Tompkins
January 9, 2014
Heroes & Villains

- **Rules**
  - No more than 3 students per team
  - There will be 15 programs worth 50 points each
    - When a correct program is submitted the team will be awarded 50 points, minus 5 points for each previous incorrect submission
  - Pre-written code is not allowed.
  - Ties will be broken with the team's written scores

- **Allowed Materials**
  - 1 computer
  - 1 mouse
  - 1 keyboard
  - 1 reference book
  - 1 printer

Spiderman and Debris

- **Input File**: spiderman.in
- **Output**: console

Problem Description

Venom has destroyed many of the skyscrapers of Manhattan. Spiderman, standing on the New York Stock Exchange, looks at the debris on Wall Street. He notices that he needs to throw the debris away and if a person is under it, he needs to rush him/her back to NYSE. The "Math-Hating" Peter Parker calls you to check how much time it will take him to clear off all the debris and save the people. You need to create a program to calculate the time in seconds needed to clear off the debris and save the victims.

Notes

1. Spiderman ALWAYS starts from NYSE.
2. Spiderman will completely clear all the debris on the left side of NYSE before starting work on the right side.
3. Each step towards the debris takes one second.
4. A debris pile without a victim takes 2 seconds to clear.
5. A debris pile with a victim takes 3 seconds to clear and get the victim out.
    a. Once a victim is out Spider swings the victim to NYSE, in 1 second
6. The both ends of Wall Street always have debris.
7. Spiderman swings back to NYSE, in 1 second, after finishing the last debris of an end.
8. The input is expected to only have **ONE NYSE.**

Input Description

The first line will tell you the number of data sets that follow. The lines following will have the whole path that Spiderman has to follow.

- "N" denotes New York Stock Exchange
- "-" denotes a clear area
- "D" denotes a debris pile with a victim
- "d" denotes a debris pile without a victim

Sample Input

3

D-Nd

DddDNdDd

D---d---N---D--d

Sample Output

10

31

32

The Code Hacker

- **Input File**: hacker.in
- **Output**: console

Problem Description

Batman sends a message to his assistant in Pennsynople. Penguin sends his 'Left-Hand' to intercept the message and bring it to him. He later notices that the message is coded. So, he comes to you to decode this message. He tells you that there is a number on the top of the paper and that is the number of letters the message is set forward. You need to create a program that can decode it.

Notes

1. The code is set 'n' number of letters forward
2. All '&' translates to a comma.
3. All ' ' are not translated
4. All '%' translates to exclamation
5. All '$' translates to a period.
6. The only punctuation will be comma, exclamation and periods.

Input Description

1. The first line contains how many letters forward the encoded message is
2. The second line has the encoded message.

Sample Input

2
Rngcug vt{ qwt o{uvke ejggug uswctgu& vjg swggp uc{u vjg{ ctg tgcnn{ {woo{%

Sample Output

Please try our mystic cheese squares, the queen says they are really yummy!

Connect Four Cheater

- **Input File**: c4cheater.in
- **Output**: console

Problem Description

Megavolt and Darkwing Duck have started playing connect four online with each other. Megavolt is continually outmatched by Darkwing's wit, so he turns to you to build him a program. The program he wants will alert him when he needs to block Darkwing or when there is a win available to him.

Notes

1. Darkwing will always play as 'D'
2. Megavolt will always play as 'M'
3. The columns will be listed 0-6
4. If there are wins and blocks available, the program will only list the winning moves
5. If there are no wins available, but there are blocks available the program will list the blocks
6. If there are no wins or blocks available the program will say "No Suggestion"
7. When listing wins the program will need to display "W ", followed by the winning columns separated by spaces.
8. When listing blocks the program will need to display "B ", followed by the blockings columns separated by spaces.
9. Each column can only be listed once and columns should be listed off in numeric order
10. Game Rules
    a. When a piece is placed into a column it will move to the lowest row that does not contain a piece
    b. Players win by getting 4 of their letter in a row (horizontally, vertically or diagonally)

Input Description

The first line of input will be the number of boards to be processed. Each board will be listed as 6 rows of '*' (Empty), ' D' (Darkwing's Piece) and 'M' (Megavolt's Piece). Each row will contain 7 characters. After each board there will be a blank line.

