Computer Science Competition
Invitational A 2023
Programming Problem Set

I. General Notes

1. Do the problems in any order you like. They do not have to be done in order from 1 to 12.

2. All problems have a value of 60 points.

3. There is no extraneous input. All input is exactly as specified in the problem. Unless specified by the problem, integer inputs will not have leading zeros. Unless otherwise specified, your program should read to the end of file.

4. Your program should not print extraneous output. Follow the form exactly as given in the problem.

5. A penalty of 5 points will be assessed each time that an incorrect solution is submitted. This penalty will only be assessed if a solution is ultimately judged as correct.

II. Names of Problems

Number	Name
Problem 1	Abhinav
Problem 2	Alan
Problem 3	Bianca
Problem 4	Cho
Problem 5	Damian
Problem 6	Feng
Problem 7	Gael
Problem 8	Himanshu
Problem 9	Justin
Problem 10	Marek
Problem 11	Priyanka
Problem 12	Sunny

1. Abhinav

Program Name: Abhinav.java **Input File: none**

Abhinav is really excited about taking a programming course. He has tried following his teacher's examples but just cannot get his program to produce the exact output required by the assignment.

Can you help Abhinav with this problem?

Input: None

Output: The exact message as shown below.

Sample input: None

Sample output:

```
Ultimate Important Languages
U-I-L
UIL!
```

2. Alan

Program Name: Alan.java **Input File: alan.dat**

In Alan's Integrated Physics and Chemistry class, Alan's teacher has just introduced the concept that temperature is traditionally given in either Fahrenheit, Celsius, or Kelvin. Alan is familiar with both Fahrenheit and Celsius since he has spent time in both America and Europe, but the concept of temperatures in Kelvin is still a bit new and unfamiliar to him. In an attempt to better understand temperatures in Kelvin, Alan has the idea to write a program that will convert a given Fahrenheit temperature to Kelvin. Alan knows the formula for converting from Fahrenheit to Kelvin is:

$$K = 5/9(F - 32) + 273.15$$

Where K is Kelvin and F is Fahrenheit. Alan is having a bit of trouble with the math though, and would like your help completing the program. Think you can assist him with the task?

Input: Input will consist of an integer N, the number of test cases. The number of test cases will be in range [1,20]. Each subsequent line will contain an integer degree F, representing the temperature in Fahrenheit. F will be in range [-250,250].

Output: Each line of output must consist of "`F degrees Fahrenheit is equal to K Kelvin`", where F is the input temperature in degrees Fahrenheit, and K is the converted temperature in Kelvin using the formula above. Output should be formatted to two decimal places after the decimal point.

Sample input:

```
5
32
100
25
212
-5
```

Sample output:

```
32 degrees Fahrenheit is equal to 273.15 Kelvin
100 degrees Fahrenheit is equal to 310.93 Kelvin
25 degrees Fahrenheit is equal to 269.26 Kelvin
212 degrees Fahrenheit is equal to 373.15 Kelvin
-5 degrees Fahrenheit is equal to 252.59 Kelvin
```

3. Bianca

Program Name: Bianca.java **Input File: bianca.dat**

Your friend Bianca has Geometry homework due next period, and she forgot all about it! Quickly write a program to solve Pythagoras theorem before the homework is due.

Use the following theorem (you are given a and b, so solve for c):

$$A^2 + B^2 = C^2$$

Input: The first line will contain a single integer `n` ($0 < n < 100$) that indicates the number of data sets that follow. Each data set will consist of two integers A and B ($0 < A, B < 1000$), separated by a space, denoting the values A and B in the equation above.

Output: Output the value of C ($0 < C < 1000000$) given the A and B values from the input, rounded to 2 decimal places, as shown below.

Sample input:

```
4
3 4
12 14
5 12
5 8
```

Sample output:

```
5.00
18.44
13.00
9.43
```

4. Cho

Program Name: Cho.java **Input File: cho.dat**

You and your friend Cho have an internship at a data lab and they need you to go through sentences and find the letter that occurs the most in each one.

Input: The first line will contain a single integer `n` ($0 < n < 100$) that indicates the number of data sets that follow. Each data set will consist of one line of an unknown number of words(strings) separated by spaces, all appearing on the same line.

Output: Output the alphabetic (A-Za-z) character which occurred in the input line the most, do not include numbers, whitespace, special characters, or anything else that isn't a letter. If there is a tie, just choose the letter that comes first alphabetically, with the ENTIRE capital alphabet coming before the lowercase alphabet (basically sort by ascii values).

Sample input:

```
3
THE HOG GOES TO THE DOG TO CATH THE FROG IN THE BOG
WITCHES MADE MY SOUP, NOW I CAN FLY WHEEEEE
Qwertyuiolkjhgfdazxcvbnm
```

Sample output:

```
O
E
Q
```

5. Damian

Program Name: Damian.java **Input File: damian.dat**

Damian has always been intrigued by numbers. He is absolutely giddy about prime numbers, but honestly, he loves anything to do with factors. He was investigating numbers and he found one type of number that he really liked. These are what he calls "growing" numbers. That is when, looking at the digits from left to right, they increase.

Here are some examples: 123, 38, 1479, 5, 123456789

These are NOT growing numbers: 73, 762, 12557 (notice those two 5s)

Then Damian got an idea, he wants to find the "growing factors" of a number. These are factors that meet the qualifications of being growing. Write a program to help him out. You will make Damian's day.

Example: The factors of 1122 are: 1,2,3,6,11,17,22,33,34,51,66,102,187,374,561,1122
The growing factors of 1122 are: 1,2,3,6,17,34

Input: Input will consist of an integer N, the number of test cases. The number of test cases will be in range [1,20]. Each subsequent line will contain an integer in the range[1,1000000].

Output: Each line of output will be a line of "growing factors" listed in ascending order with a space between each number.

Sample input:

```
5
72
144
1331
621
554433
```

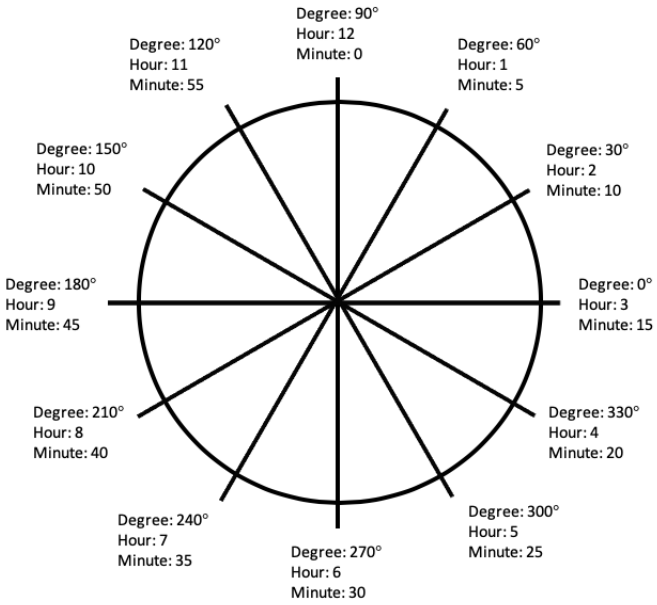
Sample output:

```
1 2 3 4 6 8 9 12 18 24 36
1 2 3 4 6 8 9 12 16 18 24 36 48
1
1 3 9 23 27 69
1 3 159
```

6. Feng

Program Name: Feng.java **Input File: feng.dat**

Feng just learned about the unit circle in her pre-calculus class. As she was leaving class, she realized that the hour and minute hands of an analog clock not only point to their respective hour and minute (representing the current time), but they each also point to a degree in range of [0,359] corresponding to the unit circle representation. (See below figure.) Feng would like your help writing a program that takes in the time, and emits two angles, one for hour, and one for minute, corresponding to the angle (in degrees) each hand would be pointing at.



Input: Input starts with a line containing an integer N, the number of test cases. N will be in range [1,20]. The following N lines contain the time, formatted as HH:MM. HH will be in range [1,12] and MM will be in range [0,59].

Output: For each test case, output the corresponding angle measures each of the two hands would be pointing to.

Hour angle (denoted as `H_Angle` below) and minute angle (denoted as `M_Angle`) will be guaranteed to be in range of [0,359]. Formatting for the output should be "`Hour: H_Angle Minute: M_Angle`"

Note: `H_Angle` and `M_Angle` should be outputted to 1 (one) decimal place.

Sample input:

```
6
08:25
09:18
12:05
03:00
03:01
03:15
```

Sample output:

```
Hour: 197.5 Minute: 300.0
Hour: 171.0 Minute: 342.0
Hour: 87.5 Minute: 60.0
Hour: 0.0 Minute: 90.0
Hour: 359.5 Minute: 84.0
Hour: 352.5 Minute: 0.0
```

7. Gael

Program Name: Gael.java **Input File: gael.dat**

You and your friend Gael have a new art project! You will be given a word and you must make a box design with it, as shown in the sample input and output.

Input: The first line will contain a single integer `n` ($0 < n < 100$) that indicates the number of data sets that follow. Each data set will consist of one word, on its own line, consisting of only capital letters.

Output: Output the shape created using the word given. It will be filled with spaces in the middle and the border will consist of the word multiple times. The corners should match up so that all the words are still spelled correctly, meaning certain words will have to be backwards. The two that will be backwards for this program are the word on the bottom of the box, and the word on the right side of the box. The necessary orientations of the 4 words in the design is as follows: The word on top of the box will go from left to right, the word on the left side of the box will go from top to bottom, the word on the bottom of the box will go from right to left, and the word on the right side of the box will go from bottom to top, see the sample output for clarification.

Sample input:

```
3
HELLO
WORLD
RACECAR
```

Sample output:

```
HELLO
E   L
L   L
L   E
OLLEH
WORLD
O   L
R   R
L   O
DLROW
RACECAR
A     A
C     C
E     E
C     C
A     A
RACECAR
```

8. Himanshu

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

Himanshu's little brother loves to play the ancient game *Moksha Patam*. *Moksha Patam* originated in India in the 2nd century AD and is associated with traditional Hindu philosophy contrasting karma and kama, or destiny and desire. Today, however, the game is most known as either *Snakes and Ladders*, or *Chutes and Ladders*. Aside from its ancient roots, the game is a great tool for helping young children learn to count.

