» (write ('ошибка')).

4 ~~64~~  $64^{513} = (2^5)^{513} = 2^{3078}$

- 1) Получим число  $2^{3078}$ .
- 2) Проверим все разряды справа налево.
- 3) Начиная с самого правого числа будем искать цифру 2 в каждом разряде ⇒
- ⇒ 4) Если мы нашли «2» ⇒ записываем номер разряда в свободную ячейку памяти.
- 5) Перебрав все разряды запишем все числа, которые мы откладываем в ячейки, в возрастающем порядке.
- 6) Выбор ответа.

- 5
- 1) Получаем на вход число (в диапазоне от 1 до  $V$ )
  - 2) По разрядам находим факториалы его цифр (цифр числа).
  - 3) Находим произведение найденных факториалов

- A) 4) Если конечное произведение равно исходному числу, то выведем это число как число факториалов первого рода.
- 5) Переберем все такие числа в диапазоне от 1 до  $V$ .
  - 6) Выведем все такие числа в ответ.

- B) 4) Если число не равно произведению цифр его факториалов, то берем следующее число в диапазоне от 1 до  $V$ .
- 5) Если таких чисел нет в ответ пишем «Нет таких чисел»



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

## Задание № 1

Нам дана двоичная запись в системе счисления Майя. В каком виде она дана этого не сказано. Предполагая, что она вводится так, как показано в условии ( (17)(19) или (1)(0)(0) ). Нетрудно заметить, что из-за того, что второй разряд содержит 18, а не 20 ступеней, то несколько изменится алгоритм перевода из системы Майя в десятичную систему счисления наоборот.

1) Алгоритм перевода из системы Майя в десятичную систему счисления:

(Заранее у нас уже подготовлено число в Майя, а именно оно хранится в виде динамического массива, в элементе которого разряд и т.д. Размер этого массива есть кол-во разрядов в числе Майя.)

У нас есть массив (динамический)  $M$  - само число и его надо преобразовать в число целого типа  $D$ .

1) Запускаем цикл от 1 до размера массива  $M$ .  
(переменная, которая отвечает за цикл -  $i$  (целый тип)).

Если  $i < 3$ , то  $D = D + M[i] \cdot 20^{i-1}$ .  
Иначе  $D = D + M[i] \cdot 20^{i-2} \cdot 18$ .  
(Внимательнее: степени есть во всех языках, а если и нет, то можно создать функцию возведения в степень)

конец цикла.

2) В переменной  $D$  - число Майя  $M$  в десятичной системе счисления. 3) Возвращаем число  $D$ .

4) Конец Алгоритма.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Итак, алгоритм ① из числа Майя в 20-ичной системе счисления делаем число в десятичной системе счисления и возвращаем это число.

Теперь нужно разработать алгоритм, который переводит число из десятичной системы счисления в систему Майя. Нам дано число  $D$  (целый тип), его нужно преобразовать в число  $M$  (динамический массив чьих-то чисел) - число Майя. Этот алгоритм будет работать как и обобщенный алгоритм перевода из десятичной в любую систему счисления, только когда мы второй раз будем делить, то нужно делить не на 20, а на 18, остальная часть алгоритма отменяться не будет.

② Алгоритм перевода числа из десятичной в систему Майя: (на вход поступает число  $D$ )

1) Пока  $D$  не равно 0 делаем цикл:

начало цикла

если размер  $M$  не равен 1, то  ~~$D = D \div 20$~~

1) добавляем в  $M$  число  $(D \bmod 20)$   
2)  $D = D \div 20$ .

иначе

1) добавляем в  $M$  число  $(D \bmod 18)$

2)  $D = D \div 18$ .

конец цикла

добавляем в  $M$  - добавляет в конец динамического массива  $M$  число.

$\bmod$  - остаток от деления

$\div$  - целая часть от деления

2) Возвращаем число  $M$ .

3) конец алгоритма.

Итак, алгоритм ② из числа, записанного в десятичной системе счисления делаем число



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Записанное в листе Майя. Это число представлено в виде динамического массива целых чисел, 1 элемент которого - 1 разряд, 2 элемент - 2 разряд и т.д.

Теперь запишем сам алгоритм решения задачи:

- 1) Введем два числа  $S_1$  и  $S_2$ .
- 2) Переведем  $S_1$  в  $D_1$  с помощью алгоритма (1), описанного выше.
- 3) Переведем  $S_2$  в  $D_2$  с помощью алгоритма (1).
- 4)  $D_3 = D_1 + D_2$ .
- 5) Преобразуем  $D_3$  в  $S_3$  с помощью алгоритма (2), описанного выше.
- 6) Выведем  $S_3$ .
- 7) Конец алгоритма.

вводима будет производиться по-урядно, т.е. сразу эта процедура ввода преобразует в динамический массив  $S_1$  и  $S_2$

$S_1, S_2$  и  $S_3$  - динамические массивы целых чисел.  
 $D_1, D_2, D_3$  - просто целые числа.

выводима будет осуществляться с конца массива  $S_3$ , т.е. начиная с последнего разряда и заканчивая первым.

Попробуем, что мой алгоритм считает два числа Майя, преобразуя их в динамические массивы. Потом преобразует эти два массива в целые числа. Складывает их. Преобразует новое число в динамический массив и выводит его, начиная с последнего разряда (т.е. разряды в массивах хранятся в обратном порядке).

(7)



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

## Задача № 2

Нетрудно заметить, что для проверки числа на триморфность не нужно хранить число, которое мы возводим в степень полностью, а достаточно хранить столько знаков, сколько было в проверяемом числе.

Приведу пример алгоритма, который проверяет число на триморфность:

нам дано число  $A$  - целого типа.

- 1) Посчитано длину числа  $A$  и записываем в переменную  $B$ .
- 2) Заведу переменную  $C = A$ , в которой буду хранить куб числа, причем она будет самого большого типа данных целых чисел (например, `long long` в `C++`), а  $A$  - меньшего.
- 3)  $C = (C \cdot A) \bmod 10^B$ . (mod-операция берет остаток от деления на число.)
- 4)  $C = (C \cdot A) \bmod 10^B$
- 5) Если  $C = A$ , то число  $A$  - триморфное иначе не триморфное.
- 6) Конец алгоритма.

Этот алгоритм быстро проверяет число на триморфность, но предполагается, что  $A$  не превосходит  $2 \cdot 10^9$  (т.к. иначе  $A$  не влезет в тип данных)



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Теперь найдем алгоритм решения самой задачи: предполагаем, что  $V$  больше чем  $U$ , если это не так, то выведем "некорректные данные".

1) Ввод  $U$  и  $V$  ( $U$  и  $V$  не должны превышать  $2 \cdot 10^9$ )

2) Обозначим  $m=0$ , где  $m$  - целый тип данных.

3) Запускаем цикл от  $U$  до  $V$  :

переменная-счетчик цикла:  $i$

1) проверяем  $i$  на триморфность с помощью алгоритма, описанного выше

2) если  $i$  - триморфное число, то выводим  $i$  иначе  $m = m + 1$ .

конец цикла.

4) если  $m = V - U + 1$ , то выводим "Нет таких чисел".

5) конец алгоритма.

С помощью данного алгоритма можно решить задачу с условием, что  $U$  и  $V$  будут "вызвать" в типе данных компьютера. Если же они не будут "вызвать" эти типы, то можно использовать алгоритмы цифровой арифметики.

### Задача n 3

Нам известно, что  $S = A + B + C$  и  $B - A = C - B$

$$\begin{cases} S = A + B + C \\ B - A = C - B \end{cases} \Rightarrow \begin{cases} S = A + B + C \\ 2B = C + A \end{cases} \Rightarrow S = \frac{3}{2}(A + C).$$

$$\frac{3}{2}(A + C) = A + C + B \Rightarrow B = \frac{A + C}{2} \text{ и } A + C = \frac{2}{3} S$$

Получается, что  $B = \frac{S}{3}$ .



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Мы немного упростим задачу:  
нам вводятся шло  $S$ .  
Если оно не кратно 3, то вывести "ошибка"  
иначе вывести  $S \div 3$ .

Казалось бы ничего сложного, но проблема в том, что  $S$  содержит не более 100 знаков, т.е. оно (шло  $S$ ) не помещается в обычной тип данных целых чисел. Для того чтобы решить эту задачу будем вводить  $S$ , как строку, а деление выполнять по алгоритму "деление столбиком".

Если шло кратно 3, то сумма его цифр так же кратно 3. Вынесем алгоритм проверки шло на кратность 3 в отдельный пункт:

нам понадобится  $sum$  - целый тип,  $sum = 0$ .

$i$  - переменная счетчик цикла, так же целый тип.

$c$  - целый тип.



1) Запускаем цикл от 1 до длины строки  $S$   
 переменная-счетчик -  $i$   
 преобразуем  $S[i]$  в  $c$ .  
 $sum = sum + c$ .  
 конец цикла.

$S[i]$  - обращение к  $i$ -ому элементу строки  $S$   
 а преобразование - перевод символа типа в целый

2) Если  $sum \bmod 3 = 0$ , то шло  $S$  - кратно 3  
 иначе шло  $S$  - не кратно 3.

$\bmod$  - операция взятия остатка от деления на шло

3) конец цикла.

Этот алгоритм проверяет кратно ли шло, записанное в виде строки, шло 3.



Теперь нужно разработать алгоритм, который делит число (возможно очень большое) на 3, причем убедиться, что это число точно разделилось на 3 без остатка.

Алгоритм принимает на вход строку и должен возвращать тоже строку. Выполнять деление будем «столбиком».

Нам понадобятся строки  $s1$ ,  $ans$ . На входе - строка  $S$ .

1) Начнём цикл: Пока  $S$  не равно "" (т.е. не пуста):

1)  $s1 = ""$ ; (пустая строка)

2)  $s1 = S[1]$  (первая цифра строки  $S$ )

3) удалим  $S[1]$ . (полностью удалим первый символ строки  $S$ )

4) while  $s1 < "3"$ , to ← строка "00001" будет меньше 3

Если  $s1 < "3"$ , то  $\left\{ \begin{array}{l} \text{в начале, когда мы занимаем} \\ \text{при делении столбиком к отве-} \\ \text{ту не приписываем 0, а не в на-} \\ \text{чале приписываем} \end{array} \right.$

$s1 = s1 + S[1]$

удалим  $S[1]$

Если  $ans$  не равно "", то  $ans = ans + "0"$

5) Преобразуем  $s1$  в число.

