OHS Programming Competition Problems

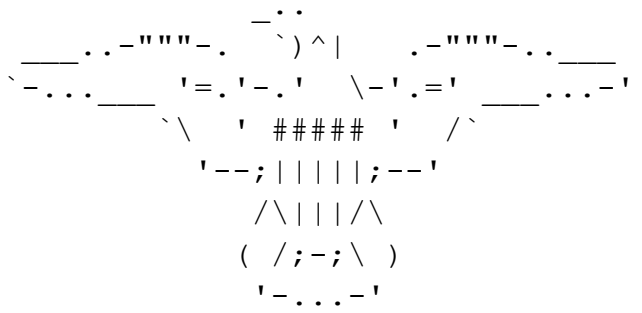
Rules

- Do not turn the page until the competition starts
- Do the problems in any order you choose
- Problems are worth varying points based on difficulty
- Incorrect answers decrease a problem's value by 1/10th
- Ties will be broken based on time
- Programs will be judged based on output
- All input will be read through standard input

Problem 1: Modern Art

(10 pts)

Description: Show your patriotism by printing this eagle.
Input: None
Output: Replicate this ascii “eagle”.



Problem 2: Name and Profession

(10 pts)

Description: It’s bring your parent night at the local elementary school, and you need to help the teacher make nametags for all the parents. Each name tag should have the person’s name and job title.

Input: The first line will be an integer telling you how many parents are attending. Each following line will be a person's name followed by a line containing their job.

Output: For each name and profession output "Hello my name is name and I am a profession" **(Bonus points if you use correct Grammar)**

Example Input:
4
Bob
Plumber
Jane
Electrician
Jerry
Chef
Alice
Programmer

Example Output:
Hello my name is Bob and I am a Plumber
Hello my name is Jane and I am **an** Electrician
Hello my name is Jerry and I am a Chef
Hello my name is Alice and I am a Programmer

Problem 3: Basic Math

(20 pts)

Description: Jessie just produced 20 lbs of blue rock candy. You're working as his distributor, and you need to write a simple two number calculator so you can figure out your profit margins.

Input: The first line will be an integer telling you how many shipments Jessie has sent. For each shipment there will be three lines of input. The first line will contain the command for the calculator, and the other lines will be the operands. The calculator commands will always be lowercase.

Output: A number for each shipment, rounded to at least 1 decimal point, which is the result of the calculation.

Example Input:

```
4
add
4
5
divide
3
7
multiply
2
3.5
subtract
4
8
```

Example Output:

```
9
0.428571
7
-4
```

Problem 4: If You Say So…

(20 pts)

Description: You're a contractor for the NSA and you're working on censoring some sensitive government documents. Help the NSA find sentences that contain information about their secret project with the codename "say".

Input: The first line will be an integer telling you how many sentences you are searching through. The following lines will be sentences that may or may not contain the word “say”.

Output: If the line contains the word “say” output it to the screen. If not output “NOT RELEVANT”.

Example Input:
5
I want to eat some pie.
What did you say?
I wish you would say you love me.
I already said that.
Did you hear about that secret organization “say”?

Example Output:
NOT RELEVANT
What did you say?
I wish you would say you love me.
NOT RELEVANT
Did you hear about that secret organization “say”?

Problem 5: Will I make it?

(20 pts)

Description: Han Solo is smuggling for Jabba the Hutt. Unfortunately times are tough, and he’s not sure if he has enough fuel to make it back to Tatooine.

Input: The first line will be an integer which represents the number of trips Han has to make. For each trip, the first line

will be the distance in parsecs, followed by a line containing his mileage in parsecs per liter of hypermatter. The last line for each trip will be the liters of hypermatter in his fuel cell.

Output: If Han has enough fuel to return to Tatooine output "Made it". If he is lacking in fuel output "Lost in space"

Example Input:

```
4
1.2
3.6
4.23
300
.5
540
13
43
15.6
5.2345
1
76
```

Example Output

```
Made it
Lost in space
Made it
Made it
```

Problem 6: Recurse my Sequence

(40 pts)

Description: You're given two sequences, and you need to determine the sum of the Nth terms of the two sequences.

