Taylor High School

Anime Fest



Help Sasuke and his fellow anime friends solve the enigma of code!

| Problem 2.1 | Letter counter |
| --- | --- |
|  |  |
| General Statement: | Emperor Lelouch has a problem; he’s too busy conquering the world to deal with simple state matters. He needs to know how many consonants and vowels are in the names of the countries he has already conquered. Additionally, help him find the number of letters there are in all the names combined. |
|  |  |
| Input: | The first line of the input will be an integer *n* that represents the number of lines of input to follow. On each of the following lines will be names of the countries he has conquered; the first letter of each country name will be capitalized, and each country name will be only one word long. |
|  |  |
| Name of Data File: | adv21.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | How many vowels, consonants, and total characters are in the given country names, outputted in the following format:  Vowels: *v*  Consonants: *c*  Total: *t* |
|  |  |
| Assumptions: | All characters after the first line are letters of the English alphabet. There is one word per line. A, E, I, O, U are defined as vowels; Y is not considered a vowel. Both capital and lower case letters are included. |
|  |  |
| Sample Input: | 2  England  Brittania |
|  |  |
| Sample Output: | Vowels: 6  Consonants: 10  Total: 16 |

| Problem 2.2 | Graduation Exam! |
| --- | --- |
|  |  |
| General Statement: | Naruto is a ninja. He is not particularly intelligent; in fact, most would classify him as a grade-A nincompoop. He, however, is determined to graduate from the ninja academy after 3 years by writing the perfect essay about the finer points of the “Henge-No-Jutsu” on his graduation exam.  There is just one problem: his teacher, Iruka Sensei, specifically outlawed the use of the “no-no” words ‘is’, ‘are’, and ‘would’. For every instance of the aforementioned words, Iruka Sensei will deduct 0.5 points from Naruto’s grade. For example, if there were 4 of these words in Naruto’s essay, Naruto would get a 98. Help Iruka Sensei find out what grade he must give Naruto. |
|  |  |
| Input: | The first line of input will be an integer *n* that represents the number of lines to follow. Each of the following lines will be a sentence from Naruto’s essay. |
|  |  |
| Name of Data File: | adv22.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | On the first line: an integer representing the number of “no-no” words in the given sentences.  On the second line: Final grade: *x*  ,where *x* is an integer representing the final grade. |
| Assumptions: | All input is valid. Round the final grade to the nearest integer. (e.g. 97.5 = 98, 94.4 = 94). |
| Sample Input: | 3  Henge No Jutsu is dependent on various factors, influenced by the innate ability of the ninja casting it.  With the proper amount of chakra control, Henge No Jutsu is able to create a clone of anything the Ninja would wish; these clones are able to imitate in order to infiltrate other villages or areas.  Many people who are would be shinobi are forced to fail the Ninja Academy many times because of the “Henge No Jutsu”. |
| Sample Output: | 7  Final grade: 97 |

| Problem 2.3 | Halloween Candy |
| --- | --- |
|  |  |
| General Statement: | It is the day after Halloween and all candy was on sale. Yuki was looking for different candy bars to buy for Zero, who, as usual, was tormented by his bouts of angst and needed a variety of sweet treats to alleviate his awful mood. Help Yuki calculate the prices of the candy that she wants to buy. |
|  |  |
| Input: | The first line of input is an integer *n* that represents the number of data collections that follow. On the next *n* lines are the data collections, one on each line. The data collection consists of the price of the candy and the percent discount, separated by a single space. |
|  |  |
| Name of Data File: | adv23.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | Each line of output will be the final price. Round to 2 decimal places and place a $ before the final price. |
|  |  |
| Assumptions: | The price may be a double or an integer. The percent discount will be an integer. |
|  |  |
| Sample Input: | 3  12.99 35  34.50 10  99.89 65 |
|  |  |
| Sample Output: | $8.44  $31.05  $34.96 |

| Problem 2.4 | Scoring Averages |
| --- | --- |
|  |  |
| General Statement: | Kenpachi is the strongest captain of the 13 Court Guard Squads in the Soul Society. He needs to narrow down his list of applicants for his squad by the next captain’s meeting. He has already chosen some of the fighters that will definitely be on the squad. But the rest of the unfortunate persons are “on the bubble”. Help Kenpachi determine who will be on the squad. |
|  |  |
| Input: | The first line of the input is an integer *n* that represents the number of lines to follow. The next few lines will first feature the name of the applicant and his or her scores for 3 battles. The name and each of the scores will be separated from each other by a single space. |
|  |  |
| Name of Data File: | adv24.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | Output the name of the person with the highest average score and the highest average score like so:  Name is the best fighter with a scoring average of ##.## points per battle. (all on one line). The highest average score should be rounded to two decimal places. The output should be formatted exactly as shown below. |
|  |  |
| Assumptions: | All scores will be integers. |
|  |  |
| Sample Input: | 3  Ichigo 20 10 25  Uryuu 12 19 12  Hitsugaya 24 31 1 |
|  |  |
| Sample Output: | Hitsugaya is the best fighter with a scoring average of 18.67 points per battle. |