6) Преобразуем  $s1 \div 3$  в строку и добавим это в конец строки  $ans$ .

7) Преобразуем  $s1 \bmod 3$  в строку и добавим это в начало строки  $ans$ .

конец цикла.

$\left\{ \begin{array}{l} \text{все преобразования} \\ \text{строки в цикле работают} \\ \text{корректно, если впереди нули.} \end{array} \right.$

2) Вернем строку  $ans$ .

Этот алгоритм делит любое число, представленное в виде строки на 3.





Теперь запишем алгоритм решения задачи:

1) введем строку  $S$

2) Если  $S$  кратна 3, то выведем "Ошибка"

(алгоритм проверки числа, представленного в виде строки представлен выше)

иначе

выведем  $S \div 3$ .

( $S \div 3$  - деление числа на 3. Этот алгоритм так же представлен выше.)

3) конец алгоритма.

Итак, эти алгоритмы обеспечивают корректное решение задачи при любых числах  $S$ .

#### Задача № 4



В этом задании опять дано огромное число ( $64^{5^{13}}$ ), которое не влезет в стандартный тип данных, поэтому будем опять же пользоваться длинной арифметикой, а именно "умножением в столбик". Число будем хранить в виде динамического массива, где  $i$ -ый элемент указывает на  $i$ -ый разряд. Число будет храниться в 100-ичной системе счисления. Это необходимо для того, чтобы удобно выполнять умножение на 64.



Алгоритм решения задачи:

нам требуется пустой динамический массив целых чисел  $S$  целочислами  $i$  и  $j$ .

1)  ~~$S \leftarrow 64$~~  добавим в  $S$  64. ( $S$ -динам. массив).

2) Запустим цикл от 2 до 513: (счетчик  $i$ )

Запустим цикл от 1 до длины  $S$  (счетчик  $j$ )

$S[j] = S[j] * 64;$

$S[j] = S[j+1] + S[j] \pmod{64}$ ; (если  $S[j+1]$  нет, то добавить его, т.к.  $S$ -динам. массив)

$S[j] = S[j] \pmod{64};$

конец цикла.

конец цикла.

3) Запустим цикл от 1 до длины  $S$ .

если  $S[i] = 2$ , то вывести  $i$

конец цикла.

4) конец алгоритма.

Итак, этот алгоритм будет корректно работать с длиной массива ( $64^{513}$ ) и выведет все ряды, равные 2.

Задача № 5

1) Ввести  $n$  и  $V$  ( $n$  и  $V$ -целые типы)

2) Запустить цикл от  $n$  до  $V$  (счетчик цикла -  $i$ )

рассчитать факториалы каждого цифрот и сложить их.  
если эта сумма равна  $i$ , то вывести  $i$

конец цикла

3) конец алгоритма

Этот алгоритм проверит все числа в диапазоне от  $n$  до  $V$  и выведет все числа, которые надо.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

 $N_1$ 

для удобства:  $a$  модь-остаток от деления  $a$  на  $b$ ;

начало.

ввод  $(N_1, N_2)$ ;

\* Если  $N_2 > N_1$ , то  $x = \text{глубина}(N_2)$   
иначе  $x = \text{глубина}(N_1)$ ;

$x_1 = \text{глубина}(N_1)$ ;  $x_2 = \text{глубина}(N_2)$ ;

если  $x_1 > x_2$ , то  $\left\{ \begin{array}{l} x = x_1; \\ \text{добавить в запись } N_2 \text{ } (x_1 - x_2) \text{ нулей} \end{array} \right\}$

иначе  $\left\{ \begin{array}{l} x = x_2; \\ \text{добавить в запись } N_1 \text{ } (x_2 - x_1) \text{ нулей} \end{array} \right\}$ .

$\text{mas}[1] = (N_1[x] + N_2[x]) \bmod 20$ ; ~~если  $x > 1$ , то~~

~~$\text{mas}[2] = \left[ \frac{N_1[x-1] + N_2[x-1]}{20} \right] + \left[ \frac{N_1[x-1] + N_2[x-1]}{18} \right] \bmod 18$ ;~~

$\text{mas}[2] = \left[ \frac{N_1[x] + N_2[x]}{20} \right]$ ;

если  $x > 1$ , то

$\text{mas}[2] = (\text{mas}[2] + N_1[x-1] + N_2[x-1]) \bmod 18$ ;

$\text{mas}[3] = \left[ \frac{\text{mas}[2] + N_1[x-1] + N_2[x-1]}{18} \right]$ ;

если  $x > 2$ , то  $i = 3$

целые пока  $(i \leq x)$  цикл

$\left\{ \text{mas}[i] = (\text{mas}[i] + N_1[x+1-i] + N_2[x+1-i]) \bmod 20; \right.$

$\left. \text{mas}[i+1] = \left[ \frac{\text{mas}[i] + N_1[x+1-i] + N_2[x+1-i]}{20} \right]; \right.$

$i = i + 1$ ;

если  $\text{mas}(i) < 0$ , то ~~вывод~~  $\text{mas}(i)$ ;



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

цикл  $j = 0$  от  $(i-1)$  до 1, шаг 1;  
вывод  $(mas[j])$ ;

конец.

№2.

Известно, что число может оканчиваться не на все цифры, тогда безбумажно.

1) последняя цифра при возведении в куб может иметь

0-0  
1-1  
2-8  
4-4  
5-5  
6-6  
7-3  
8-2  
9-9

Однако вывод, что подходит только числа оканчивающиеся на 0, 1, 4, 5, 6, 9

2) Если последняя цифра 0, то число точно не триморфное, т.к. в кубе этого числа будет в 3 раза больше как минимум в конце.

Таким образом число должно оканчиваться на

1, 4, 5, 6, 9.

И

начало.

Ввод  $(a; v)$ ;

для удобства:

а то  $d$   $b$  - остаток от деления  $a$  на  $b$ .

цикл  $i = 0$  от  $u$  до  $v$ , шаг 1.

Если  $i \bmod 10 \in \{1, 4, 5, 6, 9\}$ , то  $\{x = \text{digit}(i);$   
 $a = i \cdot i \cdot i; j = 0; l = 0;$

$y = \text{digit}(a)$ ; пока  $(x \neq 0)$  делай

$\{ \text{если } a[j] \leq i[x-j], \text{ то } l = 1 \}$

$j = j + 1;$

Если  $l = 0$ , то вывод  $(i, 'l')$ ; } конец.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

NS  
I

$$B - A - C - B$$

$$2B - A = C$$

$$S = A + B + C$$

$$S = 3B$$

- 1) S делится на 3
- 2) если S делится на 3, и  $B \neq 1$ , то точно есть хотя бы 1-ый вариант числа A, B, C. т.к. можно взять  $A = B - 1$  и  $C = B + 1$ .
- 3)  $B = 1$  не подходит т.к. все  $A < B$  уже не положительные

II

начало  
вывод (S);

для удобства:  $a \bmod b$  - остаток от деления  $a$  на  $b$ .

Если  $(S \bmod 3 = 0)$  \* ~~и~~ и  $(S \neq 3)$ , то вывод ( $\frac{S}{3}$ )  
иначе вывод (Ошибка)

NS

раз так сильно шло передавать, значит на каждую цифру выделено сколько-либо бит, и по ней можно почитать отдельно.

Предположим это шло передано в одичном блоке (для удобства)

начало.

ans := 0;

пока не конец файла

{ вывод (a);

if  $a = 2$ , то ans := ans + 1; }

вывод (ans);

конец.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Если же число надо еще и поделить то следует  
 это тоже не сложно.  
 $64^{513} = (2^6)^{513} = 2^{3078}$  (можно обойтись, т.к. оно очень большое)

~~и так же~~

начало  
 \* ~~нужно учитывать в массиве 256~~ \*  
 \* ~~массив сегментирован bubble~~ \*

(7)

$i=3; i_1=i; \text{mas}[3]=6; \text{mas}[2]=5; \text{mas}[1]=2;$

цикл  $j$  от 0 до 3078, шаг 1  
 $\{ \text{ost}=0; i=i_1;$   
 пока  $\text{mas}[i] < 0$  цикл

~~$\text{ost}=0;$~~   
 $\text{ost}_i = \left[ \frac{\text{mas}[i] + \text{mas}[i] + \text{ost}}{10} \right];$   
 $\text{mas}[i] = \frac{\text{mas}[i] + \text{mas}[i] + \text{ost}}{\text{остаток от деления на } 10};$

$\text{ost} = \text{ost}_i;$

$i = i - 1;$

Если  $\text{ost} < 0$ , то  $i_1 = i + 1;$

$\text{mas}[i] = \text{ost}_i;$

Если  $\text{ost} \geq 0$ , то  $i = i + 1;$

цикл  $j$  от  $i$  до 3, шаг 1.

$\text{bubble}(\text{mas}[j]);$

кон.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

к 5

I сократил количество вариантов.

1) если число оканчивается на нечет. цифру, кроме 1.

То оно не может быть факториалом т.к.  $3! = 2 \cdot 3$   
 $5! = 2 \cdot 3 \cdot 4 \cdot 5$   
 $\vdots$

т.к. число должно быть четным

2) если число оканчивается на 6, 8, то оно должно по проверке факториалов цифр оканчиваться на 0 т.к. есть "5" и "2"

$$6! = 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6$$

$$8! = 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8$$

3) пусть число не "1"

тогда где-то в его записи есть цифра большая 1, а значит оно четное, а значит не оканчивается на 4.

4) оставшиеся варианты: 0, 2, 4.

II

начало  
 ввод (u, v);

цикл  $chis = 0$  от u до v, шаг 1.

