

Задача 1

Автомат получает на вход последовательность целых чисел и складывает их по следующим правилам:

- 1) Если число чётное, автомат удваивает его и добавляет в сумму.
- 2) Если число нечётное, автомат добавляет его значение в сумму.

После обработки последовательности автомат **вычитает** из получившейся суммы максимальное число последовательности, кратное 3, и выводит получившееся значение как результат.

Располагая последовательностью, определите, какой результат выведет автомат.

Входные данные

На вход программе в первой строке подается натуральное число N ($5 \leq N \leq 10000$) – количество чисел. Далее в N строках подаётся по одному натуральному числу, не превышающему 1000. Если чисел, кратных 3, в последовательности нет, автомат ничего не вычитает.

Выходные данные

Вывести одно целое число – наибольшее возможное, которое можно получить по правилам, описанным в условии задачи.

Пример

Ввод	Вывод
5 7 2 3 6 9	26

Решение

```
n = int(input())
answer = str(input())

for i in range(1,n):
    x = int(input())
    y = int(answer[0])
    if x>=y:
        answer = str(x)+answer
    else:
        answer = answer+str(x)

print(int(answer))
```

Задача 2

Галактическая станция «Галилей» имеет 2 ангара для космических кораблей.

Цикл предполетной подготовки длится M ($0 < M \leq 10$) суток. Цикл начинается в тот же момент, как корабль залетает в ангар. До окончания цикла корабль не может покинуть ангар и соответственно начать новый рейс.

Сутки на станции составляют 24 часа. Календарь на станции составляет 365 дней, которые не делятся на месяцы.

Вам попали в руки фрагменты станционного журнала ($0 < N \leq 1000$ строк). В нем фиксируются даты прибытия кораблей в зону станции, а также заявки на рейсы со станции. Гарантируется, что все страницы относятся к одному году.

Если корабль подлетает к станции, а все ангары заняты, то ему **отказывают** в обслуживании. Если вылет со станции назначен на тот же день, что и прилет нового корабля, будем считать, что сначала ангар освобождается, а потом новый корабль размещается в пустом ангаре (начало цикла будет считаться с момента прилета).

Если приходит запрос на **рейс со станции**, то в рейс уходит тот корабль, который готов к вылету. Если таких кораблей несколько, то выбирается тот, который дольше находится на станции. Если готовых к рейсу кораблей нет, то заявку на рейс **отклоняют**.

По данным журнала определите скольким кораблям было отказано и сколько заявок на рейсы было отклонено в рассматриваемый период.

Входные данные:

В первой строке через пробел два числа M N .

Дальше N строк из пар целых чисел, каждая на новой строке. Каждая пара — это дата и тип события (**1** - прилет корабля, **-1** - заявка на рейс со станции). Даты отсортированы по неубыванию.

Выходные данные:

Два целых числа, каждое на новой строке:

- количество отказов в обслуживании;
- количество отклоненных заявок.

Входные данные	Вывод	Примечание
4 10	1	В журнале 10 записей.
1 1	2	Корабли, прибывшие 1-го и 2-го, были поставлены в два ангара, имеющиеся на станции.
2 1		На момент прилета второго корабля на 2-ой день текущего года оба ангара были заняты. Этот корабль получили отказ в обслуживании.
2 1		Заявка на вылет на 4-ый день была отклонена, так как оба корабля находились на техобслуживании в ангарах.
4 -1		6-го числа 1-ый ангар покинул корабль по заявке. В тот же момент его место занял прилетевший в этот же день корабль.
6 -1		Заявки на вылет 7-го и 10-го дня были удовлетворены и ангары опустели.
6 1		Заявка на 12 день была отклонена, так как не было кораблей на станции.
7 -1		20-го числа прилетел корабль и занял первый ангар.
10 -1		
12 -1		
20 1		

Решение

```
m,n = map(int,input().split())
a1,a2 = 0, 0
no = 0
non=0
for i in range(n):
    t, p = list(map(int,input().split()))
    if p == 1:
        if a1 == 0:
            a1 = t + m
        elif a2 == 0:
            a2 = t + m
        else:
            no +=1
    elif p == -1:
        if a1 <= t and a1 != 0:
            a1 = 0
        elif a2 <= t and a2 != 0:
            a2 = 0
        else:
            non +=1

print (no)
print (non)
```

Задача 3

На уроке информатики Софья научилась записывать числа в системах счисления с различными основаниями.

Ей не очень понравилось, что в качестве цифр используются начальные символы строки "0123456789ABCDEFGH...XYZ".

Софья решила, что для цифр можно использовать любые слова нужной длины и с различными буквами.

