

## 1. Castle of Georgelandia

Program Name: Castle.java

Input File: None

George is a king. He has a kingdom because he is king. He is king of the kingdom. He names the land Kingdom of George. George also calls his lands Georgelandia. George is king of Georgelandia. George has a castle in Georgelandia. Because George is king, George lives in the Castle of Georgelandia.

### Output

Print the Castle of Georgelandia

#### Example Output to Screen

```
      /:\
    /.\  \:/  /.\
   ).(   |:|  ).(
  /... \ /::\ /... \
 ).( )+: ( )... (
 ).+( )::: ( ).+(
 )... ( )... (
 |.....|
 |...../<<<<<\.....|
 |.....|>>>>>|.....|
 |.....|<<<<<|.....|
```

## 2. Flag of Georgelandia

Program Name: Flag.java

Input File: flag.dat

The Kingdom of George is the greatest kingdom on the face of this Earth. George wants the people of his kingdom to show some national pride. To do this, George needs your help. He needs you to help manufacture the flags of his nation to give out to people. The flags come in different sizes, but don't worry about the flag design: George made sure that the people would remember the flag: a blank, white flag! The border of the flag should be hashtags: "#", while the inside of the flag should be spaces: " ". Help George out with this task!

### Input

The first number  $t$  describes the number of test cases to follow. Each test case will comprise of two numbers,  $n$  and  $m$ , representing the height and width of the flag, respectively.

### Output

For each test case, print out the flag. After each test case, print out 10 hyphens:

"-----"

### Example Input File

2  
5 10  
4 6

*call new # method*

### Example Output to Screen

```
1 #####
2 #           #
3 #           #
4 #           #
5 #####
-----
#####
#           #
#           #
#####
-----
```

*first line is repeat for i until i < width  
# spaces = width - 2*

*if char = " " of*

*Substring(0, index of space)  
Substring(index of space + 1, length + 1)*

*inputfile.indexOf(" ")*

*(0, 6)*

*length = 6 + 1*

0 1 2 3 4 5  
S U B S T R

length = 6 + 1

### 3. The Conquest for Georgelandia

Program Name: Conquest.java

Input File: conquest.dat

George, the king of Georgelandia, has conquered a lot of land. So much land that nobody knows how much territory he claimed! George wants to know exactly how much land he claimed. He will give you a map of conquered territory, where "x" (lowercase x) will represent one unit of conquered territory and "." will represent one unit of unconquered territory. He wants to know exactly how many units of territory he conquered. Can you help George?

#### Input

The first number  $t$  describes the number of test cases to follow. Each test case will have two numbers,  $n$  and  $m$ , describing the height and width of the map. The following  $n$  lines contain a string of length  $m$ , each representing a line of the map. The map will only contain the characters "x" and ".".

#### Output

For each test case, print "George has conquered  $X$  unit(s) of territory", where  $X$  is replaced by the number of units of conquered territory.

#### Example Input File

```
2
3 3
...
.x.
...
4 4
xxxx
xxxx
xxxx
xxx.
```

for rows  
for width

note how many x

if char = "x"  
count ++

#### Example Output to Screen

```
George has conquered 1 unit(s) of territory-
George has conquered 15 unit(s) of territory
```

## 4. The Georgelandia Registry

**Program Name:** Registry.java

**Input File:** registry.dat

George is proud to be king of a nation whose population is booming! Because of this, George needs information about the people of his nation. He has set up a national registry that keeps track of all of his citizens, but he needs some statistics on that information. More specifically, he needs the mean age and the median annual income of all his citizens. George will send you the records, and each record will contain the citizen's full name, age, and annual income. Can you help George learn about his people?

### Input

The first integer  $n$  indicates the number of citizens in the registry. Each of the next  $n$  lines will represent a record in the registry, which contain a citizen's full name, age, and annual income separated by commas.

### Output

Print "Georgelandia's Mean Age: " followed by the average age of Georgelandia rounded to 2 decimal places. On the next line, print "Georgelandia's Median Annual Income: \$" followed by the median annual income of Georgelandia rounded to 2 decimal places.

