

## Problem A. Египетские дни рождения

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 second(s)  
Memory limit: 512 MiB

В недавней археологической экспедиции в Египте Василий Ильич и его коллеги нашли каменную плиту, на которой было много похожих записей. По ряду причин археологи решили, что это — списки рабов, задействованных на строительстве пирамид: их имена, дни рождения, родные города. Василий Ильич очень интересуется, как же надсмотрщики различали рабов, если у них совпадали все эти параметры. Для начала он хочет определить, сколько среди записей на плите различны. Помогите ему.

### Input

Первая строка содержит число  $n$  - количество записей на плите. Далее  $n$  строк, содержащих записи. Каждая строка состоит из маленьких латинских букв и не является пустой. Гарантируется, что суммарная длина строк не превосходит  $5 \cdot 10^6$ .

### Output

Выведите количество различных записей.

### Examples

standard input	standard output
4 a aa aab aa	3
3 a a a	1

## Problem B. К-я строка

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second(s)  
Memory limit: 512 MiB

Реализуйте структуру данных, которая поддерживает следующие операции:

- добавить в словарь строку  $S$ ;
- найти в словаре  $k$ -ю строку в лексикографическом порядке.

Известно, что изначально словарь пуст.

### Input

Первая строка входного файла содержит натуральное число  $N$  — количество команд ( $N \leq 10^5$ ). Последующие  $N$  строк содержат по одной команде каждая.

Команда записывается либо в виде числа  $k$ , либо в виде строки  $S$ , которая может состоять только из строчных латинских букв. Гарантируется, что при запросе  $k$ -й строки она существует. Также гарантируется, что сумма длин всех добавляемых строк не превышает  $10^5$ .

### Output

Для каждого числового запроса  $k$  выходной файл должен содержать  $k$ -ю в лексикографическом порядке строчку из словаря на момент запроса. Гарантируется, что суммарная длина строк в выходном файле не превышает  $10^5$ .

### Example

standard input	standard output
7 pushkin lermontov tolstoy gogol gorkiy 5 1	tolstoy gogol

## Problem C. Шифр Бэкона

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second(s)  
Memory limit: 512 MiB

Программисту Васе не повезло — вместо отпуска его послали в командировку на научную конференцию. «Надо повышать уровень знаний», — сказал начальник, «Важная конференция по криптографии, проводится во Франции — а там шифровали еще во времена Ришелье и взламывали чужие шифры еще во времена Виета.»

Вася быстро выяснил, что все луврские картины он уже где-то видел, вид Эйфелевой башни приелся ему еще раньше, чем мышка стерла его с коврика, а такие стеклянные пирамиды у нас делают надо всякими киосками и сомнительными забегаловками. Одним словом, смотреть в Париже оказалось просто не на что, рыбу половить негде, поэтому Васе пришлось посещать доклады на конференции.

Один из докладчиков, в очередной раз пытаясь разгадать шифры Бэкона, выдвинул гипотезу, что ключ к тайнам Бэкона можно подобрать, проанализировав все возможные подстроки произведений Бэкона. «Но их же слишком много!» — вслух удивился Вася. «Нет, не так уж и много!» — закричал докладчик, — «Подсчитайте, и вы сами убедитесь!»

Тем же вечером Вася нашел в интернете полное собрание сочинений Бэкона. Он написал программу, которая переработала тексты в одну длинную строку, выкинув из текстов все пробелы и знаки препинания. И вот теперь Вася весьма озадачен — а как же подсчитать количество различных подстрок этой строки?

### Input

На входе дана непустая строка, полученная Васей. Строка состоит только из строчных латинских символов. Ее длина не превосходит 2 000 символов.

### Output

Выведите количество различных подстрок этой строки.

### Examples

standard input	standard output
aaba	8

## Problem D. Байтландский словарь

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 second(s)  
Memory limit: 512 MiB

Байтланские учёные установили, что слова байтландского языка образуются путём конкатенации двух других (возможно, совпадающих) слов этого языка, при этом одно слово может иметь несколько вариантов происхождения. По заданному словарю языка требуется сказать, сколькими способами можно получить каждое из его слов.