Moksha Patam is a turned based game played on a square grid with varied dimensions, in which players roll a single 6-sided die (1-6) to determine the number of moves the player is to move forward. After the end of a turn, a player may potentially land on either a "S#" signifying a snake (or chute) or an "L#" signifying a ladder. If a player lands on the lower-numbered end of a ladder, the player moves up to the ladder's higher-number square. If the player lands on the higher-numbered square of a snake, the player moves down to the snake's lower-numbered square. The player or players who reach the last square, or surpass the last square, is declared the winner. As this is a turned based game, all players are guaranteed to have the same number of turns, i.e. if Player 1 reaches or surpasses the last square, Player 2 is afforded one last move to try and reach or surpass the last square as well.

The below image shows an example *Moksha Patam* board with dimension 10 x 10:

```
99 98 S3 96 95 94 93 L3 91 90
80 81 82 83 84 85 86 87 88 89
79 S2 77 L3 75 74 73 72 71 70
60 61 62 63 64 65 66 67 68 69
59 58 57 S1 55 54 53 52 51 50
L2 41 42 43 44 45 46 S2 48 49
39 38 37 36 L1 34 33 32 31 30
20 21 22 23 24 25 26 27 28 29
L2 S1 17 16 15 14 13 12 S3 10
00 01 02 03 04 05 L1 07 08 09
```

All players start at the square with index `0`. For formatting purposes for the 10 x 10 board, `0` is denoted as `00` so that each column is straight. Assuming a player is currently at index `00` and roles a 6 from the die, the player would move to index `06`, but this index has `L1` on it denoting this as a ladder. Therefore, the player is then moved to index `35` which is the higher-number squared corresponding to `L1`. If the player were at index `91` and were to roll a 1, the player would move to index `92` and would remain here until the following turn, as this is the higher-numbered latter corresponding to `L3`. If the player was at index `54` and were to roll a 2, the player would move to index `55`. Index `55` however has `S1` denoting this as a snake (chute). Therefore, the player is moved down to index `18` as this is the lower-number square corresponding to `S1`. If a player were at index `43` and were to roll a 4, the player would move to index `47` and would remain here until the following turn, as this is the lower-numbered snake corresponding to `S2`. Although depicted in this board as being on separate rows, snakes and ladders do not necessarily have to be on the different rows. This means that a player could move horizontally only, with no vertical movement on the board.

Unfortunately, with UIL Computer Science season ramping up, Himanshu doesn't have as much time to play *Moksha Patam* with his little brother due to CS practice. In an effort to not only appease his little brother's thirst for *Moksha Patam*, but to get some extra coding practice in as well, Himansu comes up with the idea to write a program that will play *Moksha Patam* against his little brother. The game will only feature two players: Player 1 and Player 2. A series of moves for each player will be given corresponding to the dice roll for that term. As soon as a winner is found and both players have had equal number of turns, the play for the current game is terminated. Can you help Himanshu write such a program?

Input: Input starts with a line containing an integer N, the number of games to be played. N will be in range [1,20]. Each game starts first with an integer D for the dimension of the board. D will be in range [2,32]. Odd dimensions

~ Problem continues on next page ~

Himanshu, continued

for the square board are allowed. This means that the last square to be reach or surpassed is not guaranteed to be at the top left. For formatting purposes all indexes will be outputted to uniform column width, so that each column is straight up and down. The following D lines contain the board for that given game play. Boards use a back-and-forth track from the bottom of the playing area (guaranteed to be the bottom left corner) to the last square. Indexes will be incremented by 1. The board will also have scattered snakes (chutes) and ladders placed throughout the board. Snakes will be represented by `S##` and ladders will be represented by `L##`. Where `##` represents the number of the respective snake or ladder. `##` will be formatted with padded zeros to make sure the formatting for each column remains straight. For every snake or ladder present, it will be guaranteed to have a corresponding end point. It can also be guaranteed that a snake nor a ladder will be present at index `00` nor at the max index of $D^2 - 1$. Following the board are two lines. The first starting with "`P1:`" and the second starting with "`P2:`", each followed by the respected moves for the player. Moves will be in range [1,6] and separated by a comma. The number of moves present may include extra moves that won't be carried out, if a player or players, is found to have reached the max index or surpassed the max index. Last, a line of 36 dashes "`-`" is presented to separate the different game inputs.

Output: For each game played, you are to output either: `Game #CURRENT_GAME_NUMBER: Player 1 wins!`, `Game #CURRENT_GAME_NUMBER: Player 2 wins!`, `Game #CURRENT_GAME_NUMBER: Both players win!`, or `Game #CURRENT_GAME_NUMBER: Neither Player 1 or Player 2 won.` Remember, each of the two players are guaranteed to have the same number of moves, resulting in the potential for two winners.

Sample input: (the new line character is only present after all the moves for a given player have been given)

```
9
10
99 98 S3 96 95 94 93 L3 91 90
80 81 82 83 84 85 86 87 88 89
79 S2 77 L3 75 74 73 72 71 70
60 61 62 63 64 65 66 67 68 69
59 58 57 S1 55 54 53 52 51 50
L2 41 42 43 44 45 46 S2 48 49
39 38 37 36 L1 34 33 32 31 30
20 21 22 23 24 25 26 27 28 29
L2 S1 17 16 15 14 13 12 S3 10
00 01 02 03 04 05 L1 07 08 09
P1:1,4,3,1,5,6,1,5,1,3,6,1,4,5,1,6,5,1,1,2,5,2,1,4,1,3,6,3,1,3,6,1,4,5,1,6,5,1,1,2,6,5
,2,4,1,3,6,4,4,3,2,3,1,6,5,2,4,1,3
P2:6,6,1,6,2,3,3,5,1,5,1,1,1,4,3,6,4,4,3,2,3,1,6,5,2,4,1,3,6,4,4,3,2,3,1,6,5,2,4,1,3,5
,2,4,1,3,6,4,4,3,2,3,1,6,5,2,4,1,3
------------------------------------
9
99 98 S3 96 95 94 93 L3 91
80 81 82 83 84 85 86 87 88
79 S2 77 L3 75 74 73 72 71
60 61 62 63 64 65 66 67 68
59 58 57 S1 55 54 53 52 51
L2 41 42 43 44 45 46 S2 48
```

~ Sample input continues on next page ~

Himanshu, continued

```
39 38 37 36 L1 34 33 32 31
20 21 22 23 24 25 26 27 28
L2 S1 17 16 15 14 13 12 S3
P1:1,6,6,6,2
P2:1,4,3,1,5
-----------------------------------


10
99 98 S3 96 95 94 93 L3 91 90
80 81 82 83 84 85 86 87 88 89
79 S2 77 L3 75 74 73 72 71 70
60 61 62 63 64 65 66 67 68 69
59 58 57 S1 55 54 53 52 51 50
L2 41 42 43 44 45 46 S2 48 49
39 38 37 36 L1 34 33 32 31 30
20 21 22 23 24 25 26 27 28 29
L2 S1 17 16 15 14 13 12 S3 10
00 01 02 03 04 05 L1 07 08 09
P1:6,6,5,6,6,6,6,6,2,5
P2:6,6,5,6,6,6,6,6,2,1
-----------------------------------
10
99 98 S3 96 95 94 93 L3 91 90
80 81 82 83 84 85 86 87 88 89
79 S2 77 L3 75 74 73 72 71 70
60 61 62 63 64 65 66 67 68 69
59 58 57 S1 55 54 53 52 51 50
L2 41 42 43 44 45 46 S2 48 49
39 38 37 36 L1 34 33 32 31 30
20 21 22 23 24 25 26 27 28 29
L2 S1 17 16 15 14 13 12 S3 10
00 01 02 03 04 05 L1 07 08 09
P1:6,6,5,6,6,6,6,6,2,5
P2:6,6,5,6,6,6,6,6,2,5
-----------------------------------
3
06 L1 08
05 S1 03
00 L1 S1
P1:1,1
P2:2,2
-----------------------------------
```

~ Sample input continues on next page ~

Himanshu, continued

```
2
03 L1
00 L1
P1:1,1
P2:2,2
-----------------------------------
5
20 L6 L5 S5 L6
L5 L4 S4 S5 S3
S3 S4 L4 L3 S4
L3 S1 L1 L2 S2
00 L1 S1 L2 S2
P1:6,6,5,6,6,6,6,6,2,5
P2:6,6,5,6,6,6,6,6,2,5
-----------------------------------
11
S21 L08 S02 S16 S03 115 L06 S12 118 S10 120
L05 L17 S17 S01 L06 104 L12 S05 101 L11 S08
S15 089 L02 L07 092 S07 S09 S03 S14 L08 S11
087 S22 L12 L13 L19 082 081 S18 S20 078 077
S04 067 L04 L07 L15 071 L18 L19 S17 L17 076
L02 064 S18 062 S02 S07 L16 L14 S13 S15 S06
S14 L10 L13 S06 S16 L09 S01 051 S20 053 S12
043 L09 L11 L14 L16 S08 L03 036 L03 L18 L10
S19 L01 S10 S13 026 027 L01 029 030 031 032
S05 L04 S11 018 017 016 015 S04 S09 S21 011
000 S23 L05 003 004 005 S23 007 L15 S22 S19
P1:1,4,3,1,5,6,1,5,1,3,6,1,4,5,1,6,5,1,1,2,5,2,1,4,1,3,6,3,1,3,6,1,4,5,1,6,5,1,1,2,6,5
,2,4,1,3,6,4,4,3,2,3,1,6,5,2,4,1,3
P2:6,6,1,6,2,3,3,5,1,5,1,1,1,4,3,6,4,4,3,2,3,1,6,5,2,4,1,3,6,4,4,3,2,3,1,6,5,2,4,1,3,5
,2,4,1,3,6,4,4,3,2,3,1,6,5,2,4,1,3
-----------------------------------
11
S21 L08 S02 S16 S03 115 L06 S12 118 S10 120
L05 L17 S17 S01 L06 104 L12 S05 101 L11 S08
S15 089 L02 L07 092 S07 S09 S03 S14 L08 S11
087 S22 L12 L13 L19 082 081 S18 S20 078 077
S04 067 L04 L07 L15 071 L18 L19 S17 L17 076
L02 064 S18 062 S02 S07 L16 L14 S13 S15 S06
S14 L10 L13 S06 S16 L09 S01 051 S20 053 S12
043 L09 L11 L14 L16 S08 L03 036 L03 L18 L10
S19 L01 S10 S13 026 027 L01 029 030 031 032
```

