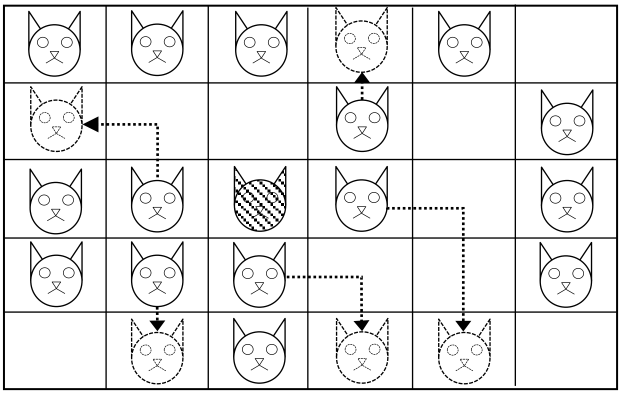
XXIX НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА
Национален кръг
Разград, 26-28 април, 2013 г.
Група А и B, 9-12 клас

ЗАДАЧА AB1. СУЕВЕРНИ ЧЕРНИ КОТКИ

Върху правоъгълна поляна са полегнали няколко суеверни черни котки. За съжаление, автоматичната система за напояване се включва, и котките трябва скоропостижно да напуснат поляната като се придвижат до някой от краищата ѝ. Тъй като са суеверни, обаче, никоя котка не може да пресече пътя на друга черна котка – т.е. да стъпи върху място, където е пребивавала друга котка.

Поляната е разграфена на $N \times M$ квадратни клетки, във всяка от които в началото има не повече от една котка. Котките могат да се придвижват от клетката, в която се намират, до някоя от четирите ѝ съседни клетки с обща страна, но само ако там вече не е стъпвала друга котка. Котка, придвижила се до крайна клетка, се брои за успешно напуснала поляната.

Напишете програма **`cats`**, която намира максималния брой котки, които могат успешно да напуснат поляната при дадените ограничения.



Фигура 1. Примерно разположение на 16 котки върху поляна с размери 5×6 (N=5, M=6). Всички котки, освен една (напр. защрихованата в средата) могат да се измъкнат, например както е показано.

Вход

От първи ред от стандартния вход се въвежда числото N. Следват N реда, описващи началното състояние на поляната: всеки ред съдържа точно M символа 1 или 0, в зависимост от това дали в съответната клетка има (1) или няма (0) котка.

Изход

Максималният брой котки, които могат успешно да напуснат поляната.

Ограничения

$1 \le N, M \le 30$

Пример 1

Вход:
```
3
0001
0110
1010
```
Изход:
```
5
```

Пример 2

Вход:
```
5
111010
000101
111101
111001
001000
```
Изход:
```
15
```

XXIX НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА
Национален кръг
Разград, 26-28 април, 2013 г.
Група A и B, 9-12 клас

ЗАДАЧА AB2. БАНКОВ КОД

Дадена е редица от цифри. След като, евентуално, премахнем някои от тях, остават цифри, които, написани една до друга в същия ред, образуват число. Например, ако е дадена редицата 21037, може по този начин да получим числата 21037, 3, 21, 27, 137 и много други, но никога числа като 31, 70, 8 и пр.

В една банка номерата на клиентите (цели числа, по-големи от нула) се закодират, използвайки такава непразна редица от цифри, че номерът да е **най-малкото** цяло положително число, което **не може** да се образува чрез евентуално премахване на цифри от редицата по описания начин. Първата цифра в тази редица е различна от 0. За номер 4 например, възможни кодиращи редици (ще ги наречем просто „кодове“) са 21076899958863, 1032, 12356789 и още много. Наистина, 1032 е „код“, съгласно определението: 1 се получава като 1~~032~~, 2 се получава като ~~103~~2, 3 се получава от ~~10~~3~~2~~, а 4 е първото цяло положително, което не се получава чрез описаната операция. Убедете се, че и другите изброени набори от цифри са кодове за 4.