### Input

В первой строке входного файла находится целое число  $N$  ( $1 \leq N \leq 10^5$ ) — количество слов в словаре. В следующих  $N$  строках записаны сами слова, состоящие из строчных латинских букв, их длина не превышает 50 символов. Все слова в словаре различны. Можно использовать одно слово дважды.

### Output

Выведите  $N$  строк, содержащих по одному целому числу. Число в  $i$ -й строке должно означать количество способов получить  $i$ -ое слово путем соединения каких-либо двух других слов из этого же словаря.

### Example

standard input	standard output
5	0
x	1
xx	2
xxx	3
xxxx	4
xxxxx	

## Problem E. Витя и странный урок

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 second(s)  
Memory limit: 512 MiB

Сегодня на уроке Витя изучал очень интересную функцию — *mex*. *Mex* набора чисел — минимальное неотрицательное число, не присутствующее в наборе чисел. Например,  $mex([4, 33, 0, 1, 1, 5]) = 2$ , а  $mex([1, 2, 3]) = 0$ .

Витя очень быстро разобрался со всеми задачами учителя, а сможете ли вы?

Даны массив, состоящий из  $n$  неотрицательных целых чисел, и  $m$  запросов. Каждый запрос характеризуется одним числом  $x$  и заключается в следующих последовательных шагах:

- Выполнить операцию побитового сложения по модулю 2 (*xor*) каждого элемента массива с числом  $x$ .
- Найти *mex* полученного массива.

Учтите, что после каждого запроса массив изменяется.

### Input

Первая строка содержит два целых положительных числа  $n$  и  $m$  ( $1 \leq n, m \leq 3 \cdot 10^5$ ), обозначающие количество элементов в массиве и количество запросов соответственно.

Следующая строка содержит  $n$  неотрицательных целых чисел  $a_i$  ( $0 \leq a_i \leq 3 \cdot 10^5$ ), представляющих элементы исходного массива.

Каждая из следующих  $m$  строк содержит запрос — одно неотрицательное целое число  $x$  ( $0 \leq x \leq 3 \cdot 10^5$ ).

### Output

Для каждого запроса выведите ответ на него в отдельной строке.

### Examples

standard input	standard output
2 2 1 3 1 3	1 0
4 3 0 1 5 6 1 2 4	2 0 0

## Problem F. Костюмы для актёров

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 second(s)  
Memory limit: 512 MiB

Актёрская труппа одно очень известного театра готовится к премьере. Они уже давно придумали сценарий, распределили роли, каждый актер уже даже выучил свой текст! Осталось разобраться только с костюмами.

Выбор подходящих костюмов - вовсе не простая задача, как может показаться. Каждый актер порой вынужден переодеваться прямо во время спектакля! Конечно, это требует времени, так что иногда успеть вовремя выйти на сцену в новом костюме гораздо труднее, чем правильно сыграть свою роль.

К счастью, в премьерной постановке каждый костюм будет состоять из не более чем 30 занумерованных от 0 до 29 частей, каждая из которых покрашена либо в белый, либо в черный цвет. Актеры заметили, что чем сильнее отличается новый костюм от уже надетого, тем дольше придется переодеваться. Чтобы как-то оценить, какие костюмы труднее всего надевать, а какие легче всего, они решили сопоставить каждому костюму свое число по следующему принципу: пусть части костюма с номерами  $y_1, y_2, \dots, y_k$  имеют черный цвет, тогда данному костюму соответствует число  $2^{y_1} + 2^{y_2} + \dots + 2^{y_k}$ .

Введя таким образом соответствие между числами и костюмами, актеры заметили, что время переодевания одного костюма в другой равно битовому исключающему ИЛИ (т.е. хог-у) соответствующих им чисел. Например, если один костюм соответствует числу 6, а другой числу 5, то поменять один из них на другой актер может за  $6 \oplus 5 = 3$  секунды (так как  $6 = 2^2 + 2^1$ , а  $5 = 2^2 + 2^0$ , то переодеть потребуется части с номерами 0 и 1, что займет  $2^0 + 2^1 = 3$  секунды); в случае же костюмов с номерами 15 и 21 на переодевание потребуется  $2^4 + 2^3 + 2^1 = 24$  секунд (ибо  $15 = 2^3 + 2^2 + 2^1 + 2^0$ ,  $21 = 2^4 + 2^2 + 2^0$ ).