~ Sample input and output continues on next page ~

Himanshu, continued

```
S05 L04 S11 018 017 016 015 S04 S09 S21 011
000 S23 L05 003 004 005 S23 007 L15 S22 S19
P1:1
P2:6
-----------------------------------
```

Sample output:

```
Game #1: Player 2 wins!
Game #2: Neither Player 1 or Player 2 won.
Game #3: Player 1 wins!
Game #4: Both players win!
Game #5: Player 1 wins!
Game #6: Both players win!
Game #7: Both players win!
Game #8: Player 2 wins!
Game #9: Neither Player 1 or Player 2 won.
```

9. Justin

Program Name: Justin.java **Input File: justin.dat**

Justin's parents are considering buying a new car and when that happens he will get the current family car. He will finally have a car of his own! He wants to help his parents by creating a report that summarizes the purchase details for each of the cars they are considering.

They will be using a car loan to pay for most of the purchase so the information needed to help them make a decision is the monthly payment, the final cost of the car after making the loan payments and the amount of interest that will have been paid. Justin has found the following formula for calculating the monthly payment.

Formula	**Legend**
$p = \frac{ar}{1-(1+r)^{-n}}$	a = amount of loan r = monthly interest rate n = number of payments p = monthly payment

The loan amount ***a*** is the purchase price less the down payment. The monthly interest rate ***r*** is 1/12 of the APR or annual percentage rate for the loan as a decimal value not a percentage. The number of monthly payments ***n*** and monthly payment amount ***p*** are as expected. The monthly payment must be rounded to nearest penny; otherwise, partial pennies add up to extra pennies after all payments.

For each car, calculate and display the monthly payment, final cost which is the total of all payments plus the down payment, and the amount of interest paid which is the total of payments less the purchase price.

Using first set of data as a sample:

Monthly payment is $587.07
Final cost is $37224.20 [587.07 * 60 + 2000.00]
Interest paid is $6236.79 [37224.20 – 30987.41]

Input: The first line contains an integer ***T*** which is the number of test cases with $3 \le T \le 20$. Each of ***T*** following lines are one complete test case and will contain four pieces of data:

Purchase price between and including $10,000.00 and $100,000.00
Down payment ***d*** with $\$0.00 \le d \le \$5000.00$
APR as a percentage with $1.0000\% \le APR \le 20.0000\%$
Number of monthly payments ***n*** with $12 \le n \le 96$

Output: One line for each test case containing 3 values: monthly payment, final cost, and total interest paid. Each value is rounded to 2 decimal places and has a leading $ with the value right-aligned in a field of 9 columns. One space separates the fields.

Sample input:

```
3
30987.41 2000 7.95 60
65432.78 3000 9.375 84
15937.07 1000 15.66 48
```

Sample output:

```
$   587.07 $ 37224.20 $  6236.79
$  1016.41 $ 88378.44 $ 22945.66
$   420.72 $ 21194.56 $  5257.49
```

10. Marek

Program Name: Marek.java **Input File: marek.dat**

Your parents have asked you to help your younger brother, Marek, with his homework. He gave you 10 bucks to do it for him while he goes to the movies with his friends. Write a computer program to do all his homework problems.

Input: The input will begin with a single integer, `t` (1 <= `t` <= 100) denoting the number of test cases to follow. Each of the `t` test cases will be composed a single line with an infix expression made up of integers and `+`, `-`, `*`, `/`, `(`, `)`. Use standard order of operations to evaluate the expressions, all integers and operators on the line will be separated by spaces.

Output: For each of the `t` test cases, print out the answer to the given infix expression. Use integer division, i.e. ½ = 0. If division by 0 ever occurs, print out "`Infinity.`"

Sample input:

```
4
1 + 2 - 1
4 * 7 - 3
3 / 2 + 1
4 + 2 * 3
```

Sample output:

```
2
25
2
10
```

11. Priyanka

Program Name: Priyanka.java **Input File: priyanka.dat**

Priyanka is an elementary school teacher working with her students on their spelling words. The students have already learned about vowels and consonants, and they are also able to alphabetize letters. Priyanka noticed one very special type of word that she thought was pretty cool. These are words with all of the vowels at the front of the word and all the consonants at the end. She also noticed that sometimes, the vowels were sorted within themselves and the consonants were sorted as well.

Then she discovered what she called "Woo-Hoo" words. In it, the vowels {A,E,I,O,U} are sorted in reverse order at the start of the word and the consonants were sorted in ascending order at the end of the word.

Here are some examples: OAKS, EAST, OOPS, EASY, A, OF

To have a little fun in class, any time one of her students spotted a spelling word like this, they would shout "WOO-HOO!!!" Thus they are called Woo-Hoo Words.

Please write a program to test a word to see if it is worthy for a shout of "WOO-HOO!!!"

Input: Input will consist of an integer N, the number of test cases. The number of test cases will be in range [1,20]. Each subsequent line will contain one string consisting only of upper-case letters with no spaces and no punctuation marks. Each string will have a length in the range [1,32]

Output: Each line of output will consist of the letters of the input string rearranged. The vowels {A,E,I,O,U} will come first in reverse alphabetical order followed immediately by the consonants in alphabetical order. There is not a space separating those two groups. If a word is a "woo-hoo" word, then "WOO-HOO" will be written before the output word with one space separating it from the word.

Sample input:

```
5
UNIVERSITY
HELLO
ABCDEFGHIJ
EACH
I
```

Sample output:

```
UIIENRSTVY
OEHLL
IEABCDFGHJ
WOO-HOO EACH
WOO-HOO I
```

12. Sunny

Program Name: Sunny.java **Input File: sunny.dat**

Sunny is only a freshman and has been really excited about taking UIL programming. She has been looking at the dual-credit courses that are available through her school and hopes to start college with enough credits to reduce her in-person college experience to 3 years. She discovered the Texas Common Course Numbering System (TCCNS) but the data she found was not very user-friendly. She needs your help to generated a sorted list of courses. (Note: the data here is less than 10% of the real data!)

Input: An unknown number of lines with each line containing a 9-character course code followed by a single space and the course title. There are no punctuation or special characters except for the dash in the course codes.

Output: Alphabetical (A…Z) list by course title with course codes following the titles in a set of parentheses, separated by a single space.

Sample input:

```
PHYS-2125 UNIVERSITY PHYSICS I LAB
MATH-1342 ELEMENTARY STATISTICAL METHODS
COSC-2325 COMPUTER ORGANIZATION
COSC-1336 PROGRAMMING FUNDAMENTALS I
PHYS-2325 UNIVERSITY PHYSICS I
COSC-1301 INTRODUCTION TO COMPUTING
COSC-1315 INTRODUCTION TO COMPUTER PROGRAMMING
COSC-1320 C PROGRAMMING
MATH-2313 CALCULUS I
COSC-1337 PROGRAMMING FUNDAMENTALS II
```

Sample output:

```
C PROGRAMMING (COSC-1320)
CALCULUS I (MATH-2313)
COMPUTER ORGANIZATION (COSC-2325)
ELEMENTARY STATISTICAL METHODS (MATH-1342)
INTRODUCTION TO COMPUTER PROGRAMMING (COSC-1315)
INTRODUCTION TO COMPUTING (COSC-1301)
PROGRAMMING FUNDAMENTALS I (COSC-1336)
PROGRAMMING FUNDAMENTALS II (COSC-1337)
UNIVERSITY PHYSICS I (PHYS-2325)
UNIVERSITY PHYSICS I LAB (PHYS-2125)
```

UIL Computer Science Competition

Invitational A 2023

JUDGES PACKET - CONFIDENTIAL

I. Instructions

1. The attached printouts of the judge test data are provided for the reference of the contest director and programming judges. Additional copies may be made if needed for this purpose.

2. This packet must remain CONFIDENTIAL. Additional copies may be made and returned to schools when other confidential contest material is returned.