{ если ~~chis~~  $chis \bmod 10 = \{0, 2, 4\}$ , то

{  $x = \text{getmax}(chis)$ ;  $st = 1$ ;

цикл  $i = 0$  от x до 1, шаг (-1)

цикл  $j = 0$  от 2 до  $chis[i]$ , шаг (1)

если  $(st = chis)$ , то ~~вывод~~  $st = st \cdot j$ ;  $st = st \cdot j$ ;  $st = st \cdot j$ ; если  $i \leq 1$ , то ~~вывод~~ (1)  
 конец.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

## Задача 2.

1. Вводим число  $U$  и число  $V$ .

1.1. Если  $U > V$ , то ~~меняем их местами~~ ОШИБКА Возврат

1.2. Если  $U$  не является числом или  $V$ , то ~~ВЫВОД~~ ОШИБКА Возврат

2. Запускаем цикл  $i$  от  $U$  до  $V$ .

2.1. Если  $i < 0$ , то возьмем модуль этого числа

2.2. Пусть  $t$  равно количеству цифр числа  $i$

2.3. Пусть  $q = 1$ , пригодится для сравнения.

2.4. Запускаем цикл  $j$  от 1 до  $t$

2.4.1.  $q = q \cdot 10$ ;

2.5. Если  $\text{mod}(i - i \cdot i, q) = i$ , то ~~ВЫВОД~~  $i$  с пробелом.

3. ~~Вывод~~ "Работа закончена"

## Задача 3.

1. Вводим число  $S$ .

1.1. Если  $S$  не является числом, то ~~ВЫВОД~~ "ОШИБКА" Возврат

X 2. Если  $S \geq 6$  и  $\text{mod}(S, 3) = 0$ , то ~~ВЫВОД~~  $\text{div}(S, 3)$ ,  
иначе ~~ВЫВОД~~ "ОШИБКА"

3. Вывод с новой строки "Работа закончена"

## Задача 4.

1. Вывод "Наливаю работу"

1.1. Переход на новую строку.

2.  $K = 64$  до возвращения в стену; ~~то~~ кол-во  $2$ ' в числе.

3. Запускаем цикл  $i$  от 1 до 513

3.1.  $K = K \cdot 64$

4. Пусть  $S$  - строка:

4.1. Сравни строгому представлению числа  $K$ .





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

4.2. Пусть  $n$  равно длине строки  $S$ .

5. Запускаем цикл  $i$  от  $n$  до  $1$ .

5.1. если  $S[i] = "2"$ , то Вывод  $n-i+1$ ; Вывод ПРОБЕЛ.  
 $t = t + 1$ ;

6. Если  $t = 0$ , то Вывод "2 не найдено"

6.1. Переход на новую строку.

7. Вывод "Работа закончена"

### Задача 5.

1. Вводим  $u$  и  $v$ .

1.1. Число  $t$  - количество факторизаций.

1.2. Если  $u$  или  $v$  не являются числом, то Вывод "Ошибка при вводе"; Возврат к пункту 1.

1.3. Если  $u > v$ , то меняем их местами.

2. Запускаем цикл  $i$  от  $n$  до  $v$

2.1. Пусть  $SUM = 0$  - сумма факторизаций.

2.2. Пусть  $S$  - строка, представление числа  $i$  в строковом виде.

2.3. Пусть  $n$  - длина строки  $S$ .

2.4. запускаем цикл  $j$  от  $1$  до  $n$ .

2.4.1. Пусть  $m = 1$ , это факторизация;  $k = \text{число}(S[j])$

2.4.2. если  $S[j] \neq "0"$ , то запускаем цикл  $p$  от  $1$  до  $k$

2.4.2.1  $m = m \cdot p$ ; иначе  $m = 0$

2.4.3  $SUM = SUM + m$ ;

2.5. если  $SUM = i$ , то Вывод  $i$  с пробелом;  $t = t + 1$

3. Если  $t = 0$ , то Вывод "Не найдено"

3.1. Переход на новую строку.

4. Вывод "Работа закончена"





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

## Задача 1.

1. Вводим  $n, m$  - кол-во разрядов в 1 и 2 числе.

Так как числа бывают натуральными, то числа будем представлять в виде массива  $a[n], b[m]$ .

1.1. Если  $n$  или  $m$  не являются числами, то вывод "Ошибка при входе" Возврат к пункту 1.

2. Запускаем цикл  $i$  от 1 до  $n$

2.1. Ввод  $a[i]$

3. Запускаем цикл  $j$  от 1 до  $m$

3.1. Ввод  $b[j]$

4. Если  $n \geq m$ , то

5.  $k = n$ , где сложение.

6.  $r = 0$

7. Запускаем цикл  $i$  от  $m$  до 1.

7.1.  $a[k] = a[k] + b[i] + r$

7.2. если  $(n - k + 1) = 2$ , то

7.2.1.  $r = \text{div}(a[k], 10)$

7.2.2.  $a[k] = \text{mod}(a[k], 10)$

иначе

7.2.1.  $r = \text{div}(a[k], 20)$

7.2.2.  $a[k] = \text{mod}(a[k], 20)$

7.3.  $k = k - 1$

иначе 5.  $k = m$ , где сложение

6.  $r = 0$

7. Запускаем цикл  $i$  от  $n$  до 1.

7.1.  $b[k] = b[k] + a[i] + r$

7.2. если  $(m - k + 1) = 2$ , то

7.2.1.  $r = \text{div}(b[k], 10)$

7.2.2.  $b[k] = \text{mod}(b[k], 10)$

иначе 7.2.1.  $r = \text{div}(b[k], 20)$

7.2.2.  $b[k] = \text{mod}(b[k], 20)$

7.3.  $k = k - 1$

8. Если  $n \geq m$ , то 8.1. Вывод  $r$  (если  $r = 0$ , не идет на увеличение)

9.2. Запускаем цикл  $i$  от 1 до  $n$

9.2.1. Вывод  $a[i]$

иначе 9.1. Вывод  $r$

9.2. Запускаем цикл  $j$  от 1 до  $m$ , 9.2.1. Вывод  $b[j]$

10. Вывод "Работа закончена"





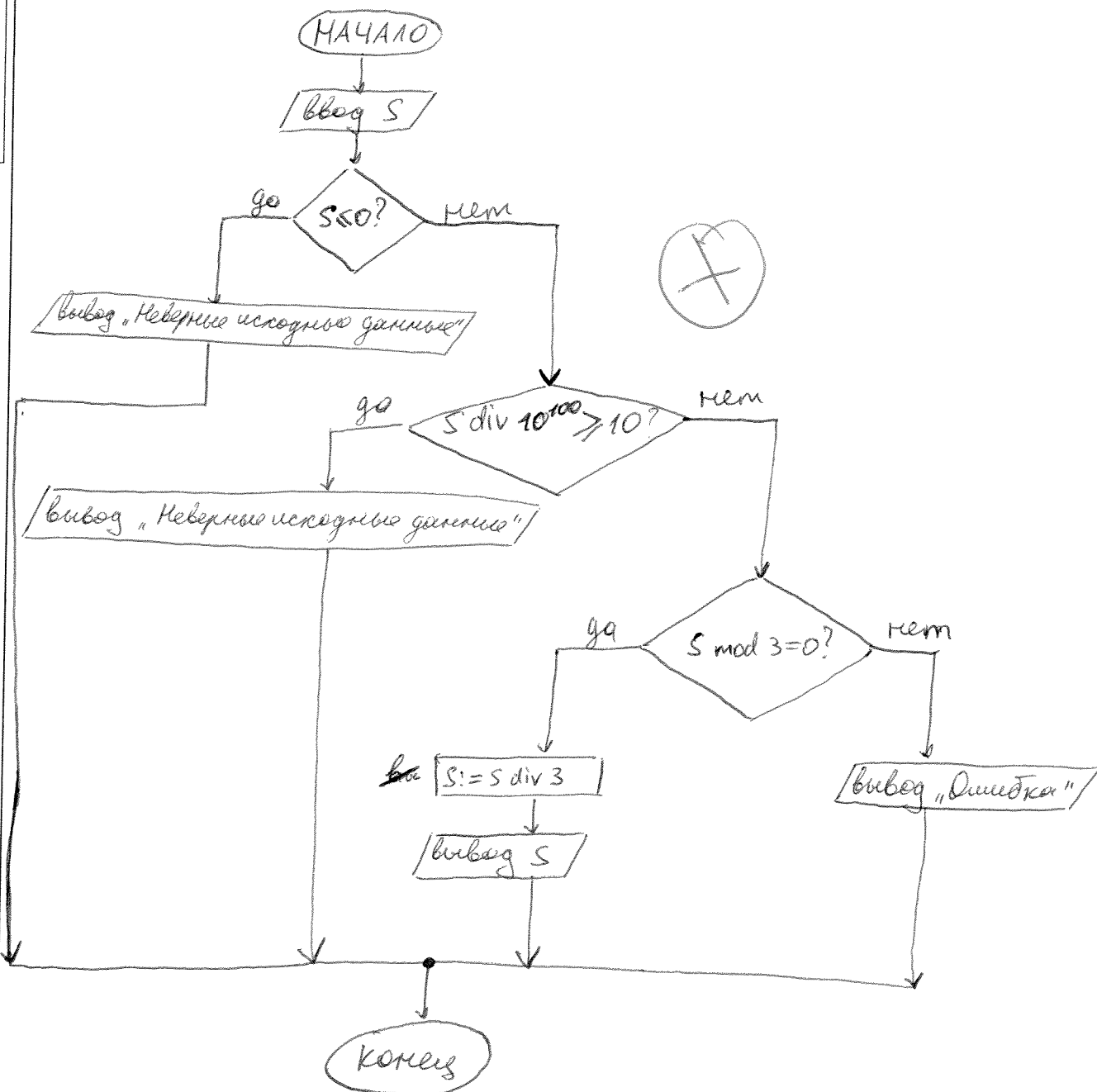
ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

- ③
1.  $S = A + B + C$  (1)
  2.  $B - A = C - B$   
 $-A = C - 2B$   
 $A = 2B - C$
  3. Подставить значение  $A$  в первое ~~уравнение~~ равенство др. выражение

$S = 2B - C + B + C$  („C“ сокращается)

$S = 3B$ .