Теперь им интересно узнать для каждого имеющегося костюма: если актер наденет этот костюм, то какое наименьшее и наибольшее время на переодевание он может потратить? Так как вся труппа очень занята репетициями, они попросили вас помочь им решить эту задачу.

### Input

В первой строке находится единственное целое число  $n$  ( $2 \leq n \leq 2 * 10^5$ ) — количество костюмов, имеющихся в театре.

Во второй строке через пробел перечислены  $n$  чисел, соответствующие имеющимся костюмам:  $x_1, x_2, \dots, x_n$  ( $0 \leq x_i < 2^{30}$ ).

### Output

Выведите  $n$  строк,  $i$ -я из которых должна содержать два числа, записанных через пробел: минимальное и максимальное время, необходимое для переодевания  $i$ -го костюма.

### Examples

standard input	standard output
2	18 18
58 40	18 18
3	0 15
8 7 8	15 15
	0 15

## Problem A. Словарь

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 second(s)  
Memory limit: 512 MiB

Дан набор слов и текст, требуется определить для каждого слова, присутствует ли оно в тексте как подстрока.

### Input

В первой строке дан текст (не более  $10^6$  строчных латинских букв). Далее дано число  $M$  — количество слов в словаре.

В следующих  $M$  строках записаны слова (не более 30 строчных латинских букв). Слова различны и отсортированы в лексикографическом порядке.

Суммарная длина слов в словаре не более  $10^5$ .

### Output

$M$  строк вида **Yes**, если слово присутствует, и **No** иначе.

### Example

standard input	standard output
trololo	No
3	Yes
abacabadabacaba	Yes
olo	
trol	

## Problem B. Вирусы

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second(s)  
Memory limit: 512 MiB

Комитет По Исследованию Бинарных Вирусов обнаружил, что некоторые последовательности единиц и нулей являются кодами вирусов. Комитет изолировал набор кодов вирусов. Последовательность из единиц и нулей называется безопасной, если никакой ее сегмент (т.е. последовательность из соседних элементов) не является кодом вируса. Сейчас цель комитета состоит в том, чтобы установить, существует ли бесконечная безопасная последовательность из единиц и нулей.

### Input

Первая строка входного файла `virus.in` содержит одно целое число  $N$ , равное количеству всех вирусных кодов. Каждая из следующих  $n$  строк содержит непустое слово, составленное из символов 0 и 1 — код вируса. Суммарная длина всех слов не превосходит 30 000.

### Output

Первая и единственная строка выходного файла должна содержать слово:

- TAK — если бесконечная, безопасная последовательность из нулей и единиц существует;
- NIE — в противном случае.

### Example

standard input	standard output
3 01 11 00000	NIE
3 011 11 0000	TAK



## Problem C. Под-бор

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second(s)  
Memory limit: 512 MiB

*Бором* называется подвешенное дерево, на каждом из рёбер которого написано по символу, причём символы, написанные на рёбрах, выходящих из общей вершины-родителя, различны. Будем называть направление от родителя к детям “вниз”. Назовем *вхождением строки*  $s$  в *бор* такую вершину бора, от которой можно пройти несколько шагов вниз таким образом, что встретившиеся символы образуют строку  $s$ .

Даны бор и несколько строк, найдите сумму количеств вхождений этих строк в этот бор.

### Input

В первой строке входного файла записано единственное число  $n$ ,  $1 \leq n \leq 100\,000$  — количество вершин бора. В следующих  $n$  строках описаны вершины бора. В  $(i + 1)$ -й строке описаны дети  $i$ -й вершины: число  $k_i$  ее детей, затем  $k_i$  пар из номера вершины-ребёнка и символа, написанного на соответствующем ребре. Номер родителя всегда меньше номера ребёнка; корнем бора является вершина номер 1.

В  $(n + 2)$ -й строке записано количество  $m$  ( $1 \leq m \leq 100\,000$ ) строк для поиска. В следующих  $m$  строках перечислены сами строки. Входные строки непусты, а их суммарная длина не превышает 100 000 символов.

Все символы, написанные на рёбрах, а также все символы, составляющие строки — маленькие латинские буквы.

### Output

Выведите одно число — сумму количеств вхождений.

