GLOBUSSOFT

TECHNOLOGY AHEAD OF TIME

Greetings From Globussoft

- Given below are 5 Programming questions, you have to solve any 3 out of 5 questions.

- These 5 questions you can attempt in any technology like C/C++, java, .Net, PHP

- To solve these 3 questions you've max. 3 hours.

- While Solving these questions you are not allowed to use any **Search Engine** like Google, Yahoo, Bing …

All the best for your test

Globussoft

QUESTION - 1

FJ has purchased N (1 <= N <= 2000) yummy treats for the cows who get money for giving vast amounts of milk. FJ sells one treat per day and wants to maximize the money he receives over a given period time. The treats are interesting for many reasons:

- The treats are numbered 1..N and stored sequentially in single file in a long box that is open at both ends. On any day, FJ can retrieve one treat from either end of his stash of treats.
- Like fine wines and delicious cheeses, the treats improve with age and command greater prices.
- The treats are not uniform: some are better and have higher intrinsic value. Treat i has value v(i) (1 <= v(i) <= 1000).
- Cows pay more for treats that have aged longer: a cow will pay v(i)*a for a treat of age a.

Given the values v(i) of each of the treats lined up in order of the index i in their box, what is the greatest value FJ can receive for them if he orders their sale optimally?

The first treat is sold on day 1 and has age a=1. Each subsequent day increases the age by 1.

Input

Line 1: A single integer, N

Lines 2..N+1: Line i+1 contains the value of treat v(i)

Output

The maximum revenue FJ can achieve by selling the treats

Example

```
Input:
5
1
3
1
5
2

Output:
43
```

QUESTION – 2

In a wireless network with multiple transmitters sending on the same frequencies, it is often a requirement that signals don't overlap, or at least that they don't conflict. One way of accomplishing this is to restrict a transmitter's coverage area. This problem uses a shielded transmitter that only broadcasts in a semicircle.

A transmitter *T* is located somewhere on a 1,000 square meter grid. It broadcasts in a semicircular area of radius *r*. The transmitter may be rotated any amount, but not moved. Given *N* points anywhere on the grid, compute the maximum number of points that can be simultaneously reached by the transmitter's signal. Figure 1 shows the same data points with two different transmitter rotations.

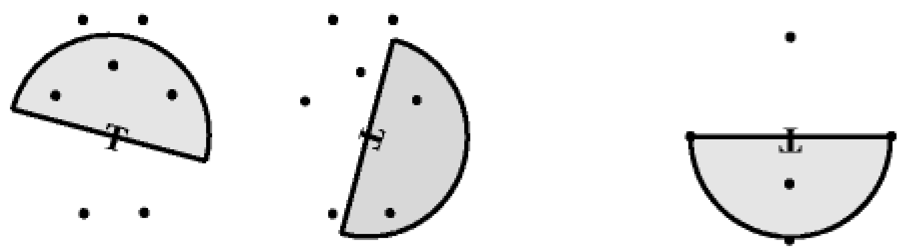


Figure 1a Figure 1b Figure 2

All input coordinates are integers (0-1000). The radius is a positive real number greater than 0. Points on the boundary of a semicircle are considered within that semicircle. There are 1-150 unique points to examine per transmitter. No points are at the same location as the transmitter.

Input consists of information for one or more independent transmitter problems. Each problem begins with one line containing the (x,y) coordinates of the transmitter followed by the broadcast radius, *r*. The next line contains the number of points *N* on the grid, followed by *N* sets of (x,y) coordinates, one set per line. The end of the input is signalled by a line with a negative radius; the (x,y) values will be present but indeterminate. Figures 1 and 2 represent the data in the first two example data sets below, though they are on different scales. Figures 1a and 2 show transmitter rotations that result in maximal coverage.

For each transmitter, the output contains a single line with the maximum number of points that can be contained in some semicircle.

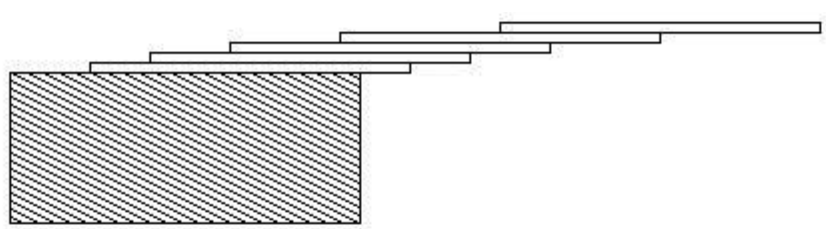
Input:

```
25 25 3.5
7
25 28
23 27
27 27
24 23
26 23
24 29
```

```
26 29
350 200 2.0
5
350 202
350 199
350 198
348 200
352 200
995 995 10.0
4
1000 1000
999 998
990 992
1000 999
100 100 -2.5
```
Output:
```
3
4
4
```

QUESTION – 3

How far can you make a stack of cards overhang a table? If you have one card, you can create a maximum overhang of half a card length. (We're assuming that the cards must be perpendicular to the table.) With two cards you can make the top card overhang the bottom one by half a card length, and the bottom one overhang the table by a third of a card length, for a total maximum overhang of 1/2 + 1/3 = 5/6 card lengths. In general you can make *n* cards overhang by 1/2 + 1/3 + 1/4 + ... + 1/(*n* + 1) card lengths, where the top card overhangs the second by 1/2, the second overhangs tha third by 1/3, the third overhangs the fourth by 1/4, etc., and the bottom card overhangs the table by 1/(*n* + 1). This is illustrated in the figure below.