II. Table of Contents

Problem #1
60 Points

1. Abhinav

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

Test Input File: None

Test Output To Screen:

```
Ultimate Important Languages
U-I-L
UIL!
```

Problem #2
60 Points

2. Alan

Program Name: Alan.java **Input File: alan.dat**

Test Input File:

```
20
-146
-57
-166
-17
-108
-173
-93
-84
-183
-54
-164
-187
99
105
178
-139
-38
-189
75
69
```

Test Output To Screen:

```
-146 degrees Fahrenheit is equal to 174.26 Kelvin
-57 degrees Fahrenheit is equal to 223.71 Kelvin
-166 degrees Fahrenheit is equal to 163.15 Kelvin
-17 degrees Fahrenheit is equal to 245.93 Kelvin
-108 degrees Fahrenheit is equal to 195.37 Kelvin
-173 degrees Fahrenheit is equal to 159.26 Kelvin
-93 degrees Fahrenheit is equal to 203.71 Kelvin
-84 degrees Fahrenheit is equal to 208.71 Kelvin
-183 degrees Fahrenheit is equal to 153.71 Kelvin
-54 degrees Fahrenheit is equal to 225.37 Kelvin
-164 degrees Fahrenheit is equal to 164.26 Kelvin
-187 degrees Fahrenheit is equal to 151.48 Kelvin
99 degrees Fahrenheit is equal to 310.37 Kelvin
105 degrees Fahrenheit is equal to 313.71 Kelvin
178 degrees Fahrenheit is equal to 354.26 Kelvin
-139 degrees Fahrenheit is equal to 178.15 Kelvin
-38 degrees Fahrenheit is equal to 234.26 Kelvin
-189 degrees Fahrenheit is equal to 150.37 Kelvin
75 degrees Fahrenheit is equal to 297.04 Kelvin
69 degrees Fahrenheit is equal to 293.71 Kelvin
```

Problem #3
60 Points

3. Bianca

Program Name: Bianca.java **Input File: bianca.dat**

Test Input File:

```
20
3 4
12 14
5 12
5 8
3 7
12 34
345 678
12 345
543 23
123 321
455 54
342 5
13 4
134 5
6 8
41 32
4 21
22 90
3 9292
45 54
```

Test Output To Screen:

```
5.00
18.44
13.00
9.43
7.62
36.06
760.73
345.21
543.49
343.76
458.19
342.04
13.60
134.09
10.00
52.01
21.38
92.65
9292.00
70.29
```

Problem #4
60 Points

4. Cho

Program Name: Cho.java **Input File: cho.dat**

Input File: (lines that are indented are continuation of previous line)

```
13
THE HOG GOES TO THE DOG TO CATH THE FROG IN THE BOG
WITCHES MADE MY SOUP, NOW I CAN FLY WHEEEEE
Qwertyuiolkjhgfdazxcvbnm
SOMEBODY ONCE TOLD ME THE WORLD WAS GONNA ROLL ME
I AINT THE SHAPRPETSTINF TOOL INT EJ SHED
IW AS LOOKInjr kind a dumb iwth the finger and th e thimmb in te shaoe of ajnd l
   on her  4hjread
well they dont sto cmingi and thiey neve stop coming
Did you ever hear the tragedy of Darth Plagueis The Wise? I thought not. It's
   not a story the Jedi would tell you. It's a Sith legend. Darth Plagueis was a
   Dark Lord of the Sith, so powerful and so wise he could use the Force to
   influence the midichlorians to create life… He had such a knowledge of the
   dark side that he could even keep the ones he cared about from dying. The
   dark side of the Force is a pathway to many abilities some consider to be
   unnatural. He became so powerful… the only thing he was afraid of was losing
   his power, which eventually, of course, he did. Unfortunately, he taught his
   apprentice everything he knew, then his apprentice killed him in his sleep.
   Ironic. He could save others from death, but not himself.
3333333444444555555%%%%%%g
ijrijijfijr2950432865438768%$#$(&**^&^^%&(^%&*%$%(*^&(*&%^*%(^&^%^%%^%%^%&^$%#)%
   ^$(*&%^%#&*%%^*)(^&%^#*&(*%%$^849756389876387854639764563756453654563987637856387637
Hello
Its me
I was wondering if after all these years youd like to meet
```

Output To Screen:

```
O
E
Q
O
T
e
e
e
g
i
l
I
E
```

Problem #5
60 Points

5. Damian

Program Name: Damian.java **Input File: damian.dat**

Test Input File:

```
10
72
144
1331
621
554433
1
1296
77
111
333333
```

Test Output to Screen:

```
1 2 3 4 6 8 9 12 18 24 36
1 2 3 4 6 8 9 12 16 18 24 36 48
1
1 3 9 23 27 69
1 3 159
1
1 2 3 4 6 8 9 12 16 18 24 27 36 48
1 7
1 3 37
1 3 7 9 13 37 39 259
```

Feng #6
60 Points

6. Feng

Program Name: Feng.java **Input File: feng.dat**

Test Input File:
```
10
02:59
09:18
12:05
03:14
03:15
03:16
10:05
08:45
04:26
02:55
```

Test Output To Screen:
```
Hour: 0.5 Minute: 96.0
Hour: 171.0 Minute: 342.0
Hour: 87.5 Minute: 60.0
Hour: 353.0 Minute: 6.0
Hour: 352.5 Minute: 0.0
Hour: 352.0 Minute: 354.0
Hour: 147.5 Minute: 60.0
Hour: 187.5 Minute: 180.0
Hour: 317.0 Minute: 294.0
Hour: 2.5 Minute: 120.0
```

Problem #7
60 Points

7. Gael

Program Name: Gael.java **Input File: gael.dat**

Test Input File:

```
12
HELLO
WORLD
RACECAR
THJDJFKSKFJJE
JKEWIFNEIFNW
IENVIENVREVEVW
RHRHTEIMACOWEMCLFWMR
QWONCEWOIVNINGREONVGEQOINVOIRENV
QEIVNRQPVNRQEOIRVNORENOWNNVQKDXABZCZMVCXNMMVMVCXZBV
VVPOIJFEWQFPOIJEWQFPOIJEWQNWFDKFDV
IFJEWNVIEWVFLTRVLKWPLMKECOKFC
INVENLKQDCNIDWUYBYWVVEYTDEW
```

Test Output To Screen:

```
HELLO
E   L
L   L
L   E
OLLEH
WORLD
O   L
R   R
L   O
DLROW
RACECAR
A     A
C     C
E     E
C     C
A     A
RACECAR
THJDJFKSKFJJE
H           J
J           J
D           F
J           K
F           S
K           K
S           F
K           J
F           D
J           J
J           H
EJJFKSKFJDJHT
JKEWIFNEIFNW
K          N
E          F
```

```
W          I
I          E
F          N
N          F
E          I
I          W
F          E
N          K
WNFIENFIWEKJ
IENVIENVREVEVW
E            V
N            E
V            V
I            E
E            R
N            V
V            N
R            E
E            I
V            V
E            N
V            E
WVEVERVNEIVNEI
RHRHTEIMACOWEMCLFWMR
H                  M
R                  W
H                  F
T                  L
E                  C
I                  M
M                  E
A                  W
C                  O
O                  C
W                  A
E                  M
M                  I
C                  E
L                  T
F                  H
W                  R
M                  H
RMWFLCMEWOCAMIETHRHR
QWONCEWOIVNINGREONVGEQOINVOIRENV
W                              N
O                              E
N                              R
C                              I
E                              O
W                              V
O                              N
I                              I
V                              O
N                              Q
I                              E
N                              G
G                              V
```

```
R                                 N
E                                 O
O                                 E
N                                 R
V                                 G
G                                 N
E                                 I
Q                                 N
O                                 V
I                                 I
N                                 O
V                                 W
O                                 E
I                                 C
R                                 N
E                                 O
N                                 W
VNERIOVNIOQEGVNOERGNINVIOWECNOWQ
QEIVNRQPVNRQEOIRVNORENOWNNVQKDXABZCZMVCXNMMVMVCXZBV
E                                                   B
I                                                   Z
V                                                   X
N                                                   C
R                                                   V
Q                                                   M
P                                                   V
V                                                   M
N                                                   M
R                                                   N
Q                                                   X
E                                                   C
O                                                   V
I                                                   M
R                                                   Z
V                                                   C
N                                                   Z
O                                                   B
R                                                   A
E                                                   X
N                                                   D
O                                                   K
W                                                   Q
N                                                   V
N                                                   N
V                                                   N
Q                                                   W
K                                                   O
D                                                   N
X                                                   E
A                                                   R
B                                                   O
Z                                                   N
C                                                   V
Z                                                   R
M                                                   I
V                                                   O
C                                                   E
```

```
X                                                       Q
N                                                       R
M                                                       N
M                                                       V
V                                                       P
M                                                       Q
V                                                       R
C                                                       N
X                                                       V
Z                                                       I
B                                                       E
VBZXCVMVMMNXCVMZCZBAXDKQVNNWONERONVRIOEQRNVPQRNVIEQ
VVPOIJFEWQFPOIJEWQFPOIJEWQNWFDKFDV
V                                   D
P                                   F
O                                   K
I                                   D
J                                   F
F                                   W
E                                   N
W                                   Q
Q                                   W
F                                   E
P                                   J
O                                   I
I                                   O
J                                   P
E                                   F
W                                   Q
Q                                   W
F                                   E
P                                   J
O                                   I
I                                   O
J                                   P
E                                   F
W                                   Q
Q                                   W
N                                   E
W                                   F
F                                   J
D                                   I
K                                   O
F                                   P
D                                   V
VDFKDFWNQWEJIOPFQWEJIOPFQWEFJIOPVV
IFJEWNVIEWVFLTRVLKWPLMKECOKFC
F                               F
J                               K
E                               O
W                               C
N                               E
V                               K
I                               M
E                               L
W                               P
V                               W
```

```
F                           K
L                           L
T                           V
R                           R
V                           T
L                           L
K                           F
W                           V
P                           W
L                           E
M                           I
K                           V
E                           N
C                           W
O                           E
K                           J
F                           F
CFKOCEKMLPWKLVRTLFVWEIVNWEJFI
INVENLKQDCNIDWUYBYWVVEYTDEW
N                           E
V                           D
E                           T
N                           Y
L                           E
K                           V
Q                           V
D                           W
C                           Y
N                           B
I                           Y
D                           U
W                           W
U                           D
Y                           I
B                           N
Y                           C
W                           D
V                           Q
V                           K
E                           L
Y                           N
T                           E
D                           V
E                           N
WEDTYEVVWYBYUWDINCDQKLNEVNI
```