Sample Input/Output Shown on the Next Page

Sample Input

```
3
*******
*******
*******
*D*****
DD*MMM*
DD*DMM*

*******
*******
*******
***DMD*
***MDM*
*D*DDMM

*******
*******
*******
D***D**
D**MD**
MM*DM**
```

Sample Output

```
B 1 2
W 3
No Suggestion
```

Portal Maze

- **Input File**: portalmaze.in
- **Output**: console

Problem Description

A couple years after defeating Captain Hammer, Dr. Horrible found a new love interest named Nickel. Nickel is a hero that helped Dr. Horrible come to the good side of the law. Dr. Horrible and Nickel live in Dr. Horrible's former lair.

Even though Dr. Horrible is now playing for the good team, Captain Hammer wants revenge on Horrible for making him feel pain. He has obtained plans for a portal gun to help him invade Horrible's home, but he is missing 3 components to build his portal gun.

Captain Hammer has to infiltrate GladOS' layer to obtain the parts he needs, but he is afraid that if he spends too much time there GladOS will find a way to hurt him. He has come to you with a map of the installation he is raiding and wants you to determine the shortest amount of time it will take him to gather the 3 components and get out.

Notes

1. Captain Hammer can only move up/down/left/right
2. Each move takes 1 second
3. It takes 0 seconds to grab a component
4. Map Symbols
    a. Entrance / Exit '#'
    b. Walls '*'
    c. Open Ground ' -'
    d. Componens '1', '2', '3'
    e. Portals 'A' – 'Z'
        i. Each portal will have a single match
        ii. When a portal is stepped on the user is teleported to the matching portal letter
        iii. Note stepping onto a portal takes 1 second, but the transportation takes 0 seconds.

Input Description

The first line of input will be the number of mazes to be process. Each maze in the file will start with a line that defines the size of the maze. The format for this line will be columns**x**rows. The maze will follow on the next lines.

Sample Input/Output Shown on the Next Page

Sample Input

```
2
10x8
**********
#--A*-A-B*
*-----3--*
******-***
*---*---B*
*---*1---*
*C2-*C---*
**********
8x4
******#*
*----B1*
*A23B-A*
********
```

Sample Output

```
20 seconds
7 seconds
```

Snake Latin

- **Input File**: snakelatin.in
- **Output**: console

Problem Description

Villymort has been talking to Hairy Hero in a very strange language, called Snake Latin. Hairy Hero, who is desperate to talk to Villymort, comes to you and tells you his obsrevations of Villymort's speech and asks you to write a program to translate English to Snake Latin, so he can talk to Villymort.

1. If the word contains no vowels, then and "ess" to the end.

2. If the word contains only one letter or stats with an 'a', then and "ss" to the front.

3. All other words use the following rule:
   Move all the letters from the start of the word up to and including the first vowel to the end of the word and then add "ss"

Additional notes:
-The translation must not include punctuation or capitalization.

Input Description

The first line of input will be the number of sentences to translate. Each given sentence will begin with a capital letter and end with a period. Sentences will contain no other punctuation and all other letters will be lowercase.

Sample Input

3
Youre a hairy hero.
World domination ftw.
Tomorrow ill buy some rats from the pet shop for my lovely snakes to eat.

Sample Output

ureyoss ssa iryhass rohess
rldwoss minationdoss ftwess
morrowtoss lliss ybuss mesoss tsrass mfross thess tpess pshoss rfoss myess velyloss kessnass toss atess

The Joker's Money Pyramid

- **Input File**: jokermoneypyramid.in
- **Output**: console

Problem Description

The Joker needs help creating his pyramid of cash to burn for his T.V. broadcast so that he may threaten Batman. However, he is suffering from extreme O.C.D., so his money must be organized in a specific pattern. He plans to draw out ways in which he may stack his bundles.

The top of the money pyramid will always be the smallest money bundle. When making a new row of money bundles, the edges will be made with the smallest money bundles. Each non-edge bundle will be the result of multiplying the two bundles above it and adding 1.

