



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 20 април 2013 г.

А. МНОГО ОБУВКИ

Планинска хижа предлага на посетителите услугата „ски екипировка“. Хижата предлага обувки с номера от 20 до 49, като от всеки номер, във всеки момент, има свободни различен брой обувки. Много голяма група туристи пристига на хижата и иска да наеме ски екипи. Всеки новодошъл скиор ще се съгласи да обуе само обувки, номерът на които да не е по-малък от номера, който той обикновено носи. Напишете програма, която определя максималния брой скиори, които ще могат да се пързаят този ден.

На първия ред на стандартния вход е зададен броят на тестовете. Всеки тест започва с ред на който са зададени броят N на пристигналите скиори и броят M на наличните в момента обувки ($N, M < 1000000$). На вторият ред са зададени, разделени с по един интервал, номерата на обувки, които пристигналите скиори носят, а на третия – номерата на наличните обувки.

На единствения ред на стандартния изход програмата трябва да изведе намерения максимален брой скиори.

ПРИМЕР

Вход

```
2
3 5
41 41 41
43 40 25 39 42
3 5
44 44 44
43 40 25 39 42
```

Изход

```
2
0
```



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 20 април 2013 г.

B. Bridges for Silvi

След дългия и тежък семестър (накрая на който всички гадни преподаватели слагат всички контролни и домашни), Силви почувства нужда от една добре заслужена почивка. За своята почивка тя реши да си направи екскурзия до Бахамските острови. Речено–сторено – Силви си взе самолетен билет и долетя до един от островите. Там обаче я чакаше неприятна изненада! Когато Силви си взе карта на островите от една сергия на плажа, установи, че реално никой не си е направил труда да се погрижи за удобството на туристите – няма построен нито един мост между различните острови – пълно безобразие! Че как се очаква човек да си направи sightseeing-а така?! Силви веднага отишла при местните власти и настояла това безобразие да бъде поправено! Властите се съгласиха веднага (все пак доста от управляващите имаха същия проблем като Силви в моментите, когато трябваше да идат до някой от останалите острови). Оказа се, че единствената причина да не са свършили работата досега сами беше, че те бяха изключително практични хора и не искаха да правят неоптимални работи – те искаха да могат да стигнат от главния остров (където са те) до всеки друг остров за минимално време, а не знаеха как да построят мостовете, така че това да може да се получи. Затова възложиха задачата на Силви – да измисли план, по който да построи мостове между някои от островите, така че времето за стигане от главния остров до който и да е друг, да е минимално. Силви реши да приеме задачата (тъй като все пак беше свършила парите и нямаше да може да си позволи транспорта с кораб между островите, а искаше да ги види всичките).

Картата на Силви беше примитивна и не съдържаше подробна информация, но това нямаше да я спре. Картата представляваше матрица с квадратчета. Всяко квадратче със земя (част от остров) беше отбелязано със знак “X”, освен полето на което се намираше Силви текущо (което е отбелязано със “S”). Имаше и 2 вида вода между островите: с “.” е отбелязана плитка част от океана, над която е подходящо да бъде построен мост, а с “*” са отбелязани по-дълбоките части, които не позволяват строенето на мост над тях. Поради естетически причини, Силви може да строи мост строго над определено квадратче от матрицата на картата, като за да е свързан моста, всеки 2 съседни участъка от него, трябва да са построени над квадратчета с обща страна. Остана да уточним, че един остров представляваше съвкупност от квадратчета, за които от всяко може да се стигне до всяко друго като се преминава единствено по страните на квадратчетата (а не по диагонал).

Вход:

На първия ред на стандартния вход стои T : броят на тестовете. За всеки тест на първия ред стоят числата N и M , които представляват броя на редовете и колоните в картата на Силви. На всеки от следващите N реда има по M знака, описващи всеки ред на картата на Силви. Гарантирано картата ще е валидна по описаното по-горе.

**Изход:**

За всеки тест на отделен ред изведете едно число: минималното необходимо време (в минути) за достигане от острова на който се намира Силви до най-отдалечения от него остров. Знае се, че преминаване от едно квадратче в друго (независимо дали по мост или по суша) отнема 1 минута. Ако е невъзможно да се построят мостове по такъв начин, че да може да се стигне до всеки друг остров, вместо това изведете **-1**.