Sequence F (The Fibonacci sequence): Start with the integers 1 and 1. The next term in the sequence is: $F_{N-1} + F_{N-2}$

1 1 2 3 5 8 13 21 34 etc...

Sequence G: Start with the integers 1, 5, 9. The next term in the sequence is calculated from this equation: $2*(G_{N-1})-6*(G_{N-2})+3*(G_{N-3})$

1 5 9 -9 -57 -33 249 525 ...

Input: Each line will contain two integers, n and m, separated by spaces. M and N will always be less than 50.

Output: For each line output the sum of the nth term of F and the mth term of G

Example Input:
5 11
3 8

Example Output:
-2140
527

Problem 7: Will It Blend?

(30 pts)

Description: You are working for Blendtec as a quality control agent and have to determine which items can be blended by which blenders.

Input: The first line will contain an integer telling you how many items there are. Each item will have a name and a blend-difficulty separated by a space. After the items there will be a line with an integer telling you how many blenders there are. Each blender will have a name and blend-level separated by a space. After the blenders there will be a line with an integer

telling you how many pairs there will be. Each pair will contain the name of an item followed by the name of a blender.

Output: For each pair of item and blender output "It blends!" if the blend-level is **greater than or equal** to the blend-difficulty, and "It doesn't blend..." if the blend-difficulty is **greater than blend level**.

Example Input:

```
4
BANANA 2
ORANGE 3
APPLESAUCE 1
ROCK 10
3
SUPERBLENDER 11
BADBLENDER 2
BLENDERMASTER 5
4
BANANA BADBLENDER
ROCK BADBLENDER
ROCK SUPERBLENDER
ORANGE BLENDMASTER
```

Example Output:

```
It blends!
It doesn't blend...
It blends!
It blends!
```

Problem 8: Kung-Fu Fighting

(30 pts)

Description: You are at your local-island kung-fu tournament and are trying to win some money by betting on who will win. Luckily you know how strong each of the competitors is so you can write a program to help you!

Input: The first line will be an integer telling you how many pairs of fighters there are to start. Each pair of fighters will have the first fighters name, followed by his strength level, followed by the second fighters name, followed by his strength level.

Output: The name and final strength of the winner of the tournament.

Tournament Rules: For each pair of fighters the winner will be determined by whoever has the highest strength level. The winner of each match will then subtract half of the losers strength from his own strength then proceed to "fight" the winner below or above him(imagine a tournament bracket).

Example Bracket:

```
Bruce(5.25) -> winner
^
Bruce(6)        vs        David(1.5)
^                         ^
Bruce(10) vs Chuck(8)     David(2) vs Brennan(1)
```

Example Input:

```
4
Bob 180 Jerry 181
Joe 10 Howy 100
Norman 1 George 70
David 2 Brennan 1
```

Example Output:

```
George 19.25
```

Problem 9: Mc'Donalds Dieting

(70 pts)

Description: You are on a diet but your friends decide they want to go to McDonalds for lunch. You don't want your friends to think you are on a diet though so you want to eat as much as you can while staying equal to or under your calorie limit, as long as you **don't eat more than one of the same item.**

Input: The first line will be an integer representing the number of data sets. For each data set the first line will be the maximum number of calories you can intake for your diet. Each line afterwards will be the name of a food item on the menu followed by how many calories it has. Each data-set will be ended by a "0"

Output: For each dataset output the maximum number of calories you can consume (eating items on the menu) without exceeding the calorie limit on one line, how many items you eat to achieve

that calorie intake on a second line, and every item you would be eating in the order to reach that calorie intake on a third line. Separate each dataset by a blank line.

Sample Input:

```
2
200
Pizza 100
Pie  500
Toast 50
Juice 70
0
1000
Hamburger 2000
Soda 400
Fries 800
Toast 80
Butter 600
0
```

Sample Output:

```
170
2
Pizza, Juice

1000
2
Soda, Butter
```

Problem 10: Hidden Treasure

(20 pts)

Description: Johnny Depp has decided to become a real pirate and has set out to find some treasure of his own. Unfortunately he is really bad at reading maps so you need to help him out by telling him the X and Y coordinates of the treasure on his treasure map.

Input: The first line will contain the width and length of the treasure map(in that order) separated by a space. A "Treasure Map" made up of lines of the character "H" with an "X" representing the location of the treasure.