### Example Input File

```
3
King George, 26, $188780.00
Elder Greg, 72, $36156.50
Merchant Georgia, 46, $30034.25
```

### Example Output to Screen

```
Georgelandia's Mean Age: 48.00
Georgelandia's Median Annual Income: $36156.50
```

## 5. Georgelandia's Budget Crisis

Program Name: Budget.java

Input File: budget.dat

Georgelandia is a vast nation that has conquered many lands. However, all that conquest does not come cheap. In fact, the wars have put Georgelandia in some serious debt. Fortunately, George can always tax his citizens more to balance out the budget. He needs you to help him figure out how much income tax he needs to enact in order to zero out his budget. He will give you the total income of all Georgelandia citizens as well as the current budget deficit, and you will tell him the income tax percentage in order to zero out the budget. He asks that if the percentage is in between two numbers, you should always round up. Can you help George?

### Input

The first number  $t$  describes the number of test cases to follow. Each test case contains two doubles, the first one representing the total income of the citizens of Georgelandia, and the second representing the budget deficit. The total income of citizens will always be greater than the deficit.

### Output

For each test case, print the minimum income tax percent that would move the budget out of deficit. If the percent is in between two numbers (33.25% for example), always round up to the next number (34%)

#### Example Input File

2      *total income   deficit*  
2000000.00 100000.00  
2531220.00 379683.00

#### Example Output to Screen

5%  
15%

$$(x) \left( \frac{\text{deficit}}{\text{total}} \right) = \text{deficit}$$
$$\left( \frac{\text{deficit}}{\text{total}} \right) (100)$$

## 6. The Great Battle of Georgelandia

**Program Name:** Battle.java

**Input File:** battle.dat

Georgelandia is under attack! The Republic of William has brought an army to the lands of Georgelandia, and you need to stop their attack! The Republic of William has multiple platoons of troops attacking different places, so George must split his troops up in order to defeat them. A George soldier cannot fight more than one platoon at once, so the troops must be split wisely. A platoon of William soldiers can be stopped if the number of George troops to stop them are at least half of the William platoon size. George is unsure if it's possible to stop their attack. Can you help George determine if his troops should retreat or fight back?

### Input

The first number  $t$  will describe the number of test cases to follow. Each test case will contain two numbers,  $n$  and  $k$ , describing the number of William platoons and the number of George soldiers at your disposal, respectively. The next line will contain  $n$  numbers, each representing the number of William troops in the platoon.

### Output

For each test case, print "Defend the Great Lands of Georgelandia!" if it is possible to defeat all the William platoons or "Retreat!!!" if otherwise.

### Example Input File

```
2
5 15
4 1 2 3 2
5 15
3 4 3 15 4
```

### Example Output to Screen

```
Defend the Great Lands of Georgelandia!
Retreat!!!
```



## 7. The Georgelandia Trade Reorganization

**Program Name:** Trade.java

**Input File:** trade.dat

Georgelandia's trade empire has expanded greatly. So much greatly that the trade dock is a complete mess now, with shipping containers scattered everywhere! George has decided that it is of great importance that the containers be organized as soon as possible. He wants them to be sorted in ascending order by volume of the container, followed by the alphabetical order of the name of the container. George knows of your computer science background, and requests that you help him write a program that quickly rearranges the order based on his criteria. Can you help George?

### Input

The first integer  $t$  describes the number of test cases. Each test case begins with a number  $n$  indicating the number of shipping containers to sort. Each of the next  $n$  lines contains the name of a shipping container as well as its dimensions.

### Output

For each test case, print the names of the shipping containers, each on a separate line, in the order described above. Print 10 equals signs at the end of every test case.

### Example Input File

```
1
3
Evergreen 10x10x2
Georgia 5x5x5
Paul 2x10x10
```

### Example Output to Screen

```
Georgia
Evergreen
Paul
=====
```

## 8. The Population of Georgelandia

Program Name: Population.java

Input File: population.dat

As you already know, the population of Georgelandia is expanding rapidly. King George knows that while this is great news, it does mean that the infrastructure of Georgelandia will be strained in these next coming years. In order to prepare for this, the Kingdom of Georgelandia needs to be able to estimate the population over time to see what they will need for the future. Luckily, George already knows that the population of Georgelandia grows by 5% every year. Given the current population, can you estimate what the population will be after a certain number of years?