Ограничения:
 $1 \leq N, M \leq 50$

Пример:

Пример: Вход	Изход
3	4
4 5	-1
.XS..	0
.X.XX	
.**..	
..*X.	
3 5	
S*.XX	
*..X.	
X.....	
4 5	
*S*XX	
XX**X	
X***X	
XXXXX	

Пояснение:

В първия пример островът, до който ще отнеме най-много време да стигнем, е острова долу вдясно (4, 4). За да стигнем до там, можем да стигнем най-бързо ако построим мост на полетата (2, 3) и (3, 4) – преминаваме по първото квадратче до втория остров, през втория остров и през втория мост до последния остров. Така времето необходимо да стигнем до него е 3 (качване на моста, преминаване до второто квадратче на моста и преминаване на отново на суша). Важно е в случая, че можем да тръгнем от която точка си искаме от началния остров и да пристигнем на произволна точка от крайния остров.

Във втория пример от началния остров не можем да стигнем до никой от другите 2 заради дълбоката вода около първия остров.

В третия пример не можем да построим никакви мостове. За щастие има само 1 остров и не се и налага – нужно ни е 0 време за да стигнем до най-далечния остров (ние вече сме на него).



Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 20 април 2013 г.

C. Phone List

Given a list of phone numbers, determine if it is consistent in the sense that no number is the prefix of another. Let's say the phone catalogue listed these numbers:

- Emergency 911
- Alice 97 625 999
- Bob 91 12 54 26

In this case, it's not possible to call Bob, because the central would direct your call to the emergency line as soon as you had dialed the first three digits of Bob's phone number. So this list would not be consistent.

Input

The first line of input gives a single integer, $1 \leq t \leq 40$, the number of test cases. Each test case starts with n , the number of phone numbers, on a separate line, $1 \leq n \leq 10000$. Then follows n lines with one unique phone number on each line. A phone number is a sequence of at most ten digits.

Output

For each test case, output ``YES" if the list is consistent, or ``NO" otherwise.

Sample Input

```
2
3
911
97625999
91125426
5
113
12340
123440
12345
98346
```

Sample Output

```
NO
YES
```



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 20 април 2013 г.

D. Optimal Parking

When shopping on Long Street, Michael usually parks his car at some random location, and then walks to the stores he needs. Can you help Michael choose a place to park which minimizes the distance he needs to walk on his shopping round?

Long Street is a straight line, where all positions are integer. You pay for parking in a specific slot, which is an integer position on Long Street. Michael does not want to pay for more than one parking though. He is very strong, and does not mind carrying all the bags around.

Input

The first line of input gives the number of test cases, $1 \leq t \leq 100$. There are two lines for each test case. The first gives the number of stores Michael wants to visit, $1 \leq n \leq 20$, and the second gives their n integer positions on Long Street, $0 \leq x_i \leq 99$.

Output

Output for each test case a line with the minimal distance Michael must walk given optimal parking.

Sample Input

```
2
4
24 13 89 37
6
7 30 41 14 39 42
```

Sample Output

```
152
70
```



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 20 април 2013 г.

E. Shopaholic

Lindsay is a shopaholic. Whenever there is a discount of the kind where you can buy three items and only pay for two, she goes completely mad and feels a need to buy all items in the store. You have given up on curing her for this disease, but try to limit its effect on her wallet.

You have realized that the stores coming with these offers are quite selective when it comes to which items you get for free; it is always the cheapest ones. As an example, when your friend comes to the counter with seven items, costing 400, 350, 300, 250, 200, 150, and 100 dollars, she will have to pay 1500 dollars. In this case she got a discount of 250 dollars. You realize that if she goes to the counter three times, she might get a bigger discount. E.g. if she goes with the items that costs 400, 300 and 250, she will get a discount of 250 the first round. The next round she brings the item that costs 150 giving no extra discount, but the third round she takes the last items that costs 350, 200 and 100 giving a discount of an additional 100 dollars, adding up to a total discount of 350.

Your job is to find the maximum discount Lindsay can get.

Input

The first line of input gives the number of test scenarios, $1 \leq t \leq 20$. Each scenario consists of two lines of input. The first gives the number of items Lindsay is buying, $1 \leq n \leq 20000$. The next line gives the prices of these items, $1 \leq p_i \leq 20000$.

Output

For each scenario, output one line giving the maximum discount Lindsay can get by selectively choosing which items she brings to the counter at the same time.

Sample Input

```
1
6
400 100 200 350 300 250
```

Sample Output

```
400
```



Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 20 април 2013 г.

