1. The Blacksmith

Program Name: Blacksmith.java **Input File: None**

You are a blacksmith in a rural town. Unfortunately, you broke your tools and can't create swords by hand! Luckily, the town was recently granted a 3D printer. Write a program that prints out the text art of the sword in the example output below.

Input
There is no input for the file.

Output
Your output will be exactly as shown in the example output below.

Example Input File
There is no input for the file.

Example Output to Screen
```
          &&&
          &
********  &  ----------------\
********  &  ----------------/
          &
          &&&
```

2. The Banker

Program Name: Banker.java **Input File: Banker.dat**

You are a banker at a local bank, and your job is to determine the final amount in the customer's bank account given a series of monetary transactions. Write a program that prints out the result of a sequence of adding and subtracting money from an account.

Input

The first line of input will contain an integer `n`, which represents the number of test cases. Each test case will start with an integer `m` on a new line. The next `m` lines will each be a decimal to the hundredths place. Each decimal will be preceded by either a + or a −. A + represents a deposit (addition) of money into the account and a − represents a withdrawal (subtraction) of money from the total. The amount of money in the account will never be below $0.

Output

Output the total amount of money in the account after all of the transactions take place. Print the total preceded by a `$` and rounded to the hundredths place. Print the solution to each case on a new line.

Example Input File

```
2
2
+1.67
+15.60
3
+.65
+1.03
-1.00
```

Example Output to Screen

```
$17.27
$0.68
```

3. The Teacher

Program Name: Teacher.java **Input File: Teacher.dat**

As a teacher, you spend a large amount of time processing grades. Fortunately, you can make your computer do all the work for you! Write a program that, given a set number of grades, prints out the average of the grades to the hundredths place.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with a single integer `x` denoting how many grades to process and average. The next `x` lines will each contain an integer representing a grade to be processed.

Output

Output the average of the grades for each test case. The average is determined by dividing the sum of all of the grades by the number of grades in the test case. Your output should round to the hundredths place and each solution should be on a new line.

Example Input File

```
2
3
100
95
90
2
100
75
```

Example Output to Screen

```
95.00
87.50
```

4. The Librarian

Program Name: Librarian.java **Input File: Librarian.dat**

You're a librarian at a local library. Each day, you receive a shipment of 10 books in a random order. Your job is to sort the books in order for them to be shelved correctly. Your library is unique in that the books are sorted alphabetically from the end of the title to the beginning. Write a program that takes in 10 books and returns them sorted in alphabetical order from the end of the title to the beginning.

Input

The first line of input will contain an integer `n`, which represents the number of test cases. Each test case will be 10 book titles, each a single word and each on a new line. There will be a blank line between each case.

Output

Print out the books, sorted in alphabetical order from the end of the title to the beginning.

Example Input File

```
1
MOBYDICK
ULYSSES
DONQUIXOTE
HAMLET
WARANDPEACE
THEODYSSEY
THEGREATGATSBY
THEADVENTURESOFHUCKLEBERRYFINN
THEILIAD
PRIDEANDPREJUDICE
```

Example Output to Screen

```
THEILIAD
WARANDPEACE
PRIDEANDPREJUDICE
DONQUIXOTE
MOBYDICK
THEADVENTURESOFHUCKLEBERRYFINN
ULYSSES
HAMLET
THEGREATGATSBY
THEODYSSEY
```

5. The Designer

Program Name: Designer.java **Input File: Designer.dat**

You are a designer who specializes in bowties. Your customers send you the size of the bowtie, the number of bowties, and the material to make the bowties out of. The height of the bowtie determines the length of a side of the bowtie and the material determines what character is used to create the bowtie. Write a program that will create a bowtie image from the information given. The bowtie is two triangles facing toward each other with a circle in the middle in the form of `[]` as shown in the example output below. If the customer requests more than one bowtie, output a bowtie to the right of the previous, with a space between each bowtie.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will be one line in the format `a b c` where `a` is an odd integer denoting the length of the bowtie, `b` is the number of bowties, and `c` is the material to use to make the requested bowties.

Output

Output the bowties as requested. If the customer requests more than one bowtie, the bowties will be to the right of the first bowtie, with a space between each bowtie. Each bowtie is made of two triangles and `[]` where the two triangles meet. The triangles will be in the format as shown below in the example output, and the bowtie will be made of the material from the input.

Example Input File

```
2
5 2 #
11 1 !
```

Example Output to Screen

```
#       # #       #
##     ## ##     ##
# #[]# # # #[]# #
##     ## ##     ##
#       # #       #

!              !
!!            !!
! !          ! !
!  !        !  !
!   !      !   !
!    !    !    !
!     ![]!     !
!    !    !    !
!   !      !   !
! !          ! !
!!            !!
!              !
```

6. The Cryptologist

Program Name: Cryptologist.java **Input File: Cryptologist.dat**

As a cryptologist, your job is to study and decipher ancient codes and ciphers. You were just handed a cipher that you must write a program to decode. The cipher works as follows:

- The text to decode is entered character by character into a 5x5 matrix, starting from the top left corner and placing each character moving down a row each time. If a column is full, start on the top of the next column to the right.
- Rotate every other row to the right one, starting with the top row. This means you are rotating the top row, the middle row, and the bottom row. The characters on the rightmost cell will wrap back to the leftmost cell.
- The decoded text is then pulled from the matrix, starting from the bottom right cell and taking each character moving up a row each time. Move to the bottom of the column to the left when you reach the top of the column.

For example, the text
"`wssrht eioh ypla rlI.ek o`"
would be put into the matrix as:

.	w	t	h	a
s				e
k	s	e	y	r
r	i	p	l	
o	h	o	l	I

When the rows are rotated, the matrix will look like this:

The decoded text becomes "`I really hope this works.`"

w	t	h	a	.
s				e
s	e	y	r	k
r	i	p	l	
h	o	l	I	o

Input

The first line of input will contain an integer `n`, which represents the number of test cases. Each test case will be a line of text that is 25 characters long. Each case will be on a separate line.

Output
Output the decoded text, one test case per line.

Example Input File
```
2
wssrht eioh ypla rlI.ek o
oy rnih eetlarahc r2!ca5o
```

Example Output to Screen
```
I really hope this works.
25 character line hooray!
```