Problem #8
60 Points

8. Himanshu

Program Name: Himanshu.java **Input File: himanshu.dat**

Test Input File: (lines that are indented are continuation of previous line)

```
11
16
0255 L036 L084 L052 L080 L023 L007 L090 0247 L061 L079 L051 L044 L045 L008 L090
S009 L087 L040 L051 L069 S005 0230 L095 L060 L099 L018 0235 L062 S001 L025 L019
L010 L015 L091 L008 L028 0218 L014 L072 0215 L043 S002 L100 S011 S007 L077 L089
L097 L048 L060 L016 L030 L098 L075 0199 L027 L023 0202 L055 L074 L048 L022 L089
L055 L041 L094 L003 L016 S004 S012 L086 L034 L054 L082 L034 L076 L092 S005 L031
0160 L004 L062 L042 S002 0165 L092 L079 L059 L071 L017 L098 L082 L067 0174 L024
S004 0158 S009 L003 L012 L031 L036 L057 L056 L041 L002 L009 S006 0146 L039 L054
L080 S008 L088 L099 S010 L056 L004 L009 L006 L057 L072 L065 L021 L064 L063 L068
0127 S011 0125 L043 L018 L039 L035 L038 L045 L053 L033 L065 L083 L077 L005 S012
L069 L053 L076 L017 0100 L075 L022 L050 L083 L040 L064 L013 L047 L066 L096 L001
L093 S006 0093 L086 0091 0090 L093 S003 L037 L032 L058 L046 L044 S008 0081 L081
L059 L073 L006 L078 L035 L013 0070 L066 L068 L095 L067 L030 L007 L084 L005 0079
L026 S010 L014 L032 0059 L100 L011 L085 L026 L049 L037 0052 L091 L027 L015 L002
L058 L070 L038 L010 0036 L012 L049 L029 L021 L028 S001 L011 L042 L096 L087 S003
0031 L074 L063 L073 L047 L024 L061 L094 L071 L070 0021 0020 L019 L020 L078 L052
0000 L029 L033 L101 L088 0005 L085 L081 S007 L020 L050 L025 L101 L097 L046 L001
P1:1,4,3,2,5,4,3,2,3,4,3,4,2,5,3,1,4,2,1,3,2,1,5,2,5,3,3,4,5,5,3,2
P1:2,5,4,3,1,3,1,4,4,1,1,4,3,3,3,2,1,1,1,3,2,1,3,3,2,4,3,1,5,1,3,1
-----------------------------------
10
099 098 097 L21 L35 L06 L14 092 L37 L31
L35 S01 L36 L34 L34 085 L10 L09 088 089
L24 L28 L22 L27 L23 L28 S01 L12 L30 L24
L32 L17 L06 L26 L19 L08 L25 L20 L27 L18
L30 L18 L05 L23 055 054 053 L29 L08 L19
L15 L10 L02 043 L22 L01 L04 047 L07 L02
L29 L33 L03 L11 035 L13 033 032 L04 L15
L17 L09 L26 023 L16 L31 L20 L38 028 L07
L16 L32 L25 016 L01 014 L03 L05 L11 L37
000 L13 L21 003 L33 L38 L12 L36 L14 009
P1:1,4,2,1,1,1,5,2,1,5,4,2,4
P1:2,2,2,1,4,2,2,5,5,5,4,5,2
-----------------------------------
24
0575 L001 0573 L036 L193 L071 S034 S059 L043 S052 S029 0564 L131 L053 S070 L175
   L089 L187 L069 S015 L091 L066 L152 L184
L042 L090 L049 S007 L017 L026 L002 L107 L111 S062 L027 L024 S051 0541 S031 L158
   L101 L046 L016 L133 L116 S068 L065 L039
L030 S032 L022 L021 L059 L047 L054 L035 L105 L198 S049 L037 0515 L141 L031 L156
   L094 L028 S027 L151 S068 L149 L185 S021
L138 L006 L118 S061 L005 L160 L100 L015 S034 L196 0490 S042 L153 0493 L188 L078
   L174 L024 L143 S040 0500 L029 L032 L014
L149 S012 L018 L176 L184 L137 L121 0472 S009 L195 S016 S040 S047 L059 L001 L192
   L127 L194 L167 0460 L064 L138 S026 0456
```

~ Input continues on next page ~

Himanshu, continued

```
S063 L040 L097 L107 L027 L189 S010 S006 0440 S027 L072 S022 0444 L135 S066 L142
   L159 L061 L041 S046 L083 L032 L179 L044
L177 L128 L159 S018 S033 L182 L084 L134 L196 L182 L120 L104 S032 L077 S041 L148
   S035 L108 L002 L150 L067 L102 0409 L054
L128 L144 0386 L155 L020 0389 S012 L113 L031 L063 L181 S030 L050 L135 L123 S055
   L157 L142 L056 L197 L052 L090 S067 L008
L070 L137 L103 S016 L013 L081 L162 L106 L099 L157 L094 L186 L019 L154 L186 L038
   L122 L140 L112 L143 L052 S001 S048 L106
L046 S028 L146 S011 L025 L039 S017 L051 L030 L102 L136 L078 L117 L191 L036 S003
   S048 S054 L165 L139 S045 L067 S038 S066
0335 S038 L166 L153 0331 S017 S059 L162 L169 L171 S023 L158 S031 L172 L083 L009
   L076 L161 S043 L034 L191 S029 L105 S039
L111 L150 L097 L044 L155 L075 L134 0295 L058 S018 S057 L180 S002 S008 L009 L019
   S061 0305 L115 L087 S033 S024 L178 S011
0287 L095 L007 S056 L118 S005 L005 L173 L038 L179 S036 L087 0275 L076 L011 L055
   L114 L048 S019 L189 L021 L108 S014 L074
S035 L035 L043 L003 L175 L115 L085 S014 L119 L125 L123 L086 L079 S053 S021 S019
   S050 0257 L141 L109 S058 L049 S069 L177
L074 L069 S049 L088 L092 S025 L166 L121 S037 L010 L170 S070 S044 L163 L169 S001
   L173 S064 S056 L131 S063 L068 S065 L130
L010 L178 L114 L077 S067 L154 S051 S020 L101 L028 S005 L098 L110 S009 L011 L056
   S003 L066 L130 S023 L073 L023 L084 0215
S042 0190 L062 L163 L126 S060 0185 L198 L015 L110 L103 L037 L091 0178 L004 L183
   L133 L174 L033 L082 L013 L151 L164 L183
L018 S055 S047 0147 L080 L045 L117 L096 L079 L086 S030 S004 L081 L003 S060 L113
   L176 S039 0162 S002 L055 0165 L064 L185
S054 L132 S013 S024 L120 L125 L164 L180 L181 S043 S010 L145 L100 S008 L023 S062
   L146 S015 L029 L124 S025 L129 L095 L057
L104 L170 S046 L085 L007 L190 L193 L165 S044 S013 S057 L140 L099 L116 L070 L127
   S052 S036 0114 S045 L092 S006 L025 L050
L071 L119 L020 L061 L160 L006 L197 L089 0087 L062 L122 L132 L145 L060 L147 0080
   L188 L194 L171 L156 L093 L008 L190 L051
S053 L129 L057 L047 0052 S020 L063 L187 S026 S004 L167 L147 L073 L161 L112 L152
   S069 L041 S058 L126 L022 L080 L058 S041
L075 L026 L098 0044 L139 S050 L004 0040 L109 L096 L014 L048 S007 L168 L195 0032
   L042 L065 L060 L053 L033 S022 S064 S028
0000 L045 L192 L012 L040 L082 0006 0007 L144 L168 L148 L088 S065 L093 L068 L124
   L072 L016 L017 L136 L034 L172 S037 L012
P1:2,5,2,4,3,4,4,5,1,1,5,3,5,5,2,3,5,1,5,5,4,5,3,3,3,5,5,4,2,1,2,2,4,3,2,4,4,3,3
   ,1,5,5,2,2,5,3,5,1,3,5,2,1,5,3,5,3,1,2,3,2,2,3,2,1,2,5,2,1,3,3,3,3,5
P1:2,5,4,5,3,4,1,3,1,4,2,2,2,3,3,1,1,5,1,3,2,3,2,2,4,2,1,1,2,4,2,3,1,1,1,5,4,3,2
   ,1,2,1,2,4,1,5,3,2,5,2,3,2,1,1,1,1,2,3,4,5,5,2,2,1,2,2,5,1,3,1,1,5,1
-------------------------------------
5
20 S2 22 L3 24
19 L3 17 S3 S2
L2 S1 S1 13 S3
L1 08 07 06 05
00 L1 02 L2 04
P1:3,3,4,3,3,4
P1:3,4,2,2,5,4
-------------------------------------
```