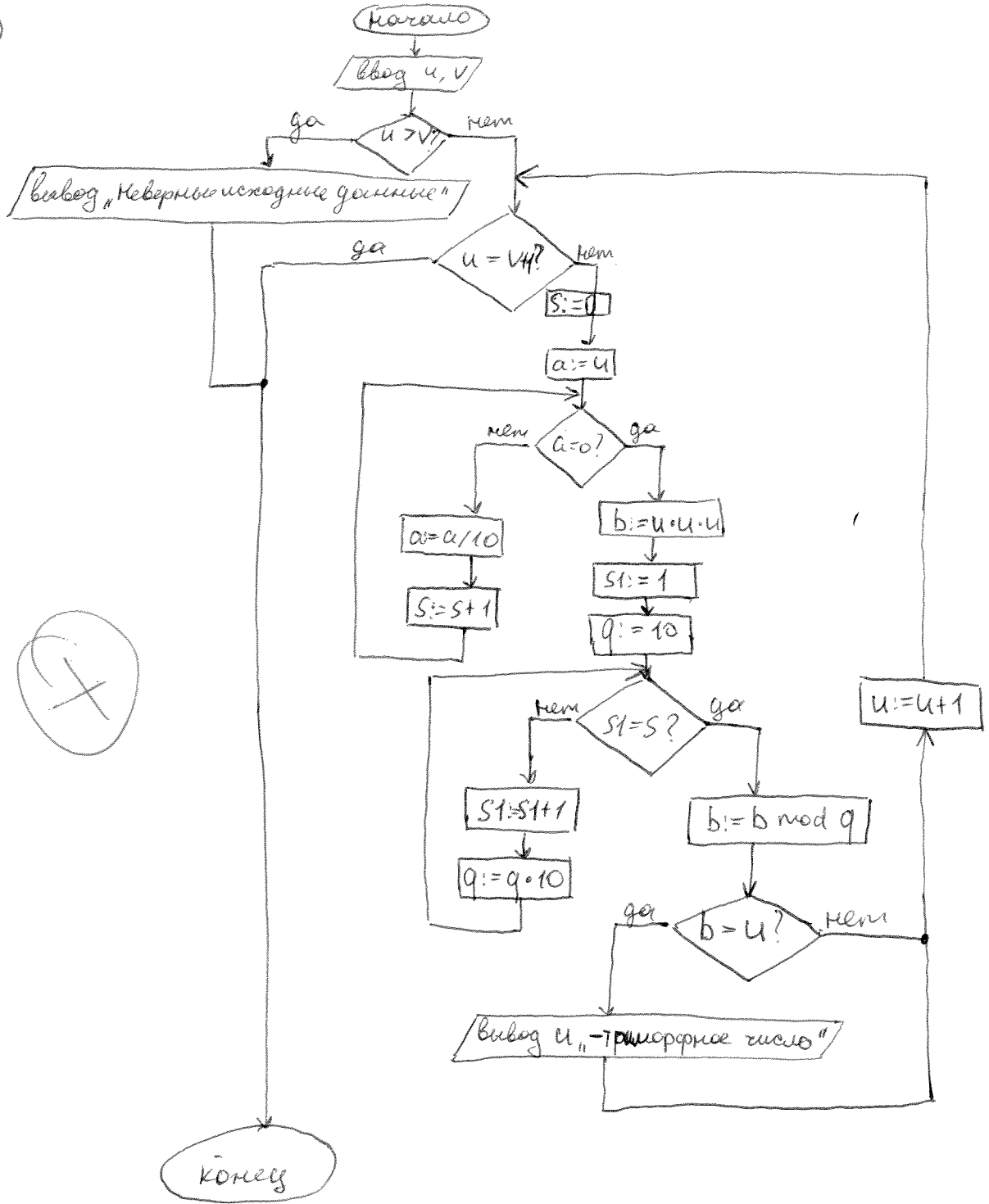
⇒ если ~~S~~ S делится на 3 без остатка, то результатом от деления и есть число B, иначе „Ошибка“





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

2





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

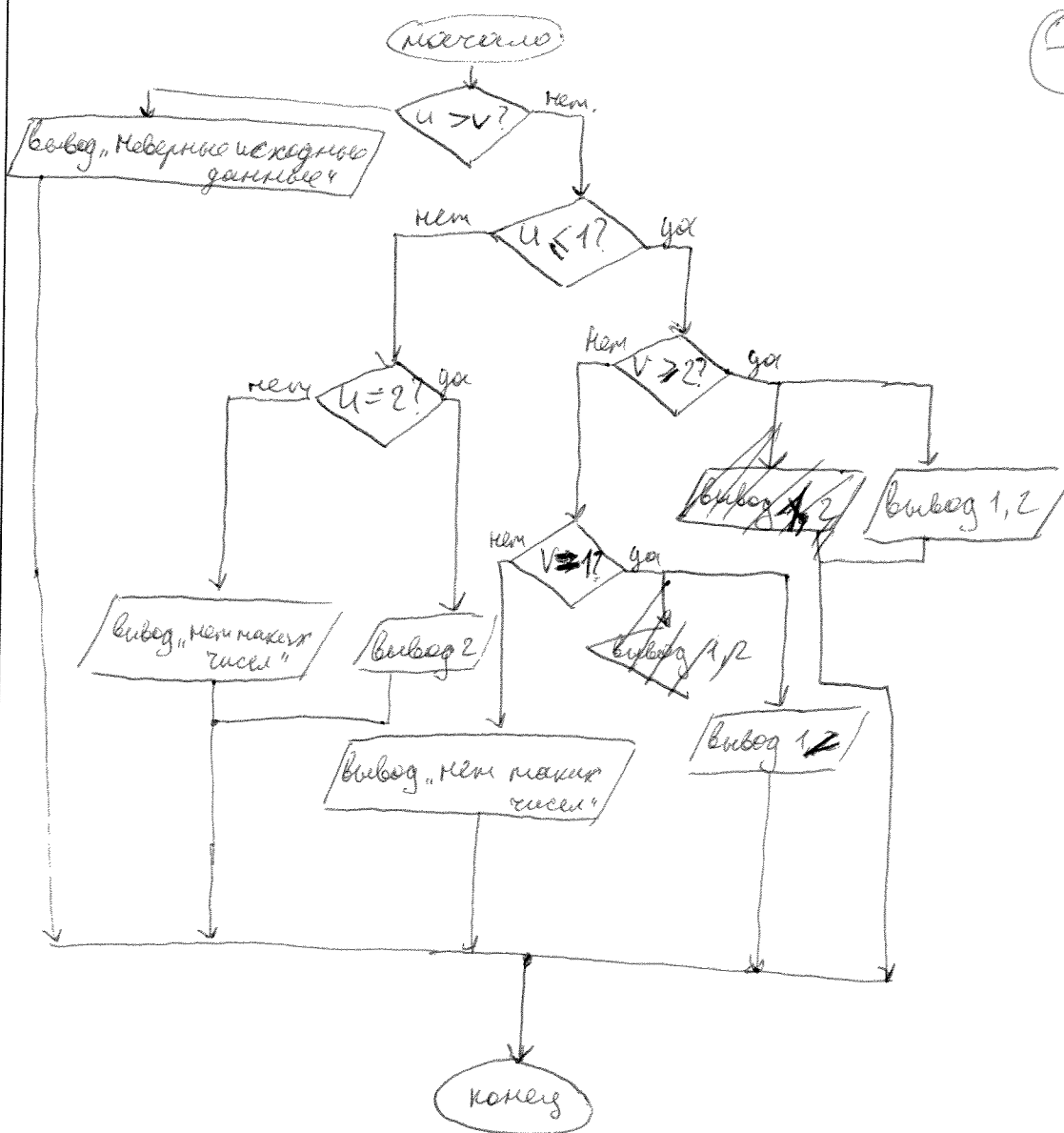
- ① 1. Смазываем число по разрядам  
 2. Если у ~~каждого~~ одной числа количество разрядов меньше, чем у другого, то вместо недостающих разрядов ставим нули.  
 3. Если <sup>однаковая</sup> сумма разрядов одинаковая:  
 а) Если это второй разряд, больше 17, то  
 а) больше 17 и это второй разряд, то в разряд числа, которое мы хотим найти записываем остаток от деления суммы разрядов на 17 и к сумме следующих разрядов прибавим 1, когда будем суммировать их.  
 б) меньше 18 и это второй разряд, то в разряд числа, которое мы хотим найти записываем сумму разрядов данного числа.  
 в) больше 19 и это не второй разряд, то в разряд числа, которое мы хотим найти записываем остаток от деления суммы разрядов на 20 и к сумме следующих разрядов прибавим 1, когда будем суммировать их.  
 г) меньше 20 и это не второй разряд, то в разряд числа, которое мы хотим найти записываем сумму разрядов данного числа.
4. Мы получили сумму чисел.

F



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

- 5) 1. При анализе чисел от 0 до 199 факториальными первого рода являются только числа 1 и 2.
2. При анализе чисел от 199 разница между произведением факториалов цифр и самим числом будет возрастать по сравнению с числом предыдущего 3 разряда («предыдущей сотни»)





ВНИМАНИЕ! Проверяется только то, что записано  
с этой стороны листа в рамке справа

④ 1. Ищем во всех словах одинаковые буквы,  
но чтобы в каждом слове было более одной такой  
буквы. Например: сокол  
стойка } во всех словах есть "с", но  
спулер } в каждом слове по одной,  
значит её в слове не берем.

2. Для каждого слова и каждой буквы из п. 1.  
сохраняем разницу позиций этих букв в слове и  
номер позиции  
Например: сосна позиция: 1, 3  
разность: 2.

3. Сравниваем разности всех слов, нужно  
чтобы разности каких-либо двух слов совпа-  
дали и разности оставшихся двух слов совпа-  
дали.

4. Если нашли такие пары слов (из п. 3), то  
одна пара будет стоять вертикально так,  
что одинаковые буквы (из п. 1) были на одном  
уровне, а другая по горизонтали при  
тех же условиях (или наоборот: первая пара  
по горизонтали, вторая - по вертикали).

5. Выведем получившийся крестовод.