F. Moogle

You got the original idea of making map software, called Moogle Maps, for the new cool Maple mPhone. It will even be capable of indicating the location of a house address like "Main Street 13". However, since the mPhone has limited storage capacity, you need to reduce the data amount. You don't want to store the exact location of every single house number. Instead only a subset of the house numbers will be stored exactly, and the others will be linearly interpolated. So you want to select house numbers that will minimise the average interpolation error, given how many house locations you have capacity to store. We view the street as a straight line, and you will always store the first and the last house location.

Given that you've stored the locations x_i and x_j for the houses with numbers i and j respectively, but no other house in between, the interpolated value for a house with number k with $i < k < j$ is

$$x_i + (x_j - x_i) \cdot \frac{k-i}{j-i}$$

Input

The first line of input gives a single integer, $1 \leq t \leq 50$, the number of test cases. For each test case, there are two lines. The first contains $2 \leq h \leq 200$ and $2 \leq c \leq h$, where h is the number of houses in the street and c is the number of house locations that can be stored. The second contains h integers in increasing order giving the location of the h houses. Each location is in the interval $[0, 1000000]$.

Output

For each test case, output the average interpolation error over all the h houses for the optimal selection of c house locations to store. The output should be given with four decimal places, but we will accept inaccuracies of up to ± 0.001 .

Sample Input

```
2
4 3
0 9 20 40
10 4
0 10 19 30 40 90 140 190 202 210
```

Sample Output

```
0.2500
0.3000
```



Департамент Информатика
 Школа „Състезателно програмиране”
 СЪСТЕЗАНИЕ, 20 април 2013 г.

G. Clickomania

Clickomania is a puzzle in which one starts with a rectangular grid of cells of different colors. In each step, a player selects ("clicks") a cell. All connected cells of the same color as the selected cell (including itself) are removed if the selected cell is connected to at least one other cell of the same color. The resulting "hole" is filled in by adjacent cells based on some rule, and the object of the game is to remove all cells in the grid.

In this problem, we are interested in the one-dimensional version of the problem. The starting point of the puzzle is a string of colors (each represented by an uppercase letter). At any point, one may select (click) a letter provided that the same letter occurs before or after the one selected. The substring of the same letter containing the selected letter is removed, and the string is shortened to remove the hole created. To solve the puzzle, the player has to remove all letters and obtain the empty string. If the player obtains a non-empty string in which no letter can be selected, then the player loses.

For example, if one starts with the string "ABBAABBAAB", selecting the first "B" gives "AAABBAAB". Next, selecting the last "A" gives "AAABBB". Selecting an "A" followed by a "B" gives the empty string. On the other hand, if one selects the third "B" first, the string "ABBAAAAB" is obtained. One may verify that regardless of the next selections, we obtain either the string "A" or the string "B" in which no letter can be selected. Thus, one must be careful in the sequence of selections chosen in order to solve a puzzle. Furthermore, there are some puzzles that cannot be solved regardless of the choice of selections. For example, "ABBAAAAB" is not a solvable puzzle.

Some facts are known about solvable puzzles:

- The empty string is solvable.
- If **x** and **y** are solvable puzzles, so are **xy**, **AxA**, and **AxAyA** for any uppercase letter **A**.
- All other puzzles not covered by the rules above are unsolvable.

Given a puzzle, your task is to determine whether it can be solved or not.

Each case of input is specified by a single line. Each line contains a string of uppercase letters. Each string has at least one but no more than 150 characters. The input is terminated by the end of file.

For each input case, print **solvable** on a single line if there is a sequence of selections that solves the puzzle. Otherwise, print **unsolvable** on a line.

Sample Input	Sample Output
ABBAABBAAB	solvable
ABBAAAAB	unsolvable



Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 20 април 2013 г.

Н. ЗЕМЕРАЗДЕЛЯНЕ

Разделянето на имущество между двама братя е трудна работа, особено когато трябва да се дели земя. Обикновено наследството се състои от много отделни парчета и всеки от братята трябва да получи такива парчета, че сумата от площите им да е половината от наследеното. А това никак не е просто, като се има предвид, че никое от парчетата не може да се разбива на по-малки парчета заради законите за уедряване на земеделските земи. Затова понякога се налага да не се дели поравно, а така, че разликата между двата дяла да е колкото може по-малка. Напишете програма, която да намира минималната възможна разлика между двата дяла.