### Example

standard input	standard output
7 2 2 a 4 b 2 3 a 6 b 0 1 5 b 1 7 b 0 0 4 b bb bbb bb	9

## Problem D. Жужжащий профессор

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second(s)  
Memory limit: 512 MiB

В одном очень известном университете один очень известный профессор очень быстро произносил свои лекции, так, что ничего невозможно было разобрать. Студенты шутили по этому поводу, что он не говорит, а жужжит. Естественно, что про загадочного профессора никто абсолютно ничего не знал.

Но вот недавно Петя Булочкин решил предпринять исследование по изучению словарного запаса профессора. С этой целью он даже посетил одну лекцию и записал все сказанное на ней на диктофон. Затем, прокручивая дома запись с десятикратным замедлением, Петя смог записать все, что сказал профессор. Но вот незадача — профессор говорил так быстро, что, даже прослушивая замедленную запись, нельзя было точно сказать, где он делал паузы между словами. Таким образом, у Пети есть некоторый текст  $S$ , состоящий только из маленьких латинских букв — лекция, которая была прочитана профессором.

Петя решил, что те слова, которые профессор употреблял только один раз во время своей лекции, его не интересуют. Кроме того, понятно, что если профессор употреблял некоторое слово два или более раз, то существуют два неперекрывающихся вхождения этого слова в текст  $S$ . Назовем непустую строку  $T$  кандидатом в слова, если существуют два неперекрывающихся вхождения  $T$  в  $S$ . Теперь Петя хочет найти все строки, которые являются кандидатами в слова. И поможете ему в этом Вы.

### Input

Единственная строка входного файла содержит от 1 до 3 000 маленьких латинских букв. Это и есть текст  $S$ , который прочитал профессор на лекции.

### Output

Единственная строка выходного файла должна содержать одно число, равное количеству строк, являющихся кандидатами в слова.

### Example

standard input	standard output
bbaabbbabb	7

## Problem E. Remarkable Substrings

Input file: *standard input*  
Output file: *standard output*  
Time limit: 3 second(s)  
Memory limit: 512 MiB

Пусть  $S$  - строка, состоящая из строчных латинских букв. Непустая подстрока  $T$  строки  $S$  называется *замечательной* строкой степени  $k$  относительно  $S$ , если строка  $k \cdot T = T + T + \dots + T$  (т.е.  $T$ , записанная  $k$  раз) есть подстрока  $S$ .

Более формально,  $S = U + k \cdot T + V$  где  $U$  и  $V$  суть некоторые (возможно, пустые) строки.

Для данной строки  $S$ , найдите максимально возможный ранг  $x$ , т.ч. существует строка  $T$ , являющаяся замечательной строкой ранга  $x$  относительно  $S$ .

### Input

Единственная строка - строка  $S$  ( $1 \leq |S| \leq 10^6$ ).

Гарантируется, что  $S$  состоит исключительно из строчных латинских букв.

### Output

Выведите одно число - ответ на задачу.

### Examples

standard input	standard output
aaabc	3
abacacacacaba	4

## Problem A. Выпуклый многоугольник

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second(s)  
Memory limit: 512 MiB

Многоугольник – замкнутая ломаная.

*Простым* называется многоугольник без самопересечений, самокасаний.

*Выпуклым* называется многоугольник, ограничивающий выпуклую область.

*Выпуклой* называется область  $A$ , обладающая свойством  $P, Q \in A \Rightarrow [P, Q] \subseteq A$ .

### Input

В первой строке одно число  $N$  ( $3 \leq N \leq 100\,000$ ). Далее в  $N$  строках по паре целых чисел — координаты очередной вершины простого многоугольника в порядке обхода по или против часовой стрелки.

### Output

Одна строка «YES», если приведённый многоугольник является выпуклым, и «NO» в противном случае.

### Examples

standard input	standard output
3 0 0 0 1 1 0	YES

## Problem B. Выпуклая оболочка

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 512 MiB

Дано  $N$  точек на плоскости. Нужно построить их выпуклую оболочку. Гарантируется, что выпуклая оболочка является невырожденной.

### Input

В первой строке число  $N$  ( $3 \leq N \leq 10^5$ ). Следующие  $N$  строк содержат пары целых чисел  $x$  и  $y$  ( $-10^9 \leq x, y \leq 10^9$ ) – координаты точек.