⑤



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

②

- Создаем цикл, который работает до момента пока  $u < v + 1$  ;
- Возводим  $u$  в куб или перемножаем 3 раза и выведем
- в переменную  $a$  получаемый результат
- переменной  $k$  задаем значение переменной  $u$
- создаем 2 цикла, который будет повторяться пока число  $u$  не станет равно 0

1)  $p$  присвоить  $p + 1$  (возникнет 0)2)  $u$  присвоить целочисленное деление  $u$  на 10создаем цикл повторяющийся  $p$  раз1)  $q$  присвоившим значение  $q \cdot 10$  (узнаем 1)

- присваиваем в значение целочисленного деления  $q$  на  $q$
- если  $v = k$  выводим число  $k$

конечу;

~~конечу.~~переменной  $k$  присваиваем значение  $S$ 

③

- Создаем цикл который работает пока  $S > 0$  ;

1)  $p$  присвоим  ~~$p + 1$~~  (возникнет 0)  
целочисленное деление числа  $S$  на  $10 + p$   
(возникнет 0)2)  $S$  присвоим  $S$  целочисленное деление на 10

конечу

если  $p$  кратно 3 выводим  $B = k : 3$  если не равно 3 выводим ошибку.

⑤

- Создаем цикл ~~который~~ работающий пока  $u < v + 1$  ;

•  $k$  присвоим значение  $k_1$  и  $k_2$  присвоим значение  $u$ • ~~создаем цикл  $u$  с помощью  $u$~~ • цикл : пока  $k > 0$  делаем вывод•  $v$  присвоим значение  $v + 1$  (возникнет 0)•  $k$  присвоим значение  $k$  целочисленное деление на 10•  $v$  делим число  $u$ •  $D$  присвоим значение 1 и  $Q$  присвоим знат. 1цикл повторяющийся  $v$  раз с выводом  $v + 1$  число $t$  присвоим значение целочисленное деление  $k_2$  на  $k_0$  $k_2$  присвоим значение целочисленное деление  $k_2$  на 10





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

второй цикл равенности кол-во вхождений которого  $m$   
символов  $i$

начинаю

$Q := Q \cdot i$  (первоначально  $i = 1$ )

$i := i + 1$

конечу

D присвоить значение  $D * Q$

Q присвоить значение 1; i присвоить значение 1

конечу

если  $n$  равно D вывести  $n$

конечу

① т.к. компьютер не может воспринимать  
19 как 1 цифру. Используя визуальную  
таблицу эквивалентности цифр каждой строки имеет  
все значения

Символ	Числовое значение	Символ	Чис. знач.
" 1 "	1	" K "	17
" 2 "	2	" L "	18
" 3 "	3	" M "	19
" 4 "	4		
" 5 "	5		
" 6 "	6		
" 7 "	7		
" 8 "	8		
" 9 "	9		
" A "	10		
" B "	11		
" C "	12		
" D "	13		
" E "	14		
" F "	15		
" H "	16		

В римском числе в компьютер  
вводится в виде строки  
символов и выводится также в  
виде строки.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

- Узнаем длину строки 1 (- 1 выделено число) -  $l_1$
- Узнаем длину строки 2 (- 2 выделено число) -  $l_2$
- 1) ~~также переводим число 1~~ К переводим  $z$  1;  $y := 0$ ;  $q = 0$ ;
- Цикл повторения столько раз сколько максимальная длина строки ~~массив~~
- ~~К переводим~~
- $a$  присваиваем значение ~~символа~~  $l_1$  члене строки 1
- $b$  присваиваем значение  $l_2$  члене строки 2
- специализированные кодировщики:
  - $a$  присваиваем значение ~~символа~~  $a$
  - $b$  присваиваем значение символа  $b$
  - $q$  присваиваем значение  $a + b + q$
- Цикл повторения 2 раза ~~массив~~
- если  $c > 19$  и  $k \neq 2$  тогда  $y = c - 19$  и  $q = q + 1$ ;
- если  $c > 19$  и  $k = 2$  тогда  $y = c - 17$  и  $q = q + 1$ ;
- конечу
- К присваиваем  $k + 1$ ;  $y$  переводим в символ по таблице кодирования и складываем с пустой строкой  $W$ ;
- конечу.
- $W$  - строка с ответом максимального перевода.
- узнаем длину  $l$  строки  $W$
- цикл повторения  $l$  раз ~~массив~~
- М пустой строке присвоим символ из строки  $W$  по номеру  $l$
- $l$  присваиваем значение  $l - 1$
- конечу.
- Выводим строку  $M$
- конечу.



④



④

Пусть первое слово - A Второе слово - B третье слово с числом - D,  
и переменной  $k=0$

Ищем длины строк A, B, C, D  
их длины соответственно равны  $l_A, l_B, l_C, l_D$

Циклы повторяющийся  $l_A \cdot l_B$  раз можно  
сравнить каждую символ строки A с каждым символом строки B

~~если цифра строки A  $\neq$  равно этой цифре~~ и  $k=0!$

если символ строки A равен символу строки B, то  
X - присваиваем символ, и XB номер этого символа в строке B  
и XA номер этого символа в строке A.

и делаем начало

циклы повторяющийся  $l_B \cdot l_C$  раз можно  
сравнить каждую символ строки A с каждым символом строки C

если символ строки A равен символу строки B и  
 $k=0$  тогда делаем Y присваиваем значение этого символа,  
Yc - номер этого символа в строке C и

YA - номер этого символа в строке A и  
Y - значение этого символа и делаем

начало

циклы повторяющийся  $l_B \cdot l_C \cdot l_D$  раз

ищем одинаковые символы в строке D и B и ~~показываем~~  
~~ZB в строке B и если нашли ZB~~ и ~~то~~ = номер в строке B  
ZD - номер в строке D

ищем одинаковые символы в строке D и C  
если нашли  $Yc = \dots$  номер в строке C и  $YD = \dots$  номер в строке D

если  $YD - YD = YA - XA$  и  $YB - XB = Yc - Yc$ , то

$k := 1$ ;

конеч; конек; конез; конез; кониз; кониз

или вводящие, такой кроссворд существует

Сам кроссворд мы можем ввести с помощью "картинки"  
повернуть выисовываем каждую символ. т.к. мы знаем длину  
строки буквы, то можем найти координаты  
каждой буквы. к примеру пусть 1 слово помещается в координате  
(100; 100) тогда второе слово имеет начало в  $(100 + XA; 100 + XB)$   
и вылезает в строке символов. Слово  $(100 + YA; 100 + Yc)$   
и слово  $(100 + YD; 100 + YB + YD)$



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

1.

Запишем первое число в строку  $S_1$ Запишем второе число в строку  $S_2$ 

Будем записывать так, что в строке  $S_1$  должно находиться число с наибольшим количеством разрядов  
 Запустим счетчик  $l=0$

Запускаем цикл от  $j=1$  до длины строки  $S_1$

(цифра с разрядом идет с конца строки)

(пропускаем ячейки строки с конца)

В каждой ячейке строки красится цифра в 20-тичной системе счисления

запускаем число  $n$ в число  $x$  записываем число  $u_j$   $j$ -той ячейки  $S_1$ в число  $y$  записываем число  $v_j$   $j$ -той ячейки  $S_2$ 

$$x + y = m \quad m = a + b$$

выполняется  ~~$n = (n \bmod 20) + l \bmod 20$~~ ;  $n = (m + l) \bmod 20$ ;  
 если  $j \neq 2$   ~~$n = (n \bmod 20) + l \bmod 20$~~   $l = (m + l) \div 20$ ;

~~$n = (n \bmod 11) + l \bmod 11$~~

(7)

выполняется  $n = (m + l) \bmod 18$ ;  
 если  $j \neq 2$   $l = (m + l) \div 18$ ;

далее складываем число  $n$  в строку  $j$ -тую  
 ячейку строки  $S_3$  \*1(mod - остаток от деления)

конец цикла \*2(div - целое от деления)

Выводим  $S_3$ 

2.

Запускаем цикл от  $i=1$  до  $V$ записываем число  $i$  в строку  $S$ запустим счетчик  $n=0$ Запускаем цикл от  $j=1$  до длины строки  $S$ запишем в строку  $S_2$   $j$ -ую цифруцифры из строки  $S$ Переведем число из строки  $S_2$  в переменную  $l$ Переведем  $j$ -тый символ из строки  $S$  в число  $m$ 

$$\text{далее } n = n + m * l * 10^{n-j}$$

конец цикла

Если  $n = i$ , тогда это число является  
 тригонометрическим

конец цикла.





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

3. Запишем исходное число  $n$  в строку  $s$ .  
 $K=0$  (запускаем счетчик)  
 Запускаем цикл от  $i=1$  до длины строки  $s$   
 переводим  $i$ -тый символ строки  $s$  в переменной  $x$   
 складываем в счетчик  $K$  переменной  $x$  ( $K=K+x$ )  
 конец цикла  
 Если остаток от деления счетчика  $K$  на 3 равен нулю ( $K \text{ mod } 3 = 0$ ), то выводим число  $c = \frac{n}{3}$ ,  
 иначе выводим слово "ошибка"



5.

массив

0!	1!	2!	3!	4!	5!	6!	7!	8!	9!
1	2	3	4	5	6	7	8	9	10

Заполним ячейки массива 1-10 массива  $K$  как показано на рисунке. Запускаем счетчик  $n=0$   
 Запишем исходное число  $n$  в строку  $s$   
 Запускаем цикл от  $i=1$  до длины строки  $s$   
 переводим  $i$ -тый символ строки  $s$  в число  $x$   
 Обращаемся к ячейке массива по номеру  $(x-1)$ . Пусть это число равно  $y$ .  
 Если  $n \leq i$ , то  $n = n \oplus y$ .  
 Конец цикла  
 Если  $n$  равно "исходному числу", то это число является факториалом первого рода  
 Я привел алгоритм для  $\Phi$  одного числа, поэтому необходимо  
 запустить цикл от  $j=0$  до  $V$   
 (где число  $j$  является "исходным числом")  
 Производим следующие операции →



конец цикла.



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

4. Пусть в строках  $s_1, s_2, s_3, s_4$  хранятся слова.