Output: The x and y coordinates of the treasure(The bottom left of the map is 0,0) formated as shown.

Example Input:

```
6 5
HHHHHH
```

HHHHHH
HHHHHH
HHHHXH
HHHHHH

Example Output:
X: 4
Y: 1

Problem 11: What a square!

(20 pts)

Description: Your friend really wants you to write a program that draws squares. So do it.

Input: The first line will be an integer telling you how many squares to draw. For each square you will be given a side length greater than 1 on a separate line.

Output: Draw each square with the character "S" as the border. Separate the squares with blank lines underneath them. (The squares will not look like squares. Just make sure there are the correct number of "S" characters)

Example Input:
3
2
6
4

Example Output:
SS
SS

SSSSSS
S   S

```
S    S
S    S
S    S
SSSSSS

SSSS
S  S
S  S
SSSS
```

Problem 12: Code Obfuscation

(40 pts)

Description: A popular game development studio has decided to release their source code. They wouldn't want to make things too easy for potential modders though and want to replace certain variables in their programs with different words.

Input: The first line will contain an integer representing how many variables there are. For each variable there will be a line with the variable name and the name it should be replaced with separated by a space. After all the variables there will be an indeterminate amount of lines of code that go until you reach a "0".

Output: For each line of code replace the name of whatever variables you read originally with their replacement name, then print the obfuscated line of code. **Names are case sensitive**

Example Input:

```
3
name x0fd
age 8yTu
person defRe
name = 12;
if (age == 22) { cout << "you are old" << endl; }
Person person = new Person();
0
```

Example Output:

```
x0fd = 12;
if (8yTu == 22) { cout << "you are old" << endl; }
Person defRe = new Person();
```

Problem 13: Factory Optimizer

(70 pts)

Description: The chip manufacturing company you work for has fallen on hard times recently, and your boss wants you to write a program to optimize the assembly line. Your program controls a robotic arm which picks circuit boards up off a conveyor belt and packages them in a box for shipping. Unfortunately the arm is slow and every time it picks up a board, the next chip passes by on the belt (it can't pick it up). Every box needs one (and only one) chip from each of the three series, and your arm can only pack one box at a time. The series of the chip is signified by the first digit of the serial number (series 1, 2 and 3).

Input: The first line will be an integer which represents the number of assembly lines. Each assembly line will consist of a line of integers which are the serial numbers for each chip coming down the line.

Output: Print out the maximum number of packed boxes for each assembly line.

Example Input:

```
3
1222 14242 3312 32322 22324 234 123123 23 23 13 22 131 123 3
23223 323333 222123 123223 312345 12234234 1241441124
3244 311 33421 32324 3442 22 1191
```

Example Output:

```
2
1
```

Problem 14: The Interpreter

(70 pts)

Description: You're working on writing a compiler for a new programming language you're writing, but development would be speed up considerably if you had an interpreter to test programs live.
The syntax for the language is as follows:
Spacing doesn't matter in the program
All variables are numbers (decimal points are possible).
equality is tested with ==, assignment is =, basic math operations +, -, *, / are supported.
If statements do not require parens or braces ({}).
All if statements are completed with the "end" statement (no quotes), and may contain an optional else:

```
if x == 2
 x = x + 1
 y = 5
else
 y = 3
 x = -1
end
```

Every program ends with a return statement which a one of the variables set earlier in the program.

Input: The first line will be an integer representing the number of programs to follow. The following lines will consist of lines representing the programs.

Output: Output the value which the program returns, use at least one decimal point.

Example Input:

```
2
x = 1
y = 4
z = -1.5
if x == 1
 z = z + 1
else
 z = 0
end
return z

a = 3
fred = 2.5
beta = a * fred
if (beta == 1)
  a = 4
end
if (beta == 2)
 a = 5
end
return a*beta
```

Example Output:

```
-0.5
30
```

Problem 15: The Gambler

(30 pts)

Description: Your friend just lost his parent's college savings in a casino game. The game was based on dice, and you need to figure out if the dice were loaded.

Input: The first line will be an integer representing the number of games to follow. Each game is represented on one line, which contain six integers separated by spaces. The first integer is the number of times one was rolled, the second integer is the number of times two was rolled and so on

Output: If the proportion of any of the dice rolls was more than 10% away from $\frac{1}{6}^{th}$ (a fair dice), print FAIR, otherwise print CHEAT.

