UTD Programming Contest for High School Students
November 5, 2011

- **Time Allowed: three hours.**
- **Each team must use only one computer - one of UTD's in the main lab.**
- **Answer the questions in any order.**
- **Use only Java, C, C++ or C#.**
- **Your program source code must be contained in one source file.**
- **Do not use the "package" construct in your Java code.**
- **Your programs must read from the named file specified in the problem header, and must output to System.out (cout for C/C++ programmers.)**
- **Do not access the web except to access Java or C/C++/C# documentation.**
- **Do not use any recording device containing code, other than UTD's computer.**
- **Your solutions must be entirely yours, typed by you during the time allowed for the contest.**
- **As soon as you have solved a problem, submit ONLY the source file via your `PC^2` client.**

Scoring

Equal points will be awarded to each question, even though they may not be equally difficult.

In order to break ties, we will use penalty points. The penalty points for a question will be zero if the question is not answered correctly. If a correct submission occurs for a question at time T minutes, then T penalty points will be added, plus 20 penalty points for each incorrect submission for that question.

Remember, remember the 5th of November,
Gunpowder, treason and plot,
I see no reason why gunpowder, and treason,
Should ever be forgot.

A. Guy

Input File: A.txt
Runtime allowed = 3 seconds

In 1604 Guy Fawkes became involved with a small group of English Catholics, led by Robert Catesby, who planned to assassinate Protestant King James 1^{st}, replace him with his daughter, Princess Elizabeth, and restore Catholic rule in England. They packed the undercroft (basement) of the House of Lords Parliamentary Building with barrels of gunpowder and planned to blow up the building when the King made one of his rare visits on the evening of November 5th. Fawkes was discovered in the undercroft with a slow fuse and a match early in the morning of November 5th.

On the days leading up to Guy Fawkes Day, November 5th, English children construct effigies of Guy out of old clothes stuffed with straw or paper, adding a plastic mask for a face. These 'Guys' are carried around in strollers. The children ask everyone they meet, "Penny for the Guy?"

The money collected is used to buy fireworks. On the evening of November 5th, bonfires are lit and poor old Guy is burned. The children let off their fireworks, preferably supervised by parents. One of the most popular fireworks is the "Banger" (called a firecracker in the USA).

How many Bangers can I buy?

Input
The input will be made up of a sequence of datasets, each one on a single line containing two integers separated by a space. The first integer gives the amount of money collected and the second gives the price of a single Banger. This list of datasets is terminated by End Of File.

Output
For each dataset, print the number of Bangers that can be bought.

```
Sample Input      Sample Output
-----------------|--------------
135 20           |6
10  15           |0
10001 57         |175
```

Guy was not burned. He was supposed to be hanged, drawn and quartered, but he jumped from the scaffold before the executioner could get started. The fall broke his neck.

B. Manhunt.

Input File: B.txt
Runtime allowed = 3 seconds

The Fens are a naturally marshy region in eastern England. Most of the Fens were drained several centuries ago, resulting in a flat, damp, low-lying agricultural region of 1500 square miles.

A crook is on the run on foot in the Fens. Satellite navigation tells us his position as x kilometers to the East and y kilometers to the North of the police station. He is heading due east at v kilometers per hour.

As soon as the satellite data comes in, a policeman on horseback starts from the police station and travels at a constant speed of p kilometers per hour. His trusty iCatchACrook tells him the direction he must travel in order to catch the crook at the earliest possible time. How long will it take to intercept the crook assuming the crook and the policeman keep to constant speeds and fixed directions?

Input
The first line of the input will contain a single integer N giving the number of data sets to follow.

Then, for each data set, a single line of text will be given containing four decimal values, x, y, v, p, where the speed of the crook will be less than the speed of the policeman, $(0.00 \leq x, y \leq 100.00), (-6.00 \leq v \leq 6.00), (10.00 \leq p \leq 25.00)$.

Output
For each dataset, output the minimum travel time of the policeman in hours accurate to two decimal places. Print two figures after the decimal point.

```
Sample Input            |Sample Output
------------------------|--------------
3                       |4.24
20.0 10.0 5.0 10.0      |1.11
10.0 10.0 3.0 15.0      |0.00
0.0 0.0 5.0 10.0        |
```

C. Parsing the Teams

Input File: C.txt
Runtime allowed = 3 seconds

After the teams have registered for a contest, the contest organizer has the chore of producing a list of teams to be input to the PC2 contest system. Write a program to automate a simplified version of this process.