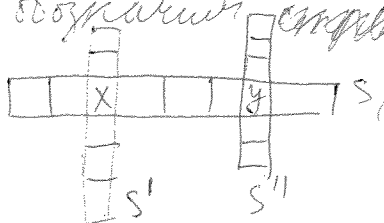
Берем ~~строку~~ первую строку  $s_1$

Затем берем вторую (любую из  $s_2, s_3$  и  $s_4$ )

И для каждой ячейки строки  $s_1$  начинаем перебор всех ячеек строки  $s_2$ . Если  $i$ -тая ячейка строки  $s_1$  ~~и~~ и  $j$ -тая ячейка строки  $s_2$  заполнены одинаковыми символами, то (обозначим эту ячейку  $x$  в  $s_1$ )

Берем еще одну строку ~~с~~ (которая еще не учитывалась) и так же начинаем перебор ячеек строки  $s_1$  с данной строки (но не перебираем ячейку  $x$ )

Если нашли одинаковые символы, то обозначим ~~строку~~ такую ячейку  $y$



Затем берем оставшуюся строку и начинаем подбирать ячейки со строками  $s'$  и  $s''$  одновременно

Путем такого перебора строк можно определить наличие пересечения из данных ~~слов~~ слов



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

N3 Так как  $B-A=C-B$ , то  $2B=A+C$   $B=\frac{1}{2}(A+C)$   
 Так как  $S=A+B+C$ , то  $S=A+C+B=A+C+\frac{1}{2}(A+C)=3\cdot\frac{1}{2}(A+C)=3B$ , тогда  $B=\frac{1}{3}S$ .

Программа на языке ~~PASC~~ Pascal ABC.net:

```

var
  s: string;
  i, l, c, x, r: longint;
begin
  read(s); {читываю слово S в строку т.к. не помещается в целочисл.}
  l:=length(s); {нахожу длину строки s}
  for i:=1 to l do c:=c+a[i]; {нахожу сумму цифр в числе s}
  if c mod 3 <> 0 then {если сумма цифр не равна 0 (делится на 3),}
    begin
      writeln('Ошибка'); {сообщаю об этом пользователю}
      exit; {полностью выхожу из программы}
    end;
  A: for i:=1 to l do {если в программе остались, то делю на 3 столбиком}
    begin
      x:=(10*r+a[i]) div 3; {нахожу целую часть деления}
      r:=(10*r+a[i]) mod 3; {остаток, для дальнейшего деления}
      if not((i=1) and (x=0)) then {исключаю случаи первой цифры 0}
        write(x); {записываю/вывожу на экран результаты деления}
      end;
    end;
  end.
  
```

Для уменьшения времени работы программы в место А можно вставить

```

код кулик:
if l < 9 then write {проверяю на вместимость в тип longint}
begin
  x:=strtoint(s); {перевожу строку в целочисл. longint}
  write(x div 3); {сразу вывожу результат}
  exit; {выхожу из программы}
end;
  
```

N4 Ясно, что  $i$ -тый покупатель будет находиться в очереди столько, сколько слов перед ним  $(i-1)$ -ого покупателя.

Программа на языке Pascal ABC.net:

```

var
  n, i, min: longint;
  c, t: array[0..10000] of longint;
begin
  read(n); {читываю кол-во покупателей};
  for i:=1 to n do read(t[i]); {читываю время обслуживания покупателя i};
  min:=t[1]; {предполагаю, что минимальное время обслуживания у 1-го покупателя}
  for i:=1 to n do
    begin
      A: c[i]=c[i-1]+t[i-1]; {нахожу время i-го покупателя в очереди}
      if t[i]<min then min:=t[i]; {нахожу минимальное время обслуживания}
    end;
  writeln(min); {вывожу минимальное время обслуживания};
  for i:=1 to n do write(c[i], ' '); {вывожу время прибывания в очередь i-го покупателя}
  end.
  
```

Если предполагается, что покупатель стоит в очереди, когда его обслуживают, то строчка А заменяется на строку:  $c[i]:=c[i-1]+t[i]$ ; т.е. включает



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Время обслуживания.

N2

```

var
  i, c, n: longint;
  a, b: array [1..10000] of longint;
begin
  read (n); { считываем кон-во чисел }
  for i:=1 to n do read (a[i]); { считываем числа }
  for i:=1 to n-1 do { сортируем числа в порядке возрастания }
    for j:=i+1 to n do
      begin
        if a[j] < a[i] then
          begin
            c:=a[i];
            a[i]:=a[j];
            a[j]:=c;
          end;
      end;
  c:=1;
  for i:=1 to n do { проверяем каждое число на чётность }
    if a[i] mod 2 = 0 then { если число чётное, то }
      begin
        b[c]:=a[i]; { во втором массиве ставим его за другими чётными }
        c:=c+1;
      end
    else
      b[n-i+1]:=a[i]; { ставим нечётные числа с конца }
  for i:=1 to n do { выписываем все числа от 1 до n, в нужном порядке }
    write (b[i], ' '); { в новом массиве из нового массива }
  end.

```



N1 Пусть алгоритм заключается в том, что  $x$  сначала переводится в с. Майя число  $\text{div } 360$ , и определяем  $\text{число mod } 360$ , а потом переводим полученное число в двоич.

Программа Pascal ABC.net:

```

var
  n, k, i, j: longint;
  a: array [1..10000] of longint;
begin
  read (n); { считываем число в десятичной системе }
  if

```

N1 Программа Pascal ABC.net:

```

var
  n, c, i: longint;
  a: array [1..10000] of longint;
begin
  read (n); { считываем десятичное число }
  while n < 70 do { повторяем цикл, пока n < 70 }
    begin
      c:=c+1; { узнаем номер разряда, который читано }
      if c = 2 then { исключаем случаи со вторым разрядом (учитываем его) }
        begin
          a[c]:=n mod 18; { запоминаем число, находящееся во втором разряде с.с. Майя }
          n:=n div 18; { уменьшаем значение, с учётом разряда }
        end
      else

```





ВНИМАНИЕ! Проверяется только то, что записано  
с этой стороны листа в рамке справа

```
begin
  a[0] := n mod 20; {используя 20-тиричную с.с.4}
  n := n div 20; {присваивая и новое значение;}
end;
```

```
for i := c downto 1 do write ('(', a[i], ','); {вписываем все остатки в обратном порядке;}
end.
```

$$\sqrt{5} \quad 64^{513} = 64^{3 \cdot 19} = (2^6)^{3 \cdot 19} = 2^{30 \cdot 19}$$

Цифра 6 будет находиться в десятичной записи, если в числе, в 2 раза меньше этого, будет цифра 8 (после неё цифра не более 4), ~~или цифра 7 (после неё цифра более 4) или цифра 3 (после неё цифра не более 4)~~. Цифра 8 будет находиться в десятичной записи, если в числе, в 2 раза меньше этого, будет цифра 4 (после неё цифра не более 4) или цифра 9 (после неё цифра не более 4). ~~и т.д.~~ Цифра 3, если до этого цифра 6 (после неё цифра более 4) или 1 (после, цифра более 4) и т.д.

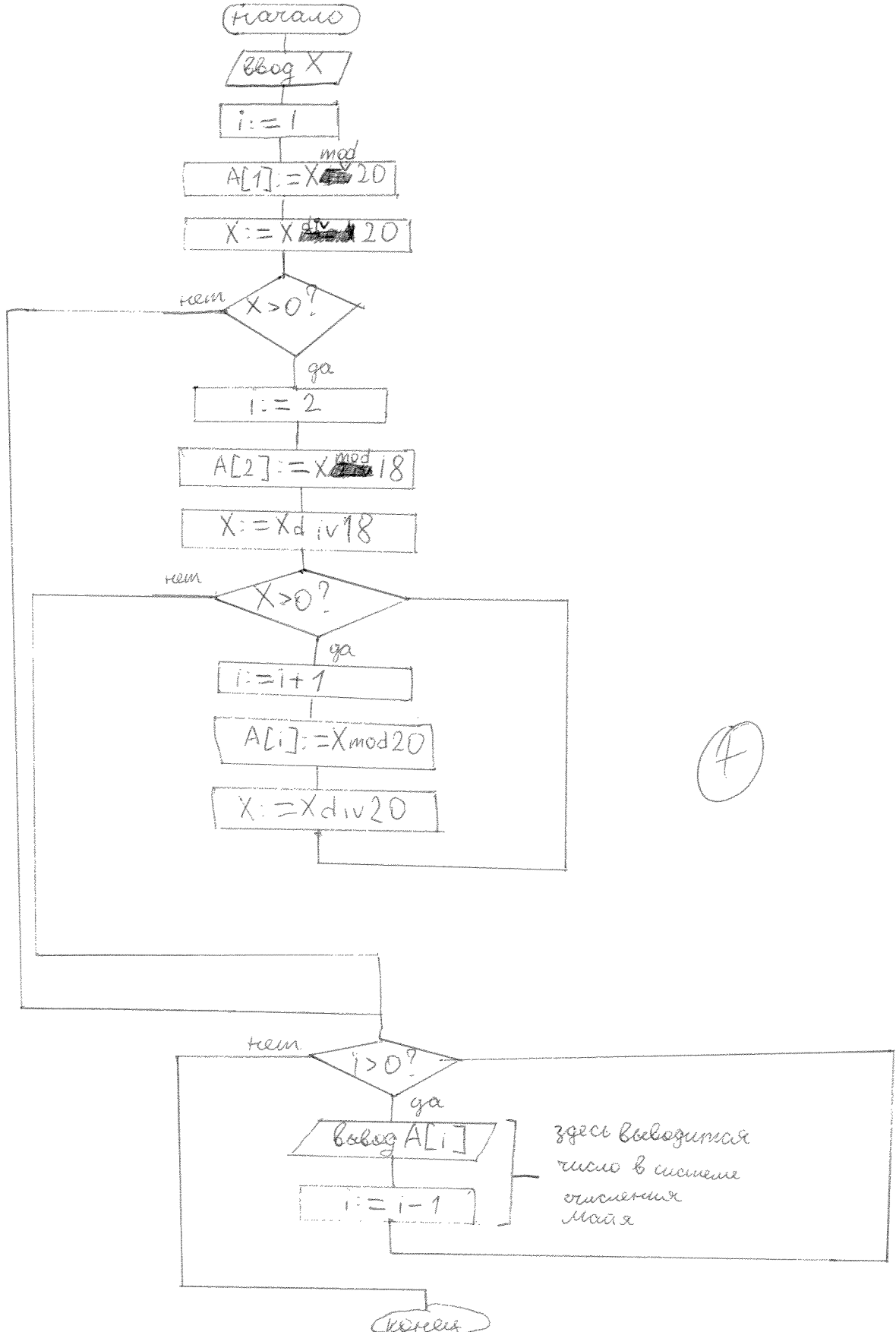




ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

Задача 1

Пусть  $X$  - число в десятичной системе  
 $A$  - массив из целых чисел  $\geq 0$   
 $i$  - счетчик