Input Description

- The smallest money bundle he has $3
- The first number in the file will be the number of money piles to be built
- Each additional line will be the number of rows for a money pile
- Note: You will not be counted off for alignment as long as each row has the correct values and those values are separated by at least 1 space

Sample Input

```
2
3
5
```

Sample Output

```
    3
   3  3
 3 10 3
       3
      3   3
    3  10  3
  3  31  31  3
3 94 962 94 3
```

Chevron Coat of Arms

- **Input File**: chevron.in
- **Output**: console

Problem Description

Your job is to reward accomplished U.S. military veterans with chevrons in order to indicate that they are the heroes of our country. Based on the rank of the officer, you will create an insignia with a specific size and number of chevrons.

Notes

A lance corporal receives one chevron, a corporal two, and a sergeant three.

Each chevron will contain a dot at its top, sides made with slashes, and the bottoms made with hyphens. Each wall piece has 2 slashes separated by 2 spaces and the bottom of every wall consists of 4 hyphens.

Chevrons will vary in size based on the number of spaces between the dashes and slashes. Multiple chevrons in one insignia (i.e. corporal and sergeant) will be identical.

The size will represent the distance between the bottom hyphens. The first walls will be (size-4) apart and the second walls will be (size-2) apart. The range for the size will be from 5 to 10.

Input Description

The first integer will indicate the number of insignias you must create. Each following line will contain the rank of the officer and the size of the chevron(s).

Sample Input

```
2
lance corporal 10
sergeant 7
```

Sample Output

```
     .
 / /   \ \
/ /     \ \
----      ----


     .
 / / \ \
/ /   \ \
----  .  ----
 / / \ \
/ /   \ \
----  .  ----
 / / \ \
/ /   \ \
----     ----
```

The DNA Notes

- **Input File**: dna.in
- **Output**: console

Problem Description

Dr. Osborn found the DNA code for one strand of lab spiders that is saving people from his notorious plans. Osborn, who is bad at programming, comes to you to help him make a program which can find the complementary strand of the DNA and gives you the notes that are listed below.

Notes

1. A is complement to T
    a. Change the given A's to T's
    b. Change the given T's to A's
2. C is complement to G
    a. Change the given C's to G's
    b. Change the given G's to C's

Input Description

1. The first number is the number of DNA codes that needs to be processed.

Sample Input

3
ACTGCGAT
CATGCATGCA
CGATGACTTGAC

Sample Output

TGACGCTA
GTACGTACGT
GCTACTGAACTG

Mole Robber Notes

- **Input File**: mole.in
- **Output**: console

Problem Description

Mole Man has been tunneling into banks vaults for his robberies and it is up to Mind Master to catch him. Mind Master has placed seismic sensors throughout the city to help him hunt down Mole Man.

He has come to you to build a program that will tell him what area of the city he should search. He needs the program to shade in the area to be search based on the coordinates of the 3 sensors that were set off.

Notes

- The city is represented with a 10 by 10 grid.
- '-' indicates an area that does not need to be searched.
- '*' indicates an area that needs to be searched
- Separate each solution by a blank line
- Sensor locations will always be given as ints
- When calculating search locations, truncate any digits beyond the decimal

Input Description

Each additional line of data will contain the 3 senor locations

- The sensor locations will be in the following format: (x1,y1) (x2,y2) (x3,y3)

Sample Input / Output is List on the Next Page

Sample Input

(4,5) (2,3) (6,3)
(2,4) (7,1) (7,8)
(1,1) (6,3) (7,6)

Sample Output

```
----------
----------
----------
--*****---
---***----
----*-----
----------
----------
----------
----------

----------
-------*--
------**--
----****--
--******--
---*****--
----****--
-----***--
-------*--
----------

----------
-*--------
--**------
---****---
----***---
-----**---
-------*--
----------
----------
----------
```

The Ultimate Showdown

- **InputFile**: battle.in
- **Output**: console

Problem Description

It looks like our Superheroes have run upon a band of Supervillains! Find out if the Superheroes should engage and capture the wrongdoers or avoid them and flee.

Since the Superheroes are the ones initiating the fight, they get pick their opponents. Each character will have different Attack Powers and Health. When a character's health reaches zero, that character is knocked out. When a hero and villain fight, each reduces his/her health by their opponent's attack power. *Attack Power will affect health in a 1:1 ratio*.

If characters still have health at the end of the first turn, a second turn will start, beginning with the Superheroes move again (Attack power gets restored, health does not). Engagement is dependent on whether you can defeat all the villains and still have at least one hero still conscious.