### Output

В первой строке выведите  $N$  – число вершин выпуклой оболочки. Следующие  $N$  строк должны содержать координаты вершин в порядке обхода. Никакие три подряд идущие точки не должны лежать на одной прямой.

### Example

standard input	standard output
3	3
0 0	0 0
1 0	1 1
1 1	1 0

## Problem C. Селфи на плоскости

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 second(s)  
Memory limit: 512 MiB

Для парада параллелей несколько школьников решило сделать групповое селфи. Если вдруг кто-то не знает, селфи — это такой модный вид фотографии, когда человек сам снимает себя и своих товарищей, как правило, на мобильный телефон. Школьники уже выбрали места, где будут стоять во время съёмки, и теперь пытаются определить, чьим телефоном будет сделана фотография.

В рамках данной задачи будем считать каждого школьника точкой на плоскости. Ребята решили, что фотографию стоит делать из точки, в которой находится какой-то школьник, причём угол обзора, в который помещаются все остальные точки, соответствующие школьникам, должен быть как можно меньше. Помогите им определить минимальный угол обзора на фотографии, вмещающей всех людей.

Более формально, ответ на задачу — такое минимальное неотрицательное вещественное число градусов  $\alpha$ , что существует некоторый человек и угол величиной в  $\alpha$  градусов с вершиной в позиции этого человека, содержащий всех людей (при этом человек может находиться на границе угла).

Длиной палки для селфи можно пренебречь.

### Input

В первой строке ввода задано единственное число  $n$  ( $1 \leq n \leq 200\,000$ ) — количество людей.

Далее следуют  $n$  строк, каждая из которых содержит по два числа  $x_i$  и  $y_i$  ( $-10^9 \leq x_i, y_i \leq 10^9$ ) — координаты позиции  $i$ -го человека. Некоторым людям может нравиться одно и то же место, поэтому позиции некоторых людей могут совпадать.

### Output

Выведите единственное вещественное число  $\alpha$  — минимальный угол обзора в градусах.

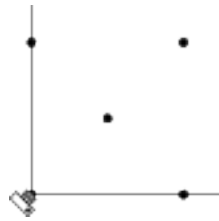
Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность относительно ответа жюри не превосходит  $10^{-6}$ . Более формально, если ваш ответ —  $ans$ , а ответ жюри —  $jury$ , ваш ответ будет признан правильным, если  $\frac{|ans - jury|}{\max(1, jury)} \leq 10^{-6}$ .

### Examples

standard input	standard output
5 1 1 3 1 2 2 1 3 3 3	90.0000000000000000
3 1 0 2 1 3 2	0.0000000000000000

### Note

Иллюстрация для первого примера:



## Problem D. Жадный король

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second(s)  
Memory limit: 512 MiB

Жил-был жадный король, который однажды приказал главному архитектору окружить королевский замок стеной. Король был настолько жаден, что не желал слушать рассказы архитектора о красивой кирпичной стене с прекрасным силуэтом и изящными высокими башнями. Вместо этого он приказал окружить замок стеной, затратив минимальное количество камня и времени, но потребовал, чтобы стена не подходила к замку ближе, чем на заданное расстояние. Если король узнает, что архитектор потратил не минимально возможное количество ресурсов, то архитектор лишится головы. Более того, король потребовал, чтобы архитектор сразу же предложил проект стены с указанием минимального количества ресурсов, необходимых для постройки.

Вы должны помочь архитектору сохранить голову, написав программу для поиска минимальной длины стены, удовлетворяющей условиям короля.

Задачу упрощает то, что замок короля имеет форму многоугольника и расположен на равнине. Архитектор уже ввел систему координат и точно измерил координаты вершин замка в футах.

### Input

Первая строка входного файла содержит числа  $N$  и  $L$ , разделенные пробелом.  $N$  ( $3 \leq N \leq 1000$ ) — это количество вершин в королевском замке, а  $L$  ( $1 \leq L \leq 1000$ ) — минимальное количество футов, на которое стена может приближаться к замку.

Следующие  $N$  строк описывают координаты замка в порядке обхода по часовой стрелке. В каждой строке через пробел записаны целые числа  $x_i$  и  $y_i$ , разделенные пробелом ( $-10000 \leq x_i, y_i \leq 10000$ ), которые обозначают координаты  $i$ -ой вершины. Все вершины различны, и никакие две стороны не пересекаются кроме как по вершинам.