~ Input continues on next page ~

Himanshu, continued

```
28
0783 S056 S074 S064 L057 L211 L063 S041 S094 L173 L148 L173 S066 S047 L111 L152
   S029 S098 S083 L139 L185 L163 S006 L136 S087 L126 L145 L085
L230 L178 L055 L050 L109 S034 L071 S014 S003 S147 S127 L073 S020 L175 L056 S100
   L077 L072 L086 S037 L221 L081 L082 L034 L182 L022 L224 S137
L213 L065 S158 S107 S057 S143 L091 L029 L150 S082 L016 L167 L113 L108 L117 S121
   L183 L202 L149 L127 S071 S021 S027 S002 L041 S027 S046 L027
L169 L023 L138 L012 L134 L144 L076 S001 L212 L006 S045 S126 S145 L132 L217 L006
   S019 L054 L053 L156 L076 L021 S065 S005 L068 L191 L127 S092
L082 S029 L179 L042 L168 L164 L093 L151 L170 S008 S039 L107 S050 S113 S065 L028
   S058 S031 S051 S058 L004 S153 L146 L197 L150 S061 L096 L203
L176 S097 S095 L043 S136 S105 L041 S038 L162 L216 L160 L009 L004 S111 L194 L054
   L045 L223 L217 L165 L227 L084 S043 S042 L185 S010 S121 S034
L044 S025 S012 S130 L124 S152 S012 L195 S156 L002 S025 L191 L230 L218 0601 L222
   L183 L013 L036 S075 L211 S133 S158 L200 L010 L109 L158 L140
L038 L198 L207 S083 L215 S010 S134 L140 L156 L002 S110 L193 L192 S032 L059 S151
   L048 L015 L063 L128 L112 L077 S030 L079 S125 L095 L166 L074
L100 L119 S036 L126 L131 S159 L219 L103 L121 L142 S082 S044 S150 S009 L139 S154
   L080 L019 L094 L219 S007 S015 S015 L035 L015 S054 S052 S021
S018 S109 L167 S129 L135 S142 S011 L180 S007 L111 L196 S112 S079 L214 S041 S148
   L220 L070 S094 L137 L046 S086 S118 S013 L225 L078 L037 L093
S073 L081 S139 L110 S017 L158 S074 L049 L177 S085 S091 L170 L007 S081 L029 S050
   L188 S093 S132 S003 L103 S114 L047 S046 S033 S108 S156 S020
L130 S143 L192 L117 S035 L174 S077 L033 S014 L024 S053 L154 L154 S088 S068 S147
   L213 L204 L074 S037 S141 S136 L069 L069 S093 L065 S084 L144
L159 L220 L141 S068 L118 S128 L164 L075 L048 S124 L112 S016 L060 L010 L083 L172
   S023 S102 L092 S080 L206 L047 L115 S114 L149 S060 S076 L032
L145 L088 S117 L068 S028 S044 L228 L122 S072 S128 L088 L031 S089 L098 L132 L209
   L100 S139 L051 L187 L091 S090 L133 L176 L061 L129 L086 S078
S001 S040 L210 S141 S154 S057 S024 L115 L097 S051 L114 S067 L210 L084 S138 S048
   S031 L184 L121 L031 S022 L037 S004 L159 L114 L199 L181 L028
L151 L089 L101 L104 L094 L229 L062 L194 L142 L120 L177 S067 L212 S096 L161 S056
   S115 S008 L053 L018 S160 L125 L040 L075 L205 S101 S062 L189
L196 S035 S064 L199 S032 L043 L008 L001 L083 S059 L124 L011 L160 L017 L180 L008
   S123 S138 L208 L001 S135 S102 L165 S150 L116 S024 L187 S100
L229 S095 S053 L129 L120 S137 L099 L032 L042 L125 L067 L171 L012 L020 L216 S047
   S149 L107 S088 S036 L026 S117 S075 S124 L005 S080 L106 S146
L206 L146 S098 S071 L155 L171 S096 S099 L222 S145 S049 S055 L201 L155 L226 L174
   L072 S122 L064 S151 S070 L013 L014 S118 L011 S066 S060 S120
S149 S011 L122 S087 L071 L214 L200 S097 L186 S002 S054 S129 S110 L218 L066 L020
   L060 S107 S072 S104 L148 S042 L087 L143 S005 L066 L178 L014
S140 L130 S086 L073 S023 L025 L147 L022 L162 L057 L036 S073 L169 L035 L089 L097
   L161 L105 S123 L166 L005 S134 S146 L118 S076 S030 S112 S069
L095 L179 S144 L061 L131 L123 S063 L116 S131 S091 L190 S159 L182 L003 L099 S038
   S043 L090 L040 L143 S160 L223 L141 S022 S155 S106 S069 S113
L190 L134 L056 L034 S108 L087 L019 L189 S148 L184 L186 S144 L147 S061 L204 L181
   S062 L096 S155 S017 L209 L163 L138 L193 S085 S026 S092 L207
L113 S130 S026 S090 L038 L203 S103 L080 L227 L157 L058 L102 S125 S103 L033 S127
   S006 L064 L079 L039 S119 S070 L128 L045 S077 L090 L016 S157
L009 S131 L195 S120 L226 S033 S115 L044 S157 S019 L172 L108 S132 L168 L025 S101
   L003 S111 L078 L201 S039 L098 L153 L221 L052 S040 L188 L102
```

~ Input continues on next page ~

Himanshu, continued

```
L106 S142 L051 L067 L058 S063 L030 S081 L027 L205 L137 L026 L104 0069 S049 S140
   S045 S106 L153 L050 L197 S016 L135 S009 L052 S048 S126 S105
L175 L198 L225 L157 S059 L017 L105 L215 L024 L046 L224 L049 L023 S109 L062 L119
   L018 L055 L007 L085 S153 L152 S135 L136 L202 L101 S099 S013
0000 L092 S028 S116 S119 S152 S116 L030 L228 L021 S122 S055 S084 L070 S089 L039
   S079 L059 S104 S078 S004 S018 L208 L123 L110 L133 S052 S133
P1:1,1,3,5,4,2,5,2,2,5,2,3,4,2,4,5,4,1,3
P1:4,3,5,4,2,3,1,4,3,4,2,5,4,2,2,2,1,3,5
-----------------------------------
31
0930 L036 L096 S090 S096 S068 S018 L103 L083 0939 0940 L097 0942 S018 L159 L059
   L098 0947 L169 0949 0950 S012 S045 0953 L132 S013 L020 L110 0958 L149 0960
S003 0928 0927 0926 0925 0924 L117 0922 S072 0920 S016 L125 L132 L025 0915 0914
   0913 L043 S022 S008 L081 0908 0907 L090 0905 0904 0903 0902 L079 L027 L035
S050 0869 0870 S086 0872 L055 L068 L058 L018 0877 S010 S002 S094 L064 L155 0883
   0884 0885 0886 L113 0888 0889 L130 0891 0892 0893 S066 L022 S028 L166 S019
L031 0866 L056 0864 0863 L161 L042 S067 0859 0858 0857 0856 L147 0854 0853 L041
   0851 S051 S045 L019 0847 S021 L173 L077 L142 0842 0841 0840 S027 0838 L161
S085 0807 S046 L028 0810 0811 S082 0813 L041 0815 0816 0817 L036 0819 L137 0821
   0822 S084 0824 S026 L118 0827 S020 L005 L171 L123 0832 0833 0834 0835 0836
0805 S029 0803 0802 0801 L068 0799 L063 S021 S035 0795 L095 0793 0792 L098 L069
   0789 0788 L153 L076 0785 0784 0783 L131 0781 L134 S080 L162 L115 0776 L078
0744 L075 0746 L106 S017 L163 L146 S019 S056 0753 L104 L057 0756 L091 0758 S070
   S030 S081 L060 S093 L102 L100 0766 0767 S079 0769 0770 L082 L040 0773 L087
L117 L166 L080 0740 L039 0738 L165 S014 L029 0734 L125 L108 0731 S044 L136 0728
   L145 L004 S072 L090 L054 L092 0721 0720 L059 0718 0717 0716 L049 0714 L148
L107 L126 0684 S092 0686 L085 0688 0689 L057 0691 0692 0693 0694 L035 L066 L150
   0698 L110 L007 S069 0702 0703 0704 L074 L032 L128 S091 0709 L151 0711 L012
S058 0680 0679 0678 L030 L165 S031 L023 0673 0672 0671 S062 L121 L093 S055 0666
   0665 0664 0663 0662 0661 S041 0659 L046 L072 0656 L139 0654 S051 0652 0651
L094 L088 0622 0623 S078 L048 L163 0627 0628 L135 S067 L072 L070 L003 S015 0635
   0636 0637 L024 L135 S042 S048 0642 0643 S074 L160 S022 S056 0648 0649 S047
S069 0618 0617 S024 S037 0614 L065 S075 0611 L081 0609 0608 S071 0606 0605 0604
   L109 L037 L084 L025 S043 S083 L122 L119 S095 L144 L014 L075 L009 0590 0589
S096 0559 L116 S015 0562 0563 S092 S009 0566 0567 0568 S085 L019 0571 S041 S040
   L088 0575 S035 0577 S031 0579 L034 L111 0582 S011 L133 L033 0586 0587 S048
0557 0556 0555 L038 0553 0552 L099 L079 S010 0548 0547 L064 0545 0544 0543 0542
   0541 L137 0539 L050 0537 0536 0535 L085 L043 L144 0531 L093 S034 L032 0527
0496 0497 0498 0499 0500 S044 L026 L169 0504 0505 S076 0507 0508 0509 L021 L170
   0512 0513 0514 L055 S066 L070 0518 L065 L172 S077 S071 0523 S078 0525 0526
S059 0494 S032 L152 L095 L078 0489 S052 L154 0486 L050 0484 0483 0482 L022 0480
   L006 L039 L139 0476 L086 0474 S086 0472 0471 0470 L008 L100 S081 0466 0465
S070 L018 0436 L092 L164 0439 S011 0441 L089 L001 S058 0445 L119 S050 0448 S057
   0450 0451 0452 S089 L086 S082 S076 0457 L063 S091 L017 0461 0462 0463 0464
L047 0432 0431 L010 0429 S087 0427 S013 0425 0424 L073 L014 S001 L115 0419 S089
   0417 S020 L127 0414 L008 0412 L048 L124 0409 S039 S006 L134 L062 0404 0403
S025 0373 L111 L051 S093 0377 L168 L152 0380 L147 L031 0383 L159 0385 L074 0387
   L107 S060 S053 0391 S027 L148 L114 0395 S023 0397 0398 L112 0400 L121 S028
S073 L052 0369 S004 L033 0366 S029 S009 0363 L102 0361 0360 0359 L167 L029 L011
   0355 L007 0353 S040 S023 S008 L028 0348 L042 L104 L020 S036 L046 L150 0341
L112 L091 0312 L013 0314 L146 0316 0317 0318 S095 L010 0321 0322 0323 L173 L053
   0326 0327 0328 0329 0330 S033 L052 0333 L133 0335 S038 L120 L143 0339 S037
```