Например, для восьмеричной системы счисления она выбрала слово "МИТРОФАН" (в этом слове 8 различных букв).

Софье понравилось, что некоторые числа теперь соответствуют словам, а некоторые слова можно закодировать числами.

Число 456_{10} превращается в слово "НИМ" ($456_{10}=710_8$ и заменяем 7 на "Н"; 1 на "И"; 0 на "М"), а слово "РОМАН" можно закодировать числом 14391 ("Р" - код цифры 4, "О" - 5, "М" - 0, "А" - 6, "Н" - 7 и $45067_8 = 4*8^4 + 5*8^3 + 0*8^2 + 6*8 + 7 = 14391$)

Софье понравилась эта игра в буквы и числа, и она решила исследовать данный способ перевода чисел (назовем этот способ перевода "строковым")

Помогите Софье найти ответ на следующую задачу.

Для отрезка натуральных чисел от А до В (включительно) определить количество натуральных чисел, которые после "строкового" перевода по alf будут иметь окончание равное строке hvost

Входные данные:

В первой строке - строка алфавита alf (различные символы из множества "0123456789A...Za...z")

Во второй строке - два натуральных числа А, В ($1 \leq A \leq B \leq 10^9$)

В третьей строке - строка hvost (все символы есть в alf, символы могут повторяться)

Выходные данные:

Одно целое число - ответ на задачу.

Примеры

Входные данные	Выходные данные	Пояснение
МЕТРО 10 100 РОМ	1	$10=20_5$; $100=400_5$ РОМ соответствует числу 340_5 , которое на отрезке $[20,400]$ встречается 1 раз
МЕТРО 1 1000 ТЕРЕМ	0	ТЕРЕМ соответствует числу $21310_5=1455$ - это числу больше 1000
МЕТРО 1 100 МММ	0	$100=400_5$ МММ соответствует набору 000, Минимальное число, которое оканчивается на 000 это 1000з
МЕТРО 4 100 Т	19	Т соответствует числу 2. Подойдут числа 7 (12_5), 12 (22_5), ..., 97 (342_5)
СТРЕКОЗА 1 987654 КОСТЕР	4	
СТРЕКОЗА 1 2345678 СССРССС	1	

Решение

```
def ppp(sss,alf): # получение числа в 10 СС
    # sss - слово для перевода, alf - алфавит
    x,p=0,len(alf) # число и основание СС
    for s in sss:
        i=alf.find(s) # индекс буквы
        x=x*p+ i # добавление "разряда" к числу
    return x # возврат числа
```

```
def solution (A,B,alf,hvost):
    #A,B -границы отрезка, hvost - искомые окончания, alf - алфавит
    p,h=len(alf),len(hvost) # основание и длина хвоста
    ph=p**h # размер блока
```

```

x=ppp(hvost,alf) # числовое выражение хвоста
A1=max(x,A)-1 # до A1 - решений нет
if hvost[0]== alf[0] : # случай с ведущими нулями
    A1=max(ph,A)-1 # до A1+1 - 1-й возможный кандидат в решение
a0,ka=A1%ph,A1//ph # вычисление префикса по A1 (сколько удалить ДО)
if a0>=x : ka+=1 # проверка начального блока
b0,kb=B%ph,B//ph # вычисление префикса по B1 (сколько ВСЕГО)
if b0>=x : kb+=1 # проверка финального блока
kr=max(0,kb-ka) # результат, учитываем, что B может быть < A1
return kr
alf=input()
A,B=map(int,input().split())
hvost=input()
kr=solution(A,B,alf,hvost)
print(kr)

```

Задача 4

Старшеклассник Миша собирает робота, который должен передвигаться по рельсам вокруг испытательного стенда. Всего робот умеет выполнять 13 различных команд, но для нас представляют интерес три из них – «вперёд», «назад», «стой». Миша решил передавать роботу инструкции в виде цифр числа: робот получает число, переводит его в систему счисления с основанием 13 и выполняет соответствующие цифрам команды. Коды команд, отвечающих за изменение скорости робота, кратны шести.

На вход подаётся N чисел с наборами команд. Сколько раз робот изменит скорость, если считать, что он не встречает препятствий?

Формат ввода

На вход программе в первой строке подается натуральное число N ($N \leq 10000$) – количество наборов команд. Далее в N строках на вход подаётся по одному целому числу в диапазоне от 0 до $4 \cdot 10^9$ – набор тринадцатеричных команд, записанных в десятичной системе счисления.

Формат вывода

Вывести одно целое число – сколько раз робот изменит скорость.