Example Input:

```
4
12 3 16 12 12 12
3 3 3 3 3 3
2 4 5 10 1 8
68 73 90 83 72 92
```

Example Output:

```
CHEAT
FAIR
CHEAT
FAIR
```

Problem 16: Tic Tac Told You So

(30 pts)

Description: The latest craze at your school is tic-tac-toe and you want to get in on the action. Write a program that can detect the winning move in a tic-tac-toe game so you can impress

your friends.

Input: The first line will contain an integer telling you how many boards you are given. Each board will be made of lines, X's and O's as shown in the sample input and will contain a situation where X is one move away from winning. Each board will be separated by a blank line.

Output: The X and Y coordinates of the box that X should play in in order to win as formatted in the example output. (Box 0, 0 is the bottom left. X and Y increment to the right and upwards)

Example Input:

```
2
X|  |O
X|  |O
 |  |

X|  |O
O|  |
 |  |X
```

Example Output:

```
(0,0)
(1,1)
```

Problem 17:
Star Wars vs. Star Trek

(30 pts)

Description: A bunch of nerds were having an intellectual debate over who would win if the Star Wars and Star Trek universes battled each-other. Help these nerds come to a decision and end the debate once and for all.

Input: The first line will contain an integer telling you how many star wars characters to read in. For each character there will be a line with the characters name followed by a line with the characters strength. After the star wars characters there will be an integer telling you how many Star Trek characters there are. For each character there will be a line with the characters name followed by a line with the characters strength. After the Star trek characters there will be an indeterminate amount of lines containing battle scenarios. Each scenario will have Star Wars character(s) to the right of a "vs." separated by " + " and Star Trek character(s) to the right also separated by " + ". End of input will be signified with a "0".

Output: For each battle scenario output "Star Wars" if the star wars characters win, "Star Trek" if the Star Trek characters win, or "Tie" if there is a tie. Winning is decided by adding up the strength of all the characters in the scenario from each universe. Whichever side has the most cumulative strength wins, and if their strength is the same then it is a tie.
On a separate line at the end output whichever side won the most scenarios as well as the number of victories they had(as formated in example output). If they each won the same amount output "Tie"

Example Input:

```
2
JARJAR 0
OBIWAN 10
2
KIRK 6
SPOCK 10
JARJAR + OBIWAN vs. KIRK + SPOCK
OBIWAN vs. SPOCK
KIRK vs. OBIWAN
KIRK vs. JARJAR
0
```

Example Output:

```
Star Trek
```

Tie
Star Wars
Star Trek
Star Trek: 2

Problem 19: The Right Triangle

(40 pts)

Description: Your friend is working on the design for a building, and he wants you to help him add some variety to the architecture. Every part of the building is a right triangle, and your friend wants to make construction simple, so he's only interested in right triangles which have integral length sides {a,b,c} (no fractions).

Input: Each line will contain an integer p which represents the perimeter of a right triangle.

Output: For each line output the number of unique right triangles which have perimeter p and integral length sides. Order of the sides does not matter, so triangle {3,4,5} and {4,3,5} are the same.

Example Input:
12
120
840

Example Output:
1
3
8

Problem 20: Substring Codes

(40 pts)

Description: You're writing the code for a new network card to read information as it comes through the network. The network is secured by only sending valid information in numbers where every three digits when interpreted as a separate number are divisible by an ascending prime.
Here is an example using the number 91835728:
Let d_1 be the first digit, d_2 be the second and so on.
$d_1d_2d_3$=918 is divisible by 2
$d_2d_3d_4$=183 is divisible by 3
$d_3d_4d_5$=835 is divisible by 5
$d_4d_5d_6$=357 is divisible by 7
$d_5d_6d_7$=572 is divisible by 11
$d_6d_7d_8$=728 is divisible by 13
If any of the three digit substrings start with a zero, reject the number.

Input: Each line will contain an integer.

Output: The sum of all the digits of the inputted integers which meet the above criteria.

Example Input:
91835728

```
2585
2601
8183572
8083506
```

Example Output:

```
97
```