~ Input continues on next page ~

Himanshu, continued

```
S049 0308 0307 0306 L083 L009 S047 S007 0301 S053 0299 0298 L156 S060 S088 S068
   0293 S079 0291 0290 L080 L145 S007 L044 0285 0284 S088 0282 0281 0280 0279
0248 L108 0250 S080 0252 0253 0254 L061 S090 0257 L044 0259 L037 S030 0262 L096
   0264 L158 0266 0267 L087 0269 S087 L127 L045 L030 0274 0275 L158 S052 0278
0247 0246 0245 0244 S055 S033 L001 S084 0239 L006 L094 S064 S003 L149 0233 0232
   0231 0230 0229 0228 0227 0226 L003 0224 L084 S054 L076 L040 L005 0218 L097
S064 0187 L106 L101 0190 L015 L172 0193 S006 L103 0196 S034 0198 0199 S017 0201
   L026 S059 0204 L002 0206 S012 0208 L034 L061 S016 0212 0213 S043 S075 L016
L141 L122 S014 0182 0181 L002 0179 L105 L016 L027 S049 0174 S004 S065 0171 0170
   S026 0168 L109 L142 0165 0164 0163 0162 0161 L138 L131 S036 S063 L073 0155
L049 S063 L045 0127 0128 0129 L123 S083 L051 L023 L062 0135 0136 L114 0138 0139
   L113 0141 0142 0143 S054 0145 L151 L126 S046 L118 0150 S077 L021 0153 0154
0123 0122 0121 L120 L056 0118 S061 0116 0115 0114 0113 L130 S024 L156 L170 S062
   0107 L058 L038 L136 L143 0102 0101 0100 L067 L155 L082 L153 S025 S057 S005
L047 S032 0064 0065 L099 S042 0068 0069 0070 0071 0072 L154 L089 0075 0076 L162
   L105 L053 L171 0081 S002 0083 L012 0085 L116 S039 L141 L004 L024 L066 L157
0061 L140 0059 L101 S038 L067 0055 0054 L164 S073 0051 0050 0049 0048 L017 L077
   0045 0044 L138 L011 0041 S074 L013 L015 0037 0036 L054 0034 S094 S001 L071
0000 L160 L168 0003 S005 0005 0006 L128 L157 0009 0010 0011 L060 0013 0014 L071
   0016 L129 L167 L124 0020 0021 S065 0023 0024 0025 L069 L129 0028 S061 L140
P1:5,5,2,3,5,3,3,1,4,4,1,4,1
P1:4,3,5,5,2,2,5,4,5,3,4,3,5
-------------------------------------
12
143 L33 L58 L10 L49 L37 L56 L32 L28 S01 S01 L15
L46 L60 L65 S02 L11 L02 L26 L42 L24 L27 L07 L25
L31 L19 L64 L51 L04 L63 L27 L33 L62 L03 L21 L12
L26 L39 L08 L30 L57 L05 L29 L13 L20 L16 L54 L54
L31 L17 L68 L07 L43 L59 L06 088 L35 L41 L64 L50
L04 L52 L42 L17 L29 L45 L38 L47 L61 L16 L67 S02
L55 L45 L23 L34 L52 L36 L59 L32 L49 L48 L53 L46
L66 L22 L36 L65 L02 L57 L63 L67 L01 L24 L30 L68
L47 L09 L62 L55 L14 L41 L05 L61 L60 L37 L28 L66
L51 L43 L20 L34 L50 L44 L06 L01 L38 L44 L19 L35
L21 L15 021 L11 L39 L53 L23 L18 L58 L14 L48 L03
000 L40 L10 L25 L09 L40 L56 L12 L08 L22 L18 L13
P1:1,5,5,4,4,5,2,2,1,1,3,4,2,4,5,2,5,3,3,2,5,2,4,1,1,4,1,3,4,3,4,2,2,2,3,2,4,3,2
   ,4,2,3,4,2
P1:1,4,3,3,5,5,1,4,1,1,3,4,1,3,1,1,3,3,3,3,3,1,2,5,2,3,2,2,4,1,2,1,4,2,1,4,3,5,5
   ,3,1,5,5,5
-------------------------------------
25
L016 S023 L122 0603 0604 L045 S008 0607 L105 0609 0610 L009 0612 L050 L043 L129
   L110 L071 0618 L038 L080 0621 0622 L107 0624
0599 0598 L135 L090 L071 0594 0593 L112 0591 L104 L083 0588 0587 L025 L087 S001
   0583 S010 L055 0580 L088 0578 L014 0576 0575
L075 S006 0552 L068 0554 L008 0556 0557 L116 L027 S016 0561 L067 0563 L133 0565
   S010 S006 S022 0569 S007 0571 0572 0573 L013
0549 L046 0547 L106 0545 0544 L135 S013 L048 0540 L051 L002 L072 L080 0535 L059
   L037 S023 L036 0530 L041 L063 L127 L036 0525
0500 S008 0502 0503 L118 L123 0506 0507 L125 L108 0510 0511 L030 0513 L022 0515
   L047 0517 L042 0519 0520 L015 L099 0523 0524
```

~ Input continues on next page ~

Himanshu, continued

```
0499 L091 S020 0496 L090 0494 0493 L114 0491 0490 0489 L104 0487 L039 L026 0484
   L097 0482 L029 0480 0479 0478 0477 L032 0475
S004 0451 0452 0453 L011 0455 L102 L073 L017 L093 L105 L047 L027 L126 0464 L069
   S012 0467 0468 S009 0470 0471 L113 0473 L084
L103 0448 0447 0446 L041 L103 L120 L098 0441 0440 0439 S007 0437 0436 0435 L083
   0433 L077 L129 L019 L102 0428 0427 0426 0425
L039 L136 0402 0403 0404 L028 0406 0407 0408 S020 L019 L115 0412 S019 0414 L121
   L049 L051 0418 0419 L134 L076 0422 0423 0424
0399 S003 L003 L099 0395 0394 0393 0392 L016 0390 0389 L130 0387 L005 L137 L109
   0383 0382 0381 0380 0379 0378 0377 S012 0375
0350 L086 0352 L120 L044 0355 S014 L059 L033 L082 0360 0361 L093 L124 L050 L111
   0366 L002 L001 0369 L078 L133 L037 L029 L100
S024 L111 L020 0346 L018 L107 0343 0342 L033 0340 L006 L077 L123 L017 0335 S015
   0333 0332 L067 L075 L060 0328 0327 0326 L091
0300 S022 0302 L013 L064 L045 0306 0307 0308 L074 L012 0311 0312 S004 0314 0315
   0316 L005 0318 0319 0320 L095 L101 0323 S011
0299 L101 0297 S021 L053 L031 0293 0292 L052 S016 0289 L088 L106 0286 0285 0284
   0283 L096 L070 0280 0279 L085 L112 0276 L138
L052 L040 0252 0253 0254 0255 0256 0257 0258 L130 0260 S021 0262 0263 0264 S018
   L124 0267 L055 L066 L098 0271 L034 0273 0274
L131 L118 L022 L007 S015 0244 L035 0242 0241 0240 L087 L061 L089 L121 0235 L085
   S017 0232 0231 0230 L023 0228 0227 0226 0225
0200 0201 L053 0203 0204 L131 0206 L056 L097 0209 0210 0211 0212 S005 L066 L044
   L040 L081 L110 L117 0220 0221 S018 L138 L024
0199 L004 L100 L035 L084 0194 L048 L030 S001 L128 L008 0188 S002 0186 L021 L094
   0183 0182 0181 0180 L117 L024 0177 0176 L109
0150 0151 S002 L074 0154 0155 L096 L065 0158 L069 L078 S013 L028 0163 0164 0165
   0166 S017 0168 L076 L004 L095 0172 L011 L128
0149 0148 0147 L068 0145 0144 0143 0142 0141 L089 L060 0138 L094 0136 0135 0134
   0133 L081 0131 0130 0129 0128 L056 0126 L043
L113 L079 0102 L134 S014 L136 L065 L122 0108 0109 0110 0111 L003 S024 L026 L092
   L007 0117 0118 L079 0120 0121 0122 0123 0124
L108 L025 L054 0096 0095 0094 0093 0092 L092 L012 0089 L116 S003 L006 L009 0084
   0083 0082 L058 L042 0079 0078 0077 0076 0075
0050 L021 L082 0053 L054 0055 L061 L073 L018 S005 0060 S009 0062 0063 L119 0065
   L010 L132 0068 L063 0070 L001 0072 L062 0074
0049 L062 L057 L115 0045 0044 L127 0042 S019 L070 L049 0038 L132 L137 L119 L034
   L086 L057 0031 0030 0029 L031 0027 L064 L058
0000 0001 L020 0003 L126 L032 0006 L046 L023 0009 L014 L015 0012 0013 0014 0015
   0016 L072 0018 0019 L010 S011 L114 L038 L125
P1:4,3,2,4,1,4,5,3,4,4,1,2,3,2,3,5,4,5,4,5,2,1,5,1,3,2,1,4,3,1,5,1,4,2,4,3,2,4,3
   ,3,4,2,5,4,4,1,5,5,4,4,1,5,3,5,2,2,1,4,5,1,1,1,1,3,5,2,5,1,1
P1:2,3,5,2,1,1,5,3,3,2,2,3,1,4,2,5,2,4,2,4,4,3,3,3,4,2,2,5,2,5,2,2,3,4,1,5,3,4,3
   ,3,1,3,5,5,5,2,3,4,1,1,2,5,2,1,2,1,2,4,4,2,2,2,5,5,2,2,3,5,2
-----------------------------------
16
255 S58 S37 S21 S67 L26 L10 L16 L47 S08 L31 L25 L29 S14 S23 S18
L03 S49 S11 S69 L09 S55 L04 S10 S08 S10 L22 L08 S48 S47 L16 L43
L36 S09 S55 L46 S30 L36 L10 S61 S46 S70 S63 S20 S23 S06 S41 S16
S24 S32 S65 L01 L29 L33 L43 L13 S57 S77 L28 L21 S54 S64 S69 S26
L01 L44 S44 L34 L33 S01 S40 S35 S28 L11 L02 S72 S71 S42 S02 S38
L17 S42 S43 S21 S43 L28 S12 S63 S01 S15 L14 S02 172 L27 L38 L08
```