Input

The input consists of one or more test cases, followed by a line containing the number 0.00 that signals the end of the input. Each test case is a single line containing a positive floating-point number *c* whose value is at least 0.01 and at most 5.20; *c* will contain exactly three digits.

Output

For each test case, output the minimum number of cards necessary to achieve an overhang of at least *c* card lengths. Use the exact output format shown in the examples.

```
Input:
1.00
3.71
0.04
5.19
0.00
Output:
3 card(s)
61 card(s)
1 card(s)
273 card(s)
```

QUESTION – 4

Technicians in a pathology lab analyze digitized images of slides. Objects on a slide are selected for analysis by a mouse click on the object. The perimeter of the boundary of an object is one useful measure. Your task is to determine this perimeter for selected objects.

The digitized slides will be represented by a rectangular grid of periods, '.', indicating empty space, and the capital letter 'X', indicating part of an object. Simple examples are

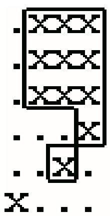
```
XX Grid 1 .XXX Grid 2
XX .XXX
.XXX
...X
..X.
X...
```

An X in a grid square indicates that the entire grid square, including its boundaries, lies in some object. The X in the center of the grid below is *adjacent* to the X in any of the 8 positions around it. The grid squares for any two adjacent X's overlap on an edge or corner, so they are connected.

XXX
X**X**X Central X and adjacent X's
XXX

An object consists of the grid squares of all X's that can be linked to one another through a sequence of adjacent X's. In Grid 1, the whole grid is filled by one object. In Grid 2 there are two objects. One object contains only the lower left grid square. The remaining X's belong to the other object.

The technician will always click on an X, selecting the object containing that X. The coordinates of the click are recorded. Rows and columns are numbered starting from 1 in the upper left hand corner. The technician could select the object in Grid 1 by clicking on row 2 and column 2. The larger object in Grid 2 could be selected by clicking on row 2, column 3. The click could not be on row 4, column 3.



One useful statistic is the perimeter of the object. Assume each X corresponds to a square one unit on each side. Hence the object in Grid 1 has perimeter 8 (2 on each of four sides). The perimeter for the larger object in Grid 2 is illustrated in the figure at the left. The length is 18.

Objects will not contain any totally enclosed holes, so the leftmost grid patterns shown below could *NOT* appear. The variations on the right could appear:

```
Impossible Possible

XXXX XXXX XXXX XXXX
X..X XXXX X... X...
XX.X XXXX XX.X XX.X
XXXX XXXX XXXX XX.X

..... ..... ..... .....
..X.. ..X.. ..X.. ..X..
.X.X. .XXX. .X... .....
..X.. ..X.. ..X.. ..X..
..... ..... ..... .....
```

The input will contain one or more grids. Each grid is preceded by a line containing the number of rows and columns in the grid and the row and column of the mouse click. All numbers are in the range 1-20. The rows of the grid follow, starting on the next line, consisting of '.' and 'X' characters.

The end of the input is indicated by a line containing four zeros. The numbers on any one line are separated by blanks. The grid rows contain no blanks.

For each grid in the input, the output contains a single line with the perimeter of the specified object.

Input:

```
2 2 2 2
XX
XX
6 4 2 3
.XXX
.XXX
.XXX
...X
..X.
X...
5 6 1 3
```

```
.XXXX.
X....X
..XX.X
.X...X
..XXX.
7 7 2 6
XXXXXXX
XX...XX
X..X..X
X..X...
X..X..X
X.....X
XXXXXXX
7 7 4 4
XXXXXXX
XX...XX
X..X..X
X..X...
X..X..X
X.....X
XXXXXXX
0 0 0 0
```

Output:

```
8
18
40
48
8
```

QUESTION – 5

A string s is called an (k,l)-repeat if s is obtained by concatenating k>=1 times some seed string t with length l>=1. For example, the string

s = abaabaabaaba

is a (4,3)-repeat with t = aba as its seed string. That is, the seed string t is 3 characters long, and the whole string s is obtained by repeating t 4 times.

Write a program for the following task: Your program is given a long string u consisting of characters ‘a’ and/or ‘b’ as input. Your program must find some (k,l)-repeat that occurs as substring within u with k as large as possible. For example, the input string

u = babbabaabaabaabab

contains the underlined (4,3)-repeat s starting at position 5. Since u contains no other contiguous substring with more than 4 repeats, your program must output the maximum k.

Input

In the first line of the input contains H- the number of test cases (H <= 20). H test cases follow. First line of each test cases is n - length of the input string (n <= 50000), The next n lines contain the input string, one character (either ‘a’ or ‘b’) per line, in order.

Output

For each test cases, you should write exactly one interger k in a line - the repeat count that is maximized.

Example

Input:

```
1
17
b
a
b
b
a
b
a
a
b
a
a
b
a
a
b
a
b
```

Output:

```
4
```