Input
The input will be the data file produced by the registration system. The file contains records, one for each high school. Each record is eight lines of text. The first line contains the name of the high school. The second line contains the teacher's name. Then follows six team-lines, some of which could be blank. Each non-blank team line contains a string of characters giving the team name, then a space, and then a single character that is either 'A' for Advanced or 'N' for Novice. The team name can contain spaces and any printable characters, except '|'. The name must not contain tabs, newlines, or other formatting characters.

Here is an example of one record:

```
Newtown High School
Teach Totaller
Giants A
Don't mess with Texas A
The Space Characters N
Suitably Numb N

Best Friends; A
```

The input will be terminated by a school name of "END".

Output The first line of the output will be the following string:

```
site|account|group|displayname
```

Then, for each non-blank team name, output the string:

```
1|teamN|Advanced|teamName (School name)
```

where the site is always 1, the account progresses through "team1", "team2", etc. and the group is either "Advanced" or "Novice". The display name is made up of the team name, then a space and then the school name in parentheses.

```
Sample Input                      |Sample Output
----------------------------------|-----------------------------------------------
Newtown High School               |site|account|group|displayname
Teach Totaller                    |1|team1|Advanced|Giants (Newtown High School)
Giants A                          |1|team2|Advanced|Don't mess with Texas! (Newtown High School)
Don't mess with Texas! A          |1|team3|Novice|The Space Characters (Newtown High School)
The Space Characters N            |1|team4|Novice|Suitably Numb (Newtown High School)
Suitably Numb N                   |1|team5|Advanced|Best Friends; (Newtown High School)
                                  |1|team6|Advanced|The Ones and Zeroes (Foxtail Academy)
Best Friends; A                   |
Foxtail Academy                   |
Mr. Teacher                       |
The Ones and Zeroes A             |
                                  |
                                  |
                                  |
                                  |
                                  |
END                               |
```

D. The Last Line?

Input File: D.txt
Runtime allowed = 3 seconds

The expression, "As Black as Newgate's Knocker," is a common colloquialism in England. It originated with the black door knob and knocker on the huge prison gate at Newgate Prison, and is used to signify a very poor outcome - "His chances of acquittal are as Black as Newgate's Knocker." Very few who entered that gate would leave freely. The prison in various forms existed from 1188 to 1904. (This part is true. The rest is a dastardly fiction devised to punish programming contestants.)

Once each month condemned prisoners were lined up inside the prison courtyard. A circular disk was hung around each prisoner's neck. The disks were black on one side and white on the other. They were initially set so that the black side was visible. Each prisoner was given a different number in [1,N] for N prisoners in the line.

The warden would then call out the numbers, 1,2,3,...,N. When he called out the number P every prisoner whose number evenly divided by P would flop his/her disk over. A guard stood by ready to shoot any prisoner whose arithmetic was 'faulty'. After the warden had finished calling out the numbers, those prisoners with white discs facing outward would be freed. The others would be led through the famous prison gate to the scaffold in the square outside.

How many prisoners will be freed?

Input
The input will comprise a sequence of values of N, $1 \leq N \leq 100$, the number of condemned prisoners for that month. This list will be terminated by a value of zero for N. Do not process this terminal line.

Output
For each value of N print a line containing a single integer equal to the number of prisoners freed.

```
Sample Input          Sample Output
--------------------|--------------
6                   |2
87                  |9
0                   |
```

E. Cannonballs

Input File: E.txt
Runtime allowed = 3 seconds

Just seven years after the Guy Fawkes affair, Johannes Kepler conjectured that the densest way to stack cannonballs is in a pyramid. It took 400 years for a proof to emerge and the proof is 300 pages long.

University of Pittsburgh mathematician Thomas C. Hales, and his former graduate student, Samuel P. Ferguson, published their proof in 2006.

It is common to see stacks of cannonballs in castes and fortresses all over the world. In one arrangement a base is made of an equilateral triangle full of cannonballs, similar to the arrangement of pool balls at the start of a game. Layers are built on this foundation such that each cannonball in a certain layer touches three cannonballs in the layer beneath it. Each layer has as many cannonballs as possible that satisfy this touching rule.

In a second arrangement the base is made of a square and each cannon ball touches four others in the layer beneath it. In both structures a pyramid is formed. Given the type of arrangement and the number of layers, how many cannon balls are in the stack?

Input
The input will begin with an integer N on a line by itself giving the number of datasets to follow. Each dataset will comprise one line of text beginning with a string, "SQUARE", or "TRIANGLE", followed by a space and then the integer Q, $1 \leq Q \leq 500$. The string indicates the base type and the integer Q indicates the number of layers.

Output
For each dataset output one line of text of form "The number of cannonballs is X" where X is the answer.