| **Problem 2.5** | **Persocon Vowel Training** |
| --- | --- |
|  |  |
| General Statement: | Hideki Motosuwa has a problem: if there are more than 20 vowels in a sentence, Chii cannot say it out loud. He needs your help to review sentences to make sure that Chii can say them. Your job is to decide the inputted sentences that Chii is able to pronounce. Your program will need to return which sentences she won’t say and how many vowels are in them. |
|  |  |
| Input: | The first line of input is an integer *n* that represents the number of data collections that follow. The next *n* lines will be sentences, each one on a separate line. |
|  |  |
| Name of Data File: | adv25.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | For every sentence Chii cannot say, output  Chii cannot say sentence x because it has y vowels.  *x* represents the sentence number, and *y* represents the number of vowels in sentence *x.* The first sentence in the input is sentence 1, the second is sentence 2, and so on. The output is to be formatted exactly as shown below. |
|  |  |
| Assumptions: | Vowels are defined as a, e, i, o, u, but not y. Capitalization in the output does not matter. |
|  |  |
| Sample Input: | 5  The fox jumps over the hound and enjoys a wonderful time in the jungle.  Wonderful melodies characterized the orchestral performance.  Joe made a great sandwich with the margarine and the peanut butter.  This program is so complex that it creates artificial intelligence.  It’s one in the morning and we are still awake and dreaming. |
|  |  |
| Sample Output: | Chii cannot say sentence 1 because it has 21 vowels.  Chii cannot say sentence 3 because it has 22 vowels.  Chii cannot say sentence 4 because it has 22 vowels. |

| **Problem 2.6** | **Out of Order!** |
| --- | --- |
|  |  |
| General Statement: | The leaders of the Black Order need to update their rosters due to the recent demise of most of their exorcists. Since all previous names were destroyed except the names Lavi, Lee, Walker, and Yu, all other names need to be inputted manually and sorted alphabetically. Because the Order's members are in disarray (no pun intended), spelling mistakes and repetition are likely to happen; therefore, every time a name is added to the roster, the roster is updated. In the case of repeated names, the list is not be updated (i.e. the name won’t be added if it’s already in the list). Help the leaders of the Black Order update their rosters. |
|  |  |
| Input: | The first line of input is an integer *n* that represents the number of data collections that follow. Each data collection consists of a single name. Each data collection is on a separate line. |
|  |  |
| Name of Data File: | adv26.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | Print out the roster every time a word is added, with brackets around the entire list and commas separating each name. A space should be inputted after each comma.  For example: [Lavi, Lee, Walker, Yu]  If a word is already in the roster, do not add it. Still, print out the roster. The output is to be formatted exactly like the sample output given below. |
|  |  |
| Assumptions: | Each line contains only one word, consisting of only letters from the English alphabet |
|  |  |
| Sample Input: | 5  Hevlaska  Lotto  Bookman  Krory  Han |
| Sample Output: | [Hevlaska, Lavi, Lee, Walker, Yu]  [Hevlaska, Lavi, Lee, Lotto, Walker, Yu]  [Bookman, Hevlaska, Lavi, Lee, Lotto, Walker, Yu]  [Bookman, Hevlaska, Krory, Lavi, Lee, Lotto, Walker, Yu]  [Bookman, Han, Hevlaska, Krory, Lavi, Lee, Lotto, Walker, Yu] |

| Problem 5.1 | Roman Madness! |
| --- | --- |
|  |  |
| General Statement: | Sousuke Sagara is an Arm Slave pilot for the mercenary group Mithril. As members of such a large and prestigious group, his teammates are numbered from 1 to around 3000. To save the space in the battle display, all of Sousuke's comrades are numbered using the Roman numeral system. Since Sousuke was raised in the war-torn Helmajistan, he, unfortunately, was not taught Roman numerals. Write a program to translate the Roman numerals into Arabic numbers to help Sousuke!  M=1000  D=500  C=100  L=50  X=10  V=5  I=1 |
|  |  |
| Input: | The first line of input will be an integer *n* that represents the number of lines to follow. Each of the next *n* lines contains a Roman numeral. |
|  |  |
| Name of Data File: | adv51.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | You will display the Arabic equivalent to the Roman numeral given in each line of input. The Arabic equivalent is an integer, and each integer should be on a separate line. The output is to be formatted exactly like the sample output given below. |
|  |  |
| Assumptions: | The largest Roman numeral given will be MMM (3000), and none of the Roman numerals given will contain more than 20 letters. |
| Sample Input: | 3  XXX  XLII  MCMLIII |
| Sample Output: | 30  42  1953 |