~ Input and Output continues on next page ~

Himanshu, continued

```
S15 S36 S33 S34 S17 S74 S51 L39 S29 S05 149 L37 L15 S28 L42 S31
S19 L31 L06 L05 S26 S30 S54 S51 S22 S05 L19 L41 S35 S04 L07 L03
S45 S03 S59 L21 S64 L32 S59 S68 S57 S06 L24 L23 L20 L46 S45 S41
L04 S50 S52 S74 S38 L30 L24 S17 L07 S56 S19 L18 S72 S77 L20 L40
S60 S07 S16 S58 L12 L26 089 S32 S48 S40 L45 L17 S75 S33 L18 L12
S52 S24 S44 S36 L02 L25 S66 S50 L34 S07 L06 S27 S31 077 L39 L15
S53 S09 S76 S39 S18 L22 L35 S34 S70 L37 S49 L35 L19 S47 S04 S25
S60 S73 S11 L14 S14 L38 S62 S62 L23 S75 S37 S68 S61 S27 L44 L05
S12 S53 S13 028 L09 S39 S76 S73 L13 L40 L47 L30 L45 S56 S29 S03
000 S46 S67 S25 S13 L32 S22 S20 008 L41 L42 L27 L11 S71 S66 S65
P1:1,3,5,3,2,1,2,2,2,3,2,1,3,2,2,1,2,5,5,4,4,3,5,1,5,5,3,5,5,2,5,5
P1:1,4,4,3,5,2,5,4,5,3,2,5,1,5,2,3,4,3,2,1,1,4,1,2,4,5,5,3,1,4,4,5
-------------------------------------
8
63 62 L1 60 L2 58 L3 56
48 49 50 51 52 53 54 55
47 46 45 44 43 42 41 40
32 S3 L1 S2 L3 37 38 39
31 30 29 S1 S2 26 25 24
16 17 18 19 S3 21 22 23
15 14 13 12 S1 L2 09 08
00 01 02 03 04 05 06 07
P1:1,2,2
P1:4,2,5
-----------------------------------
2
03 L1
00 L1
P1:1,1
P2:2,2
-----------------------------------
```

Test Output To Screen:

```
Game #1: Player 2 wins!
Game #2: Player 2 wins!
Game #3: Neither Player 1 or Player 2 won.
Game #4: Player 1 wins!
Game #5: Neither Player 1 or Player 2 won.
Game #6: Neither Player 1 or Player 2 won.
Game #7: Player 2 wins!
Game #8: Player 2 wins!
Game #9: Player 1 wins!
Game #10: Neither Player 1 or Player 2 won.
Game #11: Both players win!
```

Problem #9
60 Points

9. Justin

Program Name: Justin.java **Input File: justin.dat**

Test Input File:

```
20
30987.41 2000 7.95 60
65432.78 3000 9.375 84
15937.07 1000 15.66 48
30460.35 1500 6.99 48
65749.40 2000 2.875 84
87898.33 3000 1.9375 96
22528.87 0 16.25 18
21457.69 0 14.6875 24
17896.94 500 9.3125 36
74320.34 2500 7.75 66
79983.68 2000 8.375 90
78827.88 3000 11.99 54
98884.92 2500 6.022 96
22626.44 0 3.625 24
47337.84 2000 2.49 48
90609.26 1500 19.899 90
48742.35 1500 14.1875 60
15020.27 2500 12.465 24
14784.05 500 1.75 18
96471.16 5000 10.99 96
```

Test Output To Screen:

```
$   587.07 $ 37224.20 $  6236.79
$  1016.41 $ 88378.44 $ 22945.66
$   420.72 $ 21194.56 $  5257.49
$   693.36 $ 34781.28 $  4320.93
$   838.75 $ 72455.00 $  6705.60
$   955.38 $ 94716.48 $  6818.15
$  1418.75 $ 25537.50 $  3008.63
$  1037.23 $ 24893.52 $  3435.83
$   555.75 $ 20507.00 $  2610.06
$  1339.99 $ 90939.34 $ 16619.00
$  1169.84 $107285.60 $ 27301.92
$  1823.78 $101484.12 $ 22656.24
$  1267.67 $124196.32 $ 25311.40
$   978.78 $ 23490.72 $   864.28
$   993.34 $ 49680.32 $  2342.48
$  1913.06 $173675.40 $ 83066.14
$  1103.84 $ 67730.40 $ 18988.05
$   592.10 $ 16710.40 $  1690.13
$   804.60 $ 14982.80 $   198.75
$  1436.37 $142891.52 $ 46420.36
```

Problem #10
60 Points

10. Marek

Program Name: Marek.java **Input File: marek.dat**

Test Input File:

```
15
1 + 2 - 1
4 * 7 - 3
3 / 2 + 1
4 + 2 * 3
3 * 5 * ( 8 * 7 ) * 4 / 31
212 + ( 212 / 212 )
5234 * 43 / 432
9503 + 19302 / 572 * 32 + 22 - 39193 + 3241 * 3
5 * 0
5 / 0
300 * 300 + 300 / 300
( 3 + 2 ) / 2
( 4 + 3 ) * ( 3 + 4 )
4 + 3 * 3 + 4
2 * 4 / ( 4 * 1 + 3 ) / 3
```

Test Output To Screen:

```
2
25
2
10
108
213
520
-18889
0
Infinity.
90001
2
49
17
0
```

Problem #11
60 Points

11. Priyanka

Program Name: Priyanka.java **Input File: priyanka.dat**

Test Input File:

```
10
UNIVERSITY
HELLO
ABCDEFGHIJ
EACH
I
M
TEXAS
COMPSCI
OILS
ASTRODOME
```

Test Output to Screen:

```
UIIENRSTVY
OEHLL
IEABCDFGHJ
WOO-HOO EACH
WOO-HOO I
WOO-HOO M
EASTX
OICCMPS
WOO-HOO OILS
OOEADMRST
```

Problem #12
60 Points

12. Sunny

Program Name: Sunny.java **Input File: sunny.dat**

Test Input File:

```
PHYS-2125 UNIVERSITY PHYSICS I LAB
MATH-1342 ELEMENTARY STATISTICAL METHODS
COSC-2325 COMPUTER ORGANIZATION
COSC-1336 PROGRAMMING FUNDAMENTALS I
PHYS-2325 UNIVERSITY PHYSICS I
COSC-1301 INTRODUCTION TO COMPUTING
COSC-1315 INTRODUCTION TO COMPUTER PROGRAMMING
COSC-1320 C PROGRAMMING
MATH-2313 CALCULUS I
COSC-1337 PROGRAMMING FUNDAMENTALS II
MATH-2314 CALCULUS II
MATH-1314 COLLEGE ALGEBRA
PHYS-1310 ELEMENTARY PHYSICS
PHYS-1101 COLLEGE PHYSICS I LAB
PHYS-1117 PHYSICAL SCIENCE II LAB
MATH-2315 CALCULUS III
PHYS-1305 ELEMENTARY PHYSICS I
PHYS-1304 SOLAR SYSTEM
COSC-2336 PROGRAMMING FUNDAMENTALS III
PHYS-1107 ELEMENTARY PHYSICS II LAB
PHYS-1315 PHYSICAL SCIENCE I
MATH-2312 PRE-CALCULUS MATH
MATH-1351 MATHEMATICS FOR TEACHERS II
PHYS-2326 UNIVERSITY PHYSICS II
PHYS-1301 COLLEGE PHYSICS I
PHYS-1105 ELEMENTARY PHYSICS I LAB
PHYS-1115 PHYSICAL SCIENCE I LAB
PHYS-1102 COLLEGE PHYSICS II LAB
MATH-2314 CALCULUS II
PHYS-1317 PHYSICAL SCIENCE II
PHYS-2126 UNIVERSITY PHYSICS II LAB
MATH-2318 LINEAR ALGEBRA
MATH-2320 DIFFERENTIAL EQUATIONS
MATH-1350 MATHEMATICS FOR TEACHERS I
PHYS-1307 ELEMENTARY PHYSICS II LAB
MATH-2305 DISCRETE MATHEMATICS
PHYS-1302 COLLEGE PHYSICS II
MATH-1316 PLANE TRIGONOMETRY
PHYS-1104 SOLAR SYSTEM LAB
```

Test Output To Screen:

```
C PROGRAMMING (COSC-1320)
CALCULUS I (MATH-2313)
CALCULUS II (MATH-2314)
CALCULUS II (MATH-2314)
CALCULUS III (MATH-2315)
COLLEGE ALGEBRA (MATH-1314)
COLLEGE PHYSICS I (PHYS-1301)
COLLEGE PHYSICS I LAB (PHYS-1101)
COLLEGE PHYSICS II (PHYS-1302)
COLLEGE PHYSICS II LAB (PHYS-1102)
COMPUTER ORGANIZATION (COSC-2325)
DIFFERENTIAL EQUATIONS (MATH-2320)
DISCRETE MATHEMATICS (MATH-2305)
ELEMENTARY PHYSICS (PHYS-1310)
ELEMENTARY PHYSICS I (PHYS-1305)
ELEMENTARY PHYSICS I LAB (PHYS-1105)
ELEMENTARY PHYSICS II LAB (PHYS-1107)
ELEMENTARY PHYSICS II LAB (PHYS-1307)
ELEMENTARY STATISTICAL METHODS (MATH-1342)
INTRODUCTION TO COMPUTER PROGRAMMING (COSC-1315)
INTRODUCTION TO COMPUTING (COSC-1301)
LINEAR ALGEBRA (MATH-2318)
MATHEMATICS FOR TEACHERS I (MATH-1350)
MATHEMATICS FOR TEACHERS II (MATH-1351)
PHYSICAL SCIENCE I (PHYS-1315)
PHYSICAL SCIENCE I LAB (PHYS-1115)
PHYSICAL SCIENCE II (PHYS-1317)
PHYSICAL SCIENCE II LAB (PHYS-1117)
PLANE TRIGONOMETRY (MATH-1316)
PRE-CALCULUS MATH (MATH-2312)
PROGRAMMING FUNDAMENTALS I (COSC-1336)
PROGRAMMING FUNDAMENTALS II (COSC-1337)
PROGRAMMING FUNDAMENTALS III (COSC-2336)
SOLAR SYSTEM (PHYS-1304)
SOLAR SYSTEM LAB (PHYS-1104)
UNIVERSITY PHYSICS I (PHYS-2325)
UNIVERSITY PHYSICS I LAB (PHYS-2125)
UNIVERSITY PHYSICS II (PHYS-2326)
UNIVERSITY PHYSICS II LAB (PHYS-2126)
```