### Input

The first number  $t$  indicates the number of test cases to follow. Each test case will have 2 numbers representing the current population and the number of years George needs you to estimate.

### Output

For each test case, print "Population after  $X$  years:  $Y$ ", where  $X$  represents the number of years and  $Y$  represents the estimate population at that time rounded to the nearest digit.

### Example Input File

```
3
28 4
37 3
74 2
```

### Example Output to Screen

```
Population after 4 years: 34
Population after 3 years: 43
Population after 2 years: 82
```

$og\ pop\ (1.05)^{years}$



## 9. The History of Georgelandia

Program Name: History.java

Input File: history.dat

Georgelandia's history is very interesting. The Department of History is responsible for keeping track of all the glorious events. The Department of History needs your help classifying the events as either battles or reforms or neither. A battle is an event that either contains the word "siege" or "battle", and a reform is an event that contains the word "act" or "reform". If an event is both a battle and a reform or is neither, classify the event as neither. Help the Department of History classify the events!

### Input

The first number  $n$  indicates how many lines are to follow. Each line contains an event.

### Output

Print "X battle(s), Y reform(s), Z neither.", where X, Y, and Z are replaced with the number of battles, reforms, and events that were neither.

### Example Input File

4

Reform of 100

The Great Siege

The battle act

The Irrelevant Event

### Example Output to Screen

1 battle(s), 1 reform(s), 2 neither.

## 10. Invasion into the Republic of Paul

**Program Name:** Invasion.java

**Input File:** invasion.dat

George himself has decided to raid the Republic of Paul! George is currently on the battlefield, and needs your help determining how many enemies he can slay. On the battlefield, George is represented with a "G", empty spaces are represented with a ".", obstacles are represented with a "#", and enemies are represented with a "P". George can only move in the cardinal directions (North, South, East, West) and cannot move over obstacles. George is so great at combat that he can slay any enemy that he can reach to. Can you help him figure out how many enemies he can slay?

### Input

The first number  $t$  indicates the number of test cases to follow. Each testcase will begin with two numbers,  $n$  and  $m$ , representing the number of rows and columns in the battlefield. The next  $n$  lines contain a string length  $m$  that represents the battlefield.

### Output

For each testcase, print "George can slay  $X$  enemies", where  $X$  is replaced with the number of enemies George can slay. If George can only slay one enemy, print "George can slay 1 enemy".

### Example Input File

```
1
5 5
G....
P..#.
..#P#
...#.
..PPP
```

### Example Output to Screen

```
George can slay 4 enemies
```

## 11. The Great Ruby Rush of Georgelandia

Program Name: Ruby.java

Input File: ruby.dat

Behold, treasure! The Kingdom of Georgelandia has found a vast reservoir of rubies in the ground, and its citizens are excited about it! But the problem is that they're way too excited, and George's chief economist claims that selling so many rubies can crash the market! The economist states that the following equation can model the efficiency of the market,

$$\text{efficiency} = \sin^2(3x)$$

where  $x$  represents the total number of rubies sold from all mines. However, the mines are way too excited to sell the rubies, so for each mine you can decide whether to allow them to sell all of their rubies or none of them. Your task is to write a program that attempts to maximize the efficiency of the market by choosing which mines can sell all of their rubies and which can sell none.

### Input

The first number  $t$  indicates the number of test cases to follow. Each testcase will begin with one number  $n$  that indicates the number of mines. The value of  $n$  will be no greater than 5. The next line will contain  $n$  integers, each representing the number of rubies that mine has.

### Output

For each test case, print out the maximum achievable efficiency as a percentage, rounded to the nearest digit.

### Example Input File

```
2
3
1 2 3
4
7 13 17 19
```

### Example Output to Screen

```
56%
99%
```