7



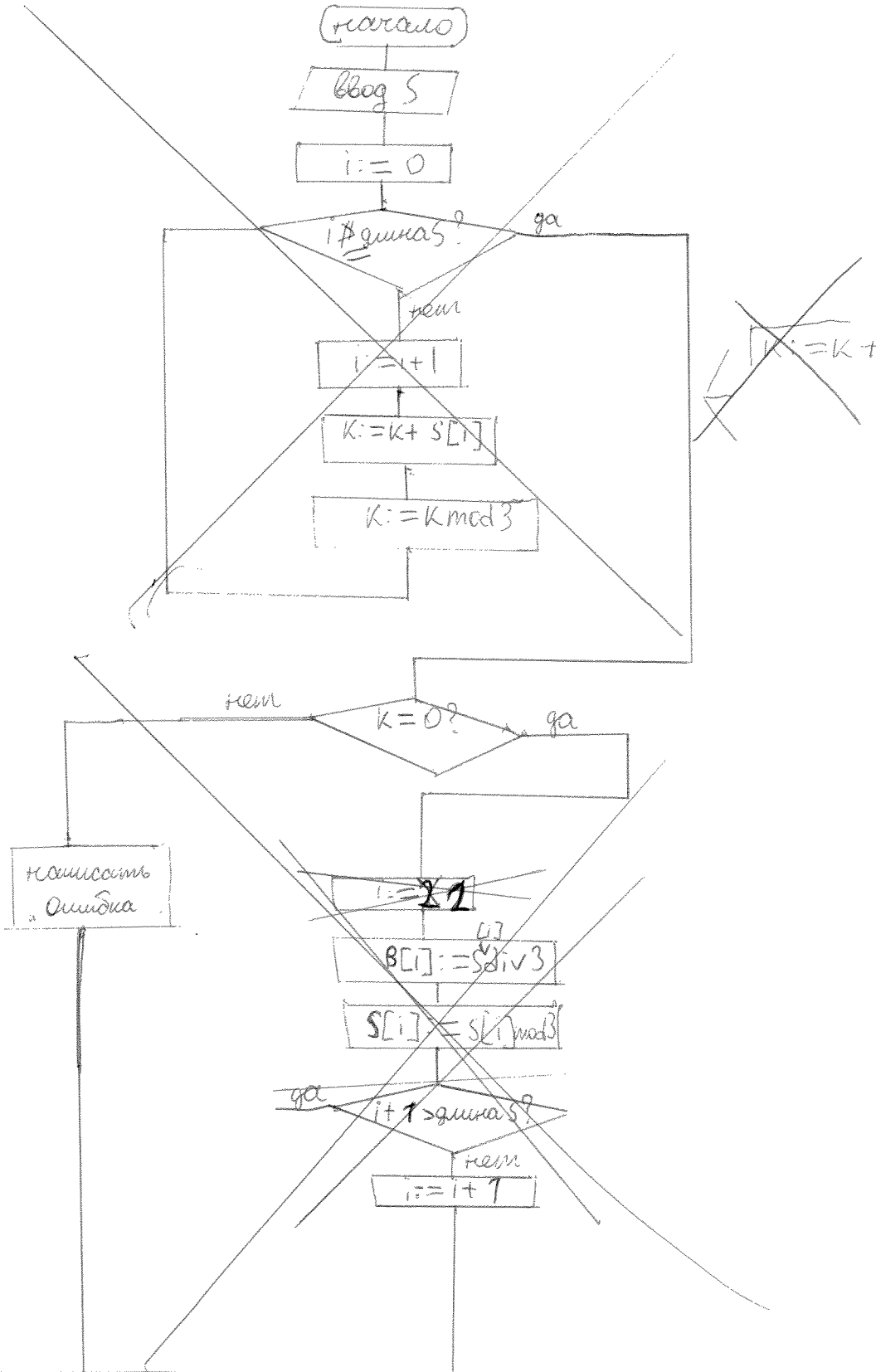
ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

задание 3

$$\left. \begin{aligned} S &= A+B+C \\ B-A &= C-B \Rightarrow 2B = A+C \end{aligned} \right\} \Rightarrow B = S/3$$

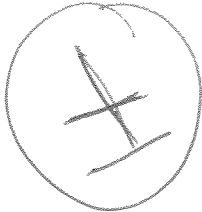
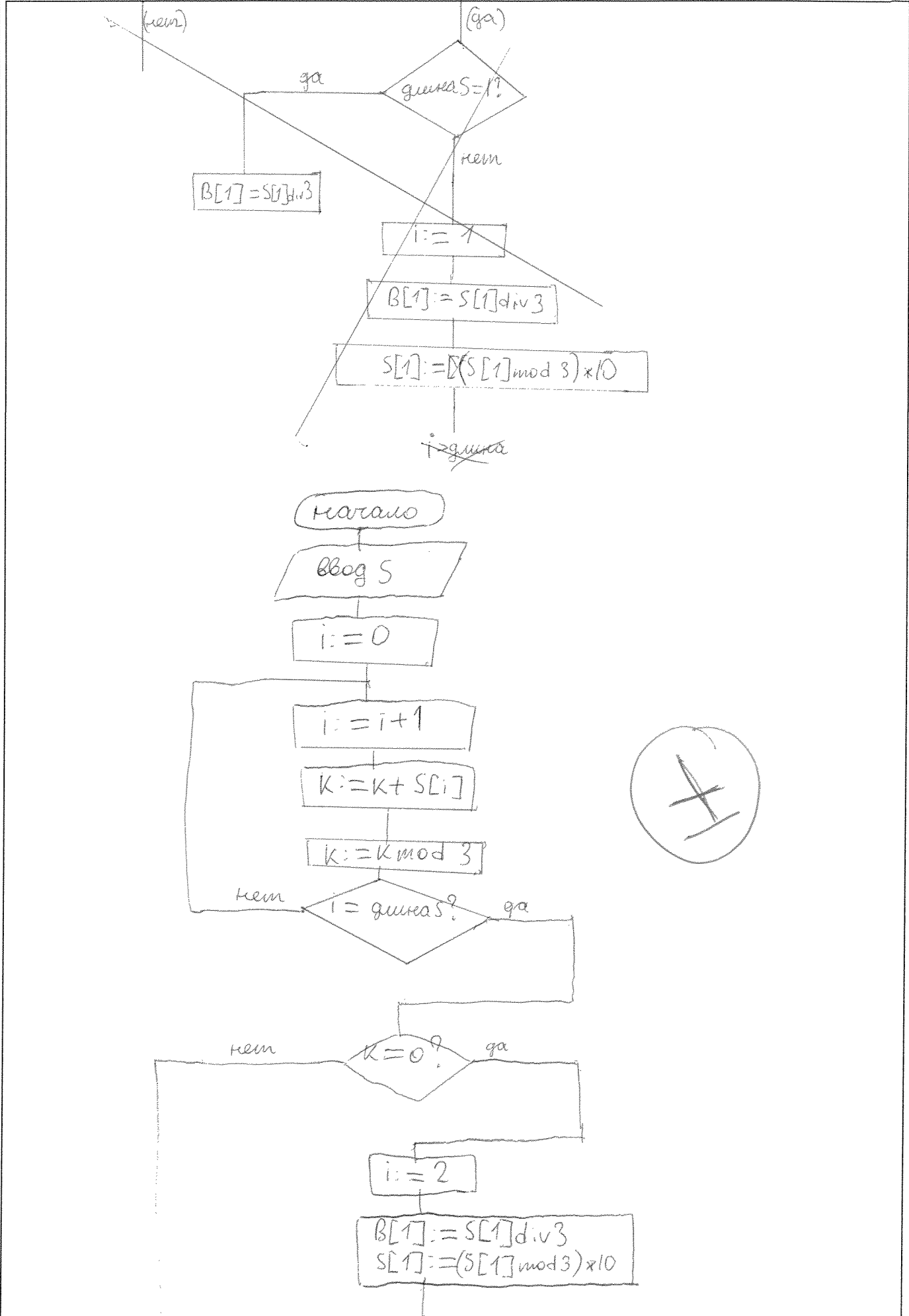
Целые:

$K$  — покажем, что, делится ли число  $S$  на 3. В самом начале алгоритма  $K=0$   
 $i$  — счетчик  
 $B$  — строка, содержащая число  $B$