На първия ред на стандартния вход е зададен броят на тестовете. Всеки тест започва с ред, на който е зададен броят N на парчетата земя, които трябва да бъдат поделени ($N < 1000$). На втория ред са зададени, разделени с по един интервал, големините на участъците – цели числа, всяко от които не надхвърля 100.

За всеки тест на един ред на стандартния изход програмата трябва да изведе намерената минимална разлика.

ПРИМЕР Вход	Изход
3	0
3	2
1 2 3	1
3	
2 2 2	
9	
1 2 3 4 5 6 7	
8 9	



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 20 април 2013 г.

I. Данъчни проблеми

Силви спестяваше пари дълго време, с надеждата накрая да спести достатъчно пари, така че да се махне от коварния бизнес свят и да се оттегли в провинцията. Най-сетне тя събра парите и реши да ги вложи в нещо с добра възвръщаемост, а именно – да купи обработваема земя, на която да гледа ориз (с който добре знаем, че човек може да си се изхранва без да има други грижи). Тя си хареса да купи земите си в Китай, защото там хем беше евтина земята (все пак си имат много), хем когато купуваш земя и веднага с нея включват по няколко китайчета, които да ти я обработват доживот за без пари! Силви веднага сметна колко квадратни метра земя може да купи със заделените пари и започна да се чуди в каква форма да купи земята, така че да е най-красива. Оказа се, обаче, че поради данъчни причини, в Китай земята се продава единствено на квадратни парцели със страни – цели числа в метри. Облагането с данъци ставало на парцел (независимо от размера му), така че Силви се отказа от идеята за красивата форма на участъка от земя който ще вземе и се съсредоточи на опити да минимизира броя парцели, които ще купи, по начин, така че да изхарчи всичките си предвидени за целта пари. Понеже тя беше събрала доста пари, това се оказа сложна задача и тя реши да се обърне към вас с молба за помощ. Помогнете на Силви да сметне минималния брой парцели, които трябва да купи, така че сумарната им площ да е толкова квадратни метра, колкото може да си позволи да купи Силви.

Вход:

На първия ред на стандартния вход стои числото **T** – броя на тестовите примери. За всеки тест, на единствен ред ще стои числото **N** – броя квадратни метри земя, които може да купи Силви.

Изход:

За всеки тест на отделен ред изведете минималния брой парцела, които Силви трябва да купи.

Ограничения:

$$1 \leq N \leq 60,000$$

Пример:

Вход	Изход
3	3
3	1
9	3
344	



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 20 април 2013 г.

J. Метро зони

Силви и Краси спестяваха цяла зима и най-накрая събраха пари за екскурзия до Лондон! Още на първия ден, обаче, те се загубиха някъде край една голяма часоникова кула, на която местните наоколо викаха Биг Бен. Това, в което бяха сигурни е, че хотелът им е около Уестминстърското Абатство, но не искаха да рискуват да се изгубят още повече, затова решиха да хванат прочутото лондонско метро.

За тяхна изненада Underground-ът се оказа доста сложно устроен. Цял Лондон е разделен на идеални кръгове, т.нар. метро зони, никои две от които не се пресичат (дори не се допират), но е възможно да са вложени една в друга. За преминаването от една зона в друга се двамата общо трябваше да заплатят скромната сума от £1.

След около час, прекаран пред огромната карта на ситито, Силви и Краси вече знаят къде точно се намират и къде точно се намира и хотелът им! Остава само да намерят такъв начин да стигнат до него, използвайки метрото, така че минимизират разходите си. Както може би се сещате, това е трудоемка задача както за Силви, така и за красивия, романтичен и аристократичен студент по испански (а.к.а Краси).

Вход: На първия ред от стандартния вход стои T , броят на тестовете. За всеки тест на първия ред стои едно естествено число N ($1 \leq N \leq 10\,000$), броят на метро зоните в Лондон; на следващите N реда има по три числа – координатите на x_i и y_i и радиусът r_i на i -тата зона ($-1\,000\,000\,000 \leq x_i, y_i \leq 1\,000\,000\,000$; $0 \leq r_i \leq 1\,000\,000\,000$); на последните два реда са координатите skx и sky на Силви и Краси и h_x и h_y – координатите на хотела им.

Изход: Минималната сума, която Силви и Краси трябва да изхарчат, за да се приберат в хотела си.

Пример:

Вход

2

1

1 9 5

-1 9

1 8

2

0 0 100

0 0 10

0 0

20 20

Изход

0

1