Пример

Ввод	Вывод
4 32 130 14 10	2

Решение

```
def count(x):  
    s = 0  
    while x>0:  
        if x%12 in [0,3,6,9]:  
            s+=1  
        x=x//12  
    return s  
  
n = int(input())  
answer = 0  
for i in range(n):  
    x = int(input())  
    answer+=count(x)  
  
print(answer)
```

Задача 5

Агрохолдинг «Дикое Поле» анализирует результаты сбора урожая. Известно, сколько тонн зерна убрали на каждом из N полей, находящихся в распоряжении холдинга. Так как несколько огромных полей сильно влияют на среднее, в агрохолдинге решили ввести другую метрику. Опорными называются поля, урожай с которых превышает пороговое значение, но меньше среднего. Определите наиболее часто встречающийся урожай с опорного поля.

Формат ввода

На вход программе в первой строке подаётся натуральное число N ($N \leq 1000$) – количество полей. Во второй строке подаётся натуральное число M ($M \leq 100$ т) – пороговое значение урожая с поля. Далее в N строках идёт по одному натуральному числу m_i – масса урожая с поля номер i ($1 \leq m_i \leq 1000$ т).

Формат вывода

Вывести одно целое число – наиболее часто встречающийся урожай с опорного поля. Если таких значений несколько, выведите наибольшее. Если таких значений нет, выведите 0.

Пример

Ввод	Вывод
7 20 10 20 32 43	32

32 175 150	
------------------	--

Решение

```
answer = 0
```

```
n = int(input())
```

```
data = [int(input()) for _ in range(n)]
```

```
food = [0]*1001
```

```
mid = sum(data)/n
```

```
for i in range(n):
```

```
    if (data[i]>mid) and (data[i]<=2*mid):
```

```
        food[data[i]]+=1
```

```
answer = food.index(max(food))
```

```
print(answer)
```

Задача 6

Электронная схема состоит из элементов **I** и **HE**.

Элемент **HE** имеет один вход и один выход. Принцип его работы следующий: если на входе появится сигнал **0**, то через 1 мс на выходе установится сигнал **1**, а если на входе **1**, то через 1 мс на выходе установится сигнал **0**.

Элемент **I** имеет два входа и один выход. Если на обоих его входах появится сигнал **1**, то через 1 мс на выходе установится сигнал **1**. Если хотя бы на один из входов поступает **0**, то через 1 мс на выходе устанавливается сигнал **0**.

Все точки подсоединения элементов пронумерованы. Если в точку поступает сигнал с выходов нескольких элементов, то в этой точке сигнал равен **0** тогда, когда со всех выходов поступает сигнал **0**. Если с одного или нескольких выходов, подсоединенных в одной точке, поступает сигнал **1**, то в этой точке устанавливается сигнал **1**. В последних двух случаях сигнал устанавливается мгновенно (без задержки).

Известно состояние (сигнал **0** или **1**) каждой точки в момент включения схемы. Необходимо выдать состояние некоторой указанной точки **K** в течение первых **T** мс с момента включения схемы.

В точках, которые соединены только с входами элементов, сигнал остается неизменным с момента включения схемы до окончания ее работы.

Формат ввода

На вход программе в первой строке подаётся натуральное число N. Далее идет N строк, каждая из которых содержит несколько целых десятичных чисел, отделенных друг от друга одним или несколькими пробелами. Первое число в строке показывает, что именно описывают оставшиеся числа данной строки:

0 - описание точки соединения;

0 m n - точка с номером **m** имеет в момент включения состояние **n** (**0** или **1**)

1 - описание элемента НЕ;

1 $x y$ - элемент НЕ, вход которого соединен с точкой под номером x , а выход - с точкой под номером y

2 - описание элемента И;

2 $x y z$ - элемент И, один вход которого соединен с точкой под номером x , второй вход соединен с точкой под номером y , а выход - с точкой под номером z

3 - описание задания.

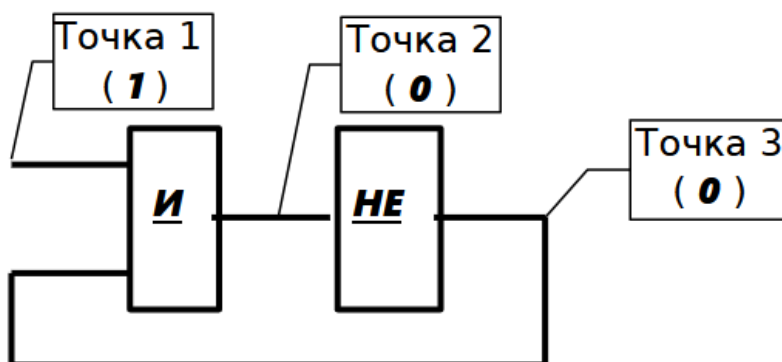
3 $K T$ - необходимо выдать состояние точки K в течение первых T мс с момента включения схемы.