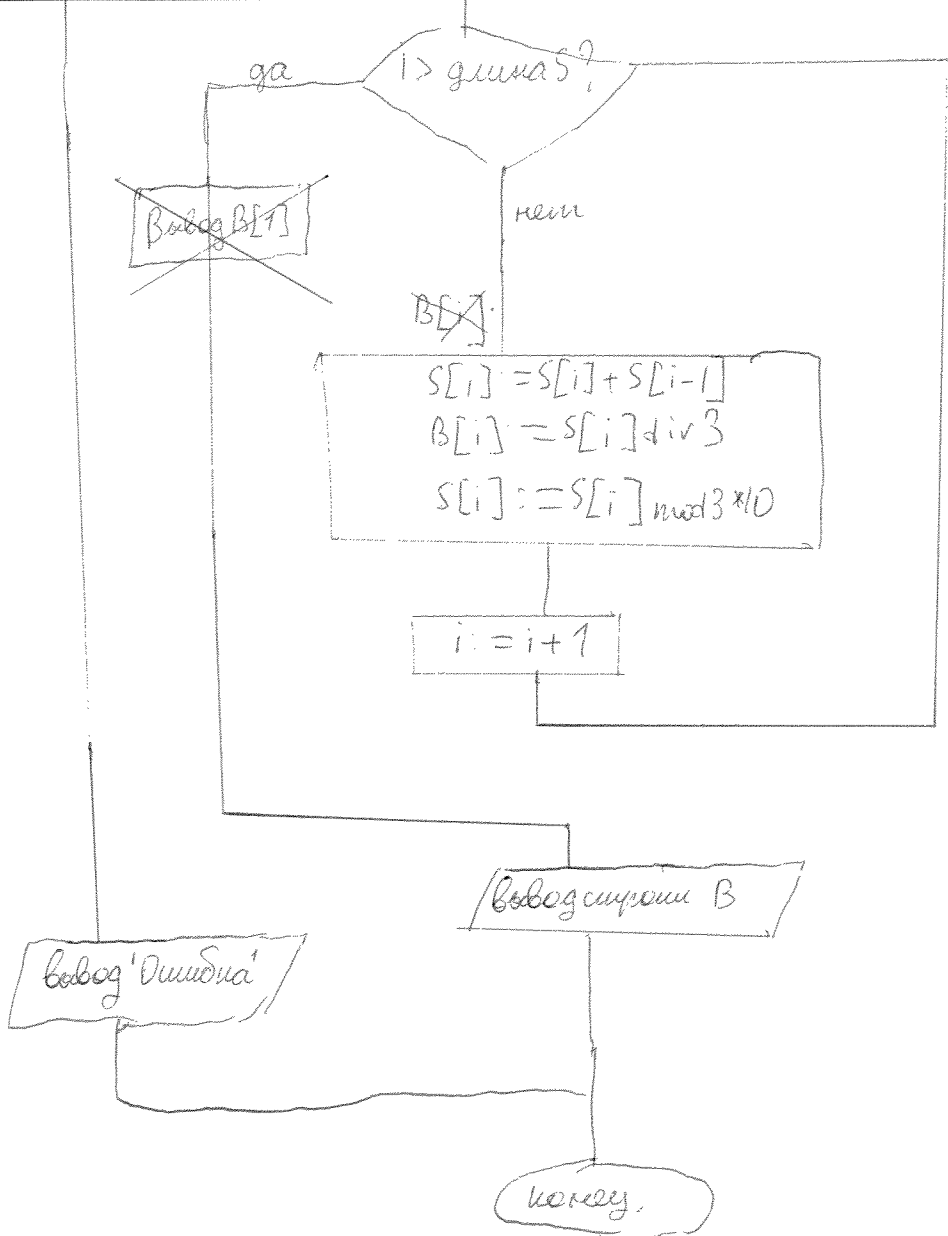


ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа



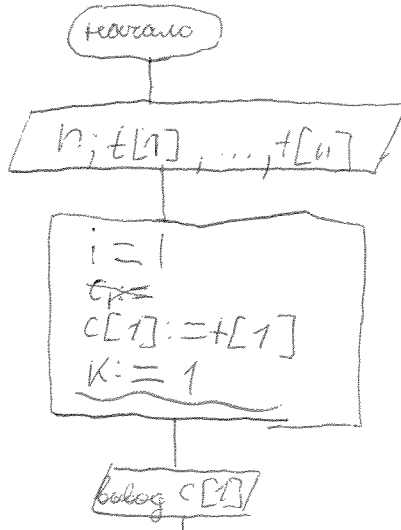


ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа



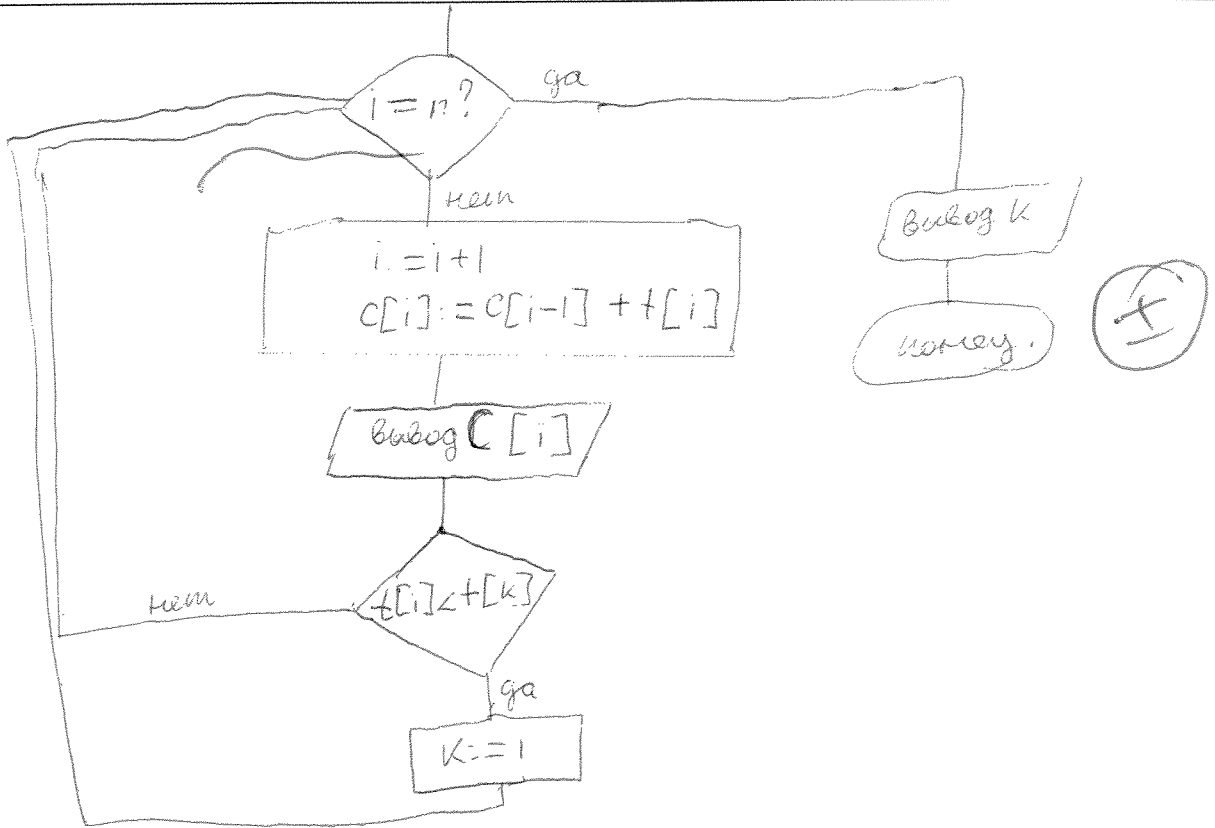
Задание 4

к номеру поочередно  
на обозначение  
каждого числа  
меньше всего  
времени.

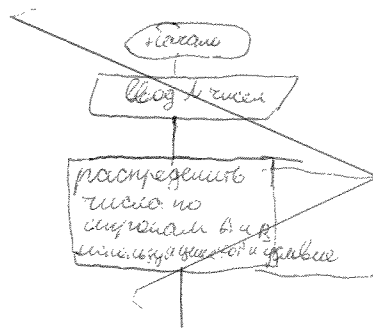




ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа



задание 2



A - строка с четными числами  
B - строка с нечетными числами

- 1) прочесть N чисел
- 2) для каждого из N чисел если оно четно, то записать в строку A, если нечетно в строку B
- 2) Уменьшая, число "узурья" упорядочить по возрастанию числа в строках A и B
- 3) Вывести сначала строку A, затем B

55-125



ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

№2

Максимальными будут числа краевые  $10^i$  и  $10^i - 1$  но в  $10^i$  краевые как в разрядах единицы длиннейших значений куба будут  $10^i - 1$  но числа, оканчиваясь на  $4, 5, 6, 9$  следовательно можно было вывести все числа:  $1$ , краевые  $10^i$  и  $10^i - 1$  оканчив. на  $4, 5, 6, 9$  и входящие в промежуток  $[U; V]$  п.о. можно задать какой-либо список и в каждой массе вывести те числа которые будут  $10^i - 1$  малы, что список от деления на  $10$  будет  $0, 1, 4, 5, 6, 9$  и вывести поучившей массив на экран, но данный алгоритм будет работать только для однозначных чисел, и если сделать для всех чисел, то можно задать массив размером  $10^{(V-U)}$  и вывести все  $10^i$  и в каждой массе вывести  $10^i - 1$  но можно упростить условие: если нам дано  $n$ -значное число то делением на  $10^k$  где  $k$  — количество цифр даем список равный  $10^k$  числу, при делении на  $10^n$

```
#include <iostream>
#include <math.h>
using namespace std;
int f(int x)
{ if (x < 10, ln, it, k; if (ln == 0;
  while (x > 9)
  { x /= 10;
    ln++; } ln++;
  b = POW(10, ln);
  k = POW(x, 3);
  it = k % b;
  if (it == x) cout << x; } }
int main()
{ int u, v, gas;
  gas = v - u;
  int arr[gas];
  for (int i = 0; i < gas; i++)
  { arr[i] = u;
    u++;
  }
  for (int i = 0; i < gas; i++)
  { f(arr[i]); } }
```





ВНИМАНИЕ! Проверяется только то, что записано с этой стороны листа в рамке справа

№4

Некоторые из условий, такие условия когда равенств вот так:



, т.к. 1 слово будет пересекать другое а 3-е будет ему параллельно, т.е. она может пересекаться только

горизонтально и вертикально. И заданная строка, состоящая из разных символов 1 слова, 2 или более ударов.

Для этого, сначала запишем слова по символу в каждый элемент массива ~~символов~~ и эти и массивы друг с другом

попарно сравниваем и если получили разные нормальные, возможные

№5

Сначала найдем целое ~~значение~~ в массиве ~~Майя~~

№5.

```

#include <iostream>
using namespace std;
int func(int x, int len)
{
    int y, t;
    t = x;
    for (int i = 0; i < len; i++)
    {
        y = x % 10;
        x /= 10;
        for (int j = 1; j <= y; j++)
            isc += j;
        if (isc == 0) cout << t << endl;
    }
}

int main()
{
    int u, v, gasm;
    cin >> u >> v;
    gasm = v - u;
    int arr [gasm];
    for (int i = 0; i < gasm; i++)
    {
        arr[i] = u; u++;
    }
    for (i = 0; i < gasm; i++)
    {
        func(arr[i]);
    }
}

```