В банката забелязали, че често броят на цифрите на кодовете, а и самият код като число, могат да се намалят, без да се промени номерът при разкодирането. Т.е., ако номерът на клиента е 4, кодът за него, който е най-малко число, е 123. Ще наречем такъв код „минимален“. Така, за номер 27, минималният код е 172012345689.

Напишете програма `kod`, която:

- по даден първоначален код на клиента извежда номера му;
- извежда минималния код за намерения номер на клиент.

Вход

От първия ред се въвежда кодът на клиента, съдържащ не повече от 2000 цифри, като първата е различна от нула.

Изход

На първия ред се извежда номерът на клиента, съответстващ на кода от входа. На следващия ред се извежда исканият в условието минимален код (най-малкото число във възможно най-малко цифри, което е код на клиента, чийто номер сте намерили).

Пример 1

Вход
112039814112723456789

Изход
30
120123456789

Пример 2

Вход
302123259873240569000156789017811122323456789

Изход
420
12323401567890123456789

Оценяване

В 50% от тестовете закодираният номер на клиента е число до 1010.

Един тест се оценява с 4 точки – 1 точка за намиране на номера и 3 точки за намиране на минималния код.

XXIX НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА
Национален кръг
Разград, 26-28 април, 2013 г.
Група А и В, 9-12 клас

ЗАДАЧА АВ3. ДОИЗРАВНЯВАНЕ

След като Хакерът Пешо си построи къща на една безлюдна крайбрежна ивица, той реши да оправи пътеката до входната врата. Съществуващата пътека от паркинга до входната врата е дълга точно N метра, като през цялото време или се изкачва нагоре, или е равна. С равните участъци няма проблем, но Пешо никак не обича да се изкачва (или да слиза по обратния път). Ивицата пред къщата му е разделена на отсечки с дължина един метър. Всяка отсечка има надморска височина – цяло неотрицателно число. Пешо е наел свръхмодерна техника, с която може да извършва следните три операции: да увеличи или намали надморската височина на една отсечка с единица, или да построи асансьор между две съседни отсечки. Цената на увеличаването и намаляването на височината е единица, а на строежа на асансьор – K. Пешо може да върви между две съседни отсечки, само ако надморската им височина е еднаква или има асансьор между тях. Цената на строежа на асансьора не зависи от височините на отсечките, които свързва. Възможно е многократното прилагане на операциите върху една и съща отсечка.

Напишете програма **`reflatten`**, която намира на каква минимална цена Пешо може да построи желаната пътека, която да може да извърви като ползва асансьорите, но без да се изкачва и слиза пеш.

Бележка: Увеличаването на надморската височина може да започне още от първата отсечка. Ако тя се издигне над паркинга, то собствениците на паркинга ще построят асансьор от паркинга до нея, без Пешо да харчи пари за това.

Вход

На първия ред на стандартния вход са зададени естествените числа N и K. На втория ред са зададени N неотрицателни цели числа в **ненамаляващ ред**, които са надморските височини на отсечките на съществуващата пътека.

Изход

На първия ред на стандартния изход програмата трябва да изведе минималната цена, за която Пешо може да построи желаната пътека.

Ограничения

$3 \le K \le 1\ 000\ 000\ 000$, а всяка отсечка от крайбрежната ивица има надморска височина не по-голяма от 1 000 000 000.

Пример

Вход	Изход
`5 3` `1 2 8 9 9`	5

Оценяване

Подзадача 1 (10 точки)

$N \le 200$

Подзадача 2 (20 точки)

$N \le 100000$. Оптималното решение може да се достигне без поставяне на асансьори.

Подзадача 3 (30 точки)

$N \leq 6000$

Подзадача 4 (40 точки)

$N \leq 100000$

Точките за всяка подзадача се получават само ако програмата премине успешно ***всички*** *тестови примери, предвидени за нея.*