### Output

Выведите минимальную длину стены в футах, удовлетворяющей условиям короля с точностью не менее 6 знаков после запятой.

### Examples

standard input	standard output
9 100 200 400 300 400 300 300 400 300 400 400 500 400 500 200 350 200 200 200	1628.3185307180



## Problem E. Внутри или вовне?

Input file: *standard input*  
Output file: *standard output*  
Time limit: 3 second(s)  
Memory limit: 512 MiB

Несколько выпуклых многоугольников вложены друг в друга так, что второй многоугольник помещён в первый, третий — во второй и так далее. Каждый следующий многоугольник лежит строго внутри предыдущего (то есть контуры многоугольников не имеют общих точек). После этого стороны многоугольников стираются и остаётся только набор вершин.

Ваша задача — восстановить всю конструкцию, то есть для каждой вершины сообщить номер многоугольника, которому она принадлежит. Отметим, что точка и отрезок являются выпуклыми многоугольниками с 1 и 2 сторонами соответственно.

### Input

Первая строка входа содержит одно целое число  $N$  ( $1 \leq N \leq 2 \cdot 10^4$ ) — количество вершин. В  $i$ -й из последующих  $N$  строк заданы два целых числа  $x_i$  и  $y_i$  — координаты  $i$ -й точки. Координаты не превосходят  $10^4$  по абсолютной величине.

### Output

Выведите  $N$  строк, по одной на каждую вершину. Каждая строка должна содержать одно целое число — номер выпуклого многоугольника для данной вершины (многоугольники занумерованы так, что внешний имеет номер 1, лежащий непосредственно в нём — номер 2 и так далее).

### Example

standard input	standard output
5	1
0 0	1
4 4	1
0 4	2
1 1	1
4 0	

## Problem F. Money, Money, Money!

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second(s)  
Memory limit: 512 MiB

Для финансового обеспечения подготовки к Чемпионату Европы Оргкомитет открыл в Аргентинско-Ямайском банке банковский валютный счёт, номинированный в евро.

В каждый из последующих  $N$  дней Оргкомитет производит некоторые операции с евро: или получает средства от спонсоров чемпионата — тогда на валютный счёт приходит положительная сумма, или рассчитывается с субподрядчиками — тогда с валютного счёта снимается некоторая сумма (то есть «приходит отрицательная сумма»).

В конце каждого дня банк разрешает перевести всю сумму евро, находящуюся на валютном счёту, на основной счёт Оргкомитета, номинированный в гривнах, по установленному условиям вклада курсу. Обратная операция условиями вклада не разрешена.

Курс обмена каждый день меняется по следующему правилу: на  $K$ -ый день один евро может быть обменян на  $K$  гривен. За каждый обмен банк забирает  $T$  гривен комиссионных. Таким образом, если в конце  $K$ -го дня на счёту Оргкомитета есть  $S$  евро, и принимается решение обменять их на гривны, на основной счёт приходят  $SK - T$  гривен (конечно,  $S$  может быть и отрицательным). В конце  $N$ -го дня, по завершении времени действия валютного вклада, банк принудительно конвертирует сумму (или долг) в гривны на основном счёту Оргкомитета и закрывает валютный счёт.

Оргкомитет Чемпионата Европы поручил Вам максимизировать суммарное количество гривен, которое будет переведено на основной счёт Оргкомитета за время действия валютного вклада.

### Input

В первой строке входного файла содержится количество тестовых случаев  $Q \leq 20$ . Каждый тестовый случай содержит два целых числа,  $N$  и  $T$ , разделённых одним или несколькими пробелами ( $1 \leq N \leq 34567$ ,  $0 \leq T \leq 255$ ). Следующая строка содержит  $N$  целых чисел  $-1000 \leq A_i \leq 1000$ , разделённых пробелами, обозначающих количество евро, поступающих (в случае, изменение вклада на валютном счёте в каждый из  $N$  дней соответственно).

### Output

Для каждого тестового случая выведите в отдельную строку выходного файла максимально возможное количество гривен, которое можно получить при заданных условиях на валютный счёт Оргкомитета за всё время действия валютного вклада.

### Example

standard input	standard output
1 7 1 -10 3 -2 4 -6 2 3	17