Sample Input	Sample Output
3	The number of cannonballs is 4
TRIANGLE 2	The number of cannonballs is 5
SQUARE 2	The number of cannonballs is 171700
TRIANGLE 100	

F. Tunnels

Input File: F.txt
Runtime allowed = 3 seconds

Assume that under the city of London the sewer tunnels are equally spaced 1000 ft apart and run east-west and north-south The tunnels intersect every 1000 ft, but some intersection points are blocked to humans. Other intersections have ladders and manhole covers above them and can be used to escape to street level and freedom.

Assuming that an escapee always chooses directions that increase his straight-line distance from the start point, find the fastest route to an escape exit.

Input
The first line of the input will contain a single integer, N, giving the number of datasets to follow. Each dataset will be in the following format:
The first line will contain one integer P, $1 \leq P \leq 20$, where there are P east-west tunnels and P north-south tunnels. The intersections are numbered 1,2, ..., P*P. Here is an example layout for P=3:

```
1 - 2 - 3
|   |   |
4 - 5 - 6
|   |   |
7 - 8 - 9
```

The second line of each dataset begins with an integer B, $B \geq 0$, giving the number of blocked intersections, and is followed by B space separated integers giving the index numbers of the blocked intersections.

The third line begins with an integer S, $S > 0$, giving the number of safe intersections where exits lead to the street level and safety. This is followed by S space-separated integers giving the index numbers of the safe intersections. A safe intersection will not also be a blocked intersection.

Output
For each dataset, output a single line giving the minimum distance in feet that an escapee must travel to get from intersection 1 to a safe exit, OR the line "RATS!" if no route to a safe exit exists.

```
Sample Input    |Sample Output
----------------|-------------
2               |3000
3               |RATS!
2 5 6           |
2 8 9           |
2               |
2 2 3           |
1 4             |
```

G. A-Mazing

Input File: G.txt
Runtime allowed = 3 seconds

Hampton Court Palace was built in 1529 and is located upstream of Central London on the River Thames. The Hampton Court Maze is a hedge maze planted some time between 1689 and 1695. The maze covers a third of an acre and contains half a mile of paths. It is a "Simply Connected Maze".

A simply connected maze can be solved by moving forward while keeping your right hand on the righthand wall. A robot has been trained to follow this strategy. You are going to test the robot's logic by simulating its behavior in several mazes. Each maze will have an entrance next to the top left corner and an exit next to the bottom right corner. The floor inside of the maze is made up of squares. Here is the robot's algorithm:

```
Enter the maze. You will be facing south.
Repeat the following until you exit the maze or your battery runs out.
   if(you are in the square next to the exit)
        leave the maze;
   else {
         if(the square to your right is not occupied by a wall)
                turn right and move into that square;
         else if(the square ahead of you is occupied by a wall)
                turn left on the spot;
         else
               move forward one square;
        }
```

The wall is indicated by '#' characters, the entrance is indicated by an 'N' character next to the top left corner, and an 'X' character next to the bottom right corner shows the exit.

The righthand figure below shows the robot's trajectory: N,1,2,3,4,...,9,a,b,c,...,o,X

```
Input outline       Robot's trajectory

#N#####      N      #N#####
# #   #      ^      #1#   #
# ### #      |      #2### #
#     #      |      #34567#
##### #             #####8#
#   # #             #hgf#9#
#   # #             #i e#a#
#     #             #j dcb#
#     #             #klmno#
#####X#             #####X#
```

The robot could visit the same square more than once. Your job is to calculate the number of moves that the robot makes, 25 in the above example, including one to enter and one to exit.

Input
The first line of the input will contain a single integer N giving the number of maze outlines to follow. Then, for each outline, the following sequence of lines will be given:

Start Line: a line **START m n** will be given, where the layout of the maze is m characters wide and n characters tall (including symbols for the walls), ($3 \leq$ m,n$\leq$40).

Layout: n lines will follow, each comprising m characters showing the outline of a maze. '#' characters indicate walls. The 'N' character represents the entrance and the 'X' character represents the exit.

End Line: a line containing the string **END**.

Output For each maze output a single integer on a line by itself giving the number of moves that the robot made OR, if the robot moved more than 100 times, output the line, "Get me that Bunny!".

```
Sample Input      |   Sample Output
------------------|-----------------
2                 |25
START 7 10        |44
#N#####           |
# #   #           |
# ### #           |
#     #           |
##### #           |
#   # #           |
#   # #           |
#     #           |
#     #           |
#####X#           |
END               |
START 10 8        |
#N########        |
#        #        |
#  ##### #        |
####     #        |
#  ##### #        |
#  #     #        |
#    #   #        |
########X#        |
END               |
```