Note:

Each round one hero will fight a villain. If the number of combatants is unequal than some heroes or villains have to sit out for that round.

Input Description

The first line of input will tell you how many heroes / villains are on each side. At the start of the battle the number of heroes and villains will always be the same. The following lines will contain a name followed by a comma and two numbers separated by a space. The two numbers will be the Attack Power and Health, respectively.

Sample Input

```
3
Mario, 35 22
Luigi, 30 25
Peach, 5 5
Bowser, 50 40
Hammer Bro., 15 10
Big Boo, 30 20

3
Superman, 999 1500
Batman, 650 1000
Spiderman, 350 500
Magneto, 750 1500
The Joker, 450 750
Dr. Octopus, 450 1250

2
Megaman, 50 40
Simon Belmont, 40 45
Krang, 50 10
Ganon, 90 100
```

Sample Output

```
Yes! The heroes will be victorious!
Yes! The heroes will be victorious!
No! Avoid at all costs!
```

Sewer Maze

- **Input File**: sewer.in
- **Output**: console

Problem Description

Spider-man is lured into the sewers of New York by the aroma Lizard's lunch. Spider-man will tell you the layout of the sewer and you have to print out the directions he must follow to reach the meal in the least number of moves, on a single line. The following line he wants the directions out in the least number of moves.

Notes

- Walls – 'X'
- Empty – 'O'
- Entrance – 'S'
- Lunch – 'L'
- Spiderman can only move North 'N', East 'E', South 'S' and West '

Input Description

The first line of the input file will consist of a single integer that represents the number of data sets to solve. For each data set, the first line will consist of two integers, 'r' and 'c' that represent the number of rows and columns for the sewer, respectively.

The next 'r' lines will be 'c' characters in length and represent the maze.

Sample Input

```
2
5 7
XXXXXXX
XSOOOOX
XXOXXOX
XXXLOOX
XXXXXXX

9 9
XXXXXXXXX
XOOOOSOOX
XOOOXOXXX
XOOXXOOOX
XOXXXOXOX
XOXLXOXOX
XOOOOOXXX
XOOOOOXXX
XXXXXXXXX
```

Sample Output

```
EEEESSWW
EENNWWWW

SSSSSWWN
SEENNNNN
```

Star Bomb

- **Input File**: starbomb.in
- **Output**: console

Problem Description:

Ironman's robot army is getting destroyed by Dr. Evil's minions. Iron man has developed some new missiles that he could launch to any place in the battle. Turn the tide of battle by designing a program that provides the co-ordinates to launch the star-bomb missiles at. Determine the position that will have the most "points." When displaying the location to attack use the following format: (column, row)

Input Description:

- The First integer is the number of inputs
- Before each input, there is a second integer, which is the size of the starbomb
- The map is always 10 by 10 and will contain the following symbols:
  - 'H' – Hero commander: -3 points
  - 'B' – Evil Commander: +3 points
  - 'h' - Hero bot: -1 point
  - 'b' - Evil minion: +1 point
  - '-' - Empty

Star patterns are as follows:

1:

-	-	-
-	X	-
-	-	-

2:

X	X	X
X	X	X
X	X	X

3:

X	-	X	-	X
-	X	X	X	-
X	X	X	X	X
-	X	X	X	-
X	-	X	-	X

Sample Input:

2

1

-	-	h	-	-	b	-	-	-	-
-	-	-	-	-	-	-	-	-	-
-	-	b	-	-	-	h	-	-	-
-	-	-	-	-	H	-	-	-	-
-	-	b	-	b	H	-	-	-	H
B	-	-	-	h	-	-	-	-	-
-	-	b	-	-	-	b	-	-	-
-	b	b	-	-	-	-	-	h	-
-	-	-	-	-	b	-	-	-	-
-	-	-	-	b	-	-	-	-	-

3

-	-	h	-	-	b	-	-	-	-
-	-	-	-	-	-	-	-	-	-
-	-	b	-	-	-	h	-	-	-
-	-	-	-	h	h	-	-	-	-
B	-	b	h	B	H	-	-	-	H
-	b	b	-	h	-	-	-	-	-
-	b	b	-	-	-	b	-	-	-
-	b	b	B	b	-	-	-	h	-
-	-	-	h	-	b	-	-	-	-
-	-	-	b	b	-	-	-	-	-

Sample Output:

(0,5)

(2,6)

Lifesteal Leaperman

- **Input File**: leaperman.dat
- **Output**: console

Problem Description

Leaperman has a special superpower—every time he hurts an opponent in battle, the distance he can leap increases.