Формат вывода

T строк: первая строка - состояние точки K в первую мс, вторая строка - состояние точки K во вторую мс, и так далее до T мс.

Пример

Пусть имеется схема, приведенная на рисунке. Необходимо выдать состояние точки 3 в течение 5 мс с момента включения схемы.



Исходные данные:	Пояснение
6	
0 1 1	точка 1, в момент включения сигнал 1
0 2 0	точка 2, в момент включения сигнал 0
0 3 0	точка 3, в момент включения сигнал 0
1 2 3	элемент <u>НЕ</u> , вход соединен с точкой 2, выход - 3
2 1 3 2	элемент <u>И</u> , один вход - точка 1, другой - 3, выход - 2
3 3 5	задание: выдать состояние точки 3 в течение 5 мс

Для приведенного примера результат будет следующий:

1
1
0
0
1

Примечание. Количество элементов **И** и **НЕ** конечно и составляет от 0 до 20 каждого из элементов. В задании количество мс не может превышать 50.

Решение

```
program scheme;
```

```
const
```

```
  np = 50;  
  nn = 50;  
  ni = 50;  
  nres = 80;
```

```
type
```

```
  point = record  
    num: byte;  
    vol: byte;  
    whod: boolean;
```

```
  end;
```

```
  tn = record  
    whod: byte;  
    wyh: byte;
```

```
  end;
```

```
  ti = record  
    wh1, wh2: byte;  
    wyh: byte;
```

```
  end;
```

```
  typ_p = array[1..np] of point;  
  typ_n = array[1..nn] of tn;  
  typ_i = array[1..ni] of ti;  
  typ_res = array[1..nres] of byte;
```

```
procedure init(var points: typ_p; var eltne: typ_n; var elti: typ_i;  
               var qp, qn, qi, wyh, time: byte);
```

```
var
```

```
  n : integer;  
  name: string;  
  ident: byte;
```

```
begin
```

```
  Readln(n);
```

```
  qp := 0;
```

```
  qn := 0;
```

```
  qi := 0;
```

```
  for var i:=1 to n do
```

```
  begin
```

```
    read(ident);
```

```
    case ident of
```

```
      0:
```

```
        begin
```

```
          inc(qp);
```

```
          readln(points[qp].num, points[qp].vol);
```

```
          points[qp].whod := true;
```

```
        end;
```

```

1:
  begin
    inc(qn);
    readln(eltne[qn].whod, eltne[qn].wyh);
    points[eltne[qn].wyh].whod := false;
  end;
2:
  begin
    inc(qi);
    readln(elti[qi].wh1, elti[qi].wh2, elti[qi].wyh);
    points[elti[qi].wyh].whod := false;
  end;
3:
  begin
    readln(wyh, time);
  end;
end
end;

end;

procedure step(var points: typ_p; var eltne: typ_n; var elti: typ_i;
               qp, qn, qi, wyh, t: byte; var res: typ_res);
var
  newpoints: typ_p;
  i: byte;
begin
  for i := 1 to qp do
    begin
      newpoints[i].whod := points[i].whod;
      if points[i].whod then
        newpoints[i].vol := points[i].vol
      else
        newpoints[i].vol := 0;
      end;
    for i := 1 to qn do
      begin
        newpoints[eltne[i].wyh].vol :=
          newpoints[eltne[i].wyh].vol or (1 - points[eltne[i].whod].vol)
      end;
    for i := 1 to qi do
      begin
        newpoints[elti[i].wyh].vol := newpoints[elti[i].wyh].vol or
          (points[elti[i].wh1].vol and points[elti[i].wh2].vol)
      end;
    res[t] := newpoints[wyh].vol;
    for i := 1 to qp do
      begin
        points[i].vol := newpoints[i].vol;
      end
    end;
  end;

procedure out(res: typ_res; time: byte);
var

```

```
t: byte;
begin

  for t := 1 to time do
    writeln(res[t]);

end;


var
  t, time, qp, qn, qi, wyh, i: byte;
  points: typ_p;
  eltne: typ_n;
  elti: typ_i;
  rslt: typ_res;
  stud: typ_res;

begin
  init(points, eltne, elti, qp, qn, qi, wyh, time);
  for t := 1 to time do
    step(points, eltne, elti, qp, qn, qi, wyh, t, rslt);
    out(rslt, time);

  end.
```