However, his leaping distance does not increase at a constant rate. It starts out at 0 meters, then increases to 1 after scoring one hit, and then to 3 after scoring another.

After that, his new leaping distance becomes a sum of the last three distances. So if Leaperman hurt his opponent 3 times, his leaping distance would be 4. If he hurt them 4 times, his distance would be 8, and so on.

Create a program that will display the number of hits he made, based on his leaping distance.

Input Description

The input will have int *n* on the first line, indicating the number of jumps, and then *n* lines after that. On each line after the first, there will be one int, representing Leaperman's leaping distance.

Output Description

The output should print out the numbers of hits scored before a jump, each on a different line.

Sample Input

```
4
1
9812284
0
1052
```

Sample Output

```
1
27
0
12
```

The Joker Got Away

- **Input File**: batman.in
- **Output**: console

Problem Description

Recently Batman was unable to catch the joker, due to not being able to reach the scene of the crime fast enough.

He is getting ready to test multiple new jet fuels and needs a program to calculate which fuel gives him the best acceleration. For each test he will record the time, starting velocity and ending velocity.

Your program will need to take the inputs from each test and display the acceleration gained from each fuel and which fuel was the best.

Notes:

The formula for velocity is:

$$\bar{a} = \frac{v - v_0}{\Delta t}$$

a = acceleration

v = final velocity

v_0 = initial velocity

Δt = change in time

Round each acceleration to the nearest thousandth place.

Input Description

Each line of data will be for a different fuel in the following format:

(change in time, starting velocity, ending velocity)

Sample Input

(5, 39.0, 800.0)

(30, 20.0, 7000.0)

Sample Output

Fuel 1 152.200

Fuel 2 232.667

Fuel 2 has the best acceleration

Hideout Fees

- **Input File**: hideout_fees.in
- **Output**: console

Problem Description

Hid-a-tron is a company that helps villains getaway when they are being pursued. They hide villains for up to 24 hours and move them to a new hideout at midnight.

The company charges 1 energy bar for the first two hours they hide you, with a minimum of 2 hours. For each addition half hour they charge 1 energy cube. The company guarantees to have things clean for your getaway with in twelve hours, as part of this guarantee they do not charge for additional time after 12 hours. If you are still hiding you at midnight you are charged 13 energy cubes for moving you to a new hideout.

The company has hired you to build a program to calculate hiding fees. The program will take in two military times. When the first time is greater than the second time, it means the hiding was carried into the next day. The receipt will need to list how long the villain was hidden, if they had to be moved and the total charge.

When the minutes hidden are greater than 0 and less than 30, the minutes will be rounded to 30. When the minutes are greater than 30, the minutes will be set to 0 and the hours will be rounded up by one. The fee must be listed in as few cubes as possible.

Notes

- Hid-a-tron has a minimum charge of 2 hours
- The first two hours costs 1 energy bar
- Each half hour is 1 additional energy cube
    - 1 to 29 minutes are rounded to 30 minutes
    - 31+ minutes are rounded to the next hour
- Hid-a-tron does not charge for more than 12 hours of hide time
- The fee for a midnight move is 13 energy cubes
- 5 energy cubes make 1 energy bar
    - Convert as many cubes to bars as possible
- The receipt will be in the following format:
    - hh:mm (r)- b.c
        - hh – charged hours
        - m – charged minutes
        - r – moved y or n (y will stand for yes and n will stand for no)
        - b – bars charged
        - c – cubes charged

Input Description

The first line of input will be the number of villain fees to be calculated. Each next will contain two military times in the format hhmm, separated by a space. The first time will be when the villain arrived and the second time will be when the villain left.

Sample Input

3
1131 1745
2000 0500
0305 2302

Sample Output

06:30 (n) – 2.4
09:00 (y) – 6.2
12:00 (n) – 5.0