| Problem 5.2 | Black Lagoon comes to save the day |
| --- | --- |
|  |  |
| General Statement: | The leader of Hotel Moscow, Balalaika, has decided to reorganize the roster of mercenaries under her control. She has decided that the names of her officers should be alphabetized by the last letter of their names (because she's crazy) instead of the old way of alphabetizing things by their first letter. Balalaika has commissioned the Black Lagoon team, a 4-man mercenary team loyal to her, to construct a program to do it for her. Of course, these people are mercenaries and not java programmers, so they are forcing you to do it. Be afraid. |
|  |  |
| Input: | The first line of input is an integer *n* that represents the number of data collections to follow. For each data collection, the first line will contain an integer *x* that represents how many names are to follow. On the next x lines are the names to be alphabetized, each on a separate line. |
|  |  |
| Name of Data File: | adv52.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | The output is on one line, with commas in between each word, in the following format:  According to Balalaika : name1, name2, name3  The output is to be formatted exactly like the sample output given below. |
|  |  |
| Assumptions: | All input will be valid. |
|  |  |
| Sample Input: | 2  5  Test  Revy  Dutch  Rock  Benny  3  Kisuke  Ichigo  Orihime |
|  |  |
| Sample Output: | According to Balalaika : Dutch, Rock, Test, Benny, Revy  According to Balalaika : Kisuke, Orihime, Ichigo |

| Problem 5.3 | It’s Counting! |
| --- | --- |
|  |  |
| General Statement: | Lupin III is an international thief. He has just discovered a new type of password system which is based on the occurrence of words in select phrases, and he is able to "coerce" his enemies to give him. Whilst cracking this password, Lupin realizes that this will become an extremely popular system. Therefore, he thinks that having someone write a program which cracks the password for him will allow for faster and more effortless heists. It is your job, as a kidnapped programmer, to accomplish this task. Good Luck. |
|  |  |
| Input: | The first line of the input will be an integer *n* that represents the number of lines to follow. Each of the next *n* lines contains sentences; each word of the sentence is separated by a single space. |
|  |  |
| Name of Data File: | adv53.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | You will display the words in the set alphabetically on separate lines. Each word will be followed by the number of occurrences of each word in the total input. Print out 5 stars between the word and the number.  (e.g. EXAMPLE\*\*\*\*\*7)  The output is to be formatted exactly like the sample output below. |
|  |  |
| Assumptions: | There are only alphabetical characters in a word (i.e. no punctuation). All input is in capital letters. |
| Sample Input | 4  THIS IS A SIGN  THIS IS ANOTHER SIGN  CAN YOU READ THIS STOP STARING AT ME  GO GO GO GO GO GO |
| Sample Output  CHANGED | A\*\*\*\*\*1  ANOTHER\*\*\*\*\*1  AT\*\*\*\*\*1  CAN\*\*\*\*\*1  GO\*\*\*\*\*6  IS\*\*\*\*\*2  ME\*\*\*\*\*1  READ\*\*\*\*\*1  SIGN\*\*\*\*\*2  STARING\*\*\*\*\*1  STOP\*\*\*\*\*1  THIS\*\*\*\*\*3  YOU\*\*\*\*\*1 |

| Problem 5.4 | Nom Nom! |
| --- | --- |
|  |  |
| General Statement: | *In a nonexistent world, the popularity of a city was based on the usage of its area code when dialed on a rotary phone. The usage is determined by calculating how many times the dial must be turned from 0 to reach the specified number. In other words, to dial ‘2’, the dial must be turned 2 times, since it is 2 away from 0, so the amount of use would be 2. For the area code, ‘281’, the usage is 11, or 2+8+1.*  Monkey D. Luffy, a famous pirate of the Straw Hat Crew, is trying to figure out where he should be travelling to make himself, and his crew, better known throughout the world. Luffy decides that the best way to determine this is to gauge the popularity of each city. Unfortunately, Luffy does not have the patience to write a program to do so and is busy eating, so the task is left to you. |
|  |  |
| Input: | The first line of the input will be an integer *n* that represents the number of lines to follow. Each of the next *n* lines will have the name of a city and then its area code, separated by a single space. |
|  |  |
| Name of Data File: | adv54.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | The list of names and area codes sorted from low to high according to popularity in the form ‘p City Name a’, where p is the popularity and a is the area code.  To calculate the popularity of each city, determine the usage of the area code as if it was dialed on a rotary phone (refer to explanation in general statement above). If two cities have the same popularity, order them alphabetically. The output is to be formatted exactly like that for the sample output given below. |
|  |  |
| Assumptions: | The city name may be more than one word. |
| Sample Input | 6  Dallas 214  Fort Worth 817  Lubbock 806  Austin 512  Denton 940  Houston 248 |
| Sample Output  CHANGED | 7 Dallas 214  8 Austin 512  13 Denton 940  13 Houston 248  14 Lubbock 806  16 Fort Worth 817 |

| Problem 5.5 | The 'W' of Woe |
| --- | --- |
|  |  |
| General Statement: | The Japanese police force is pursuing a Contractor who is writing the name of his victim in the shape of a 'W' into the ground to taunt the police. Hei, a Contractor who hunts other Contractors, and his partner Amber, who is able to look into the future, decide to help the police by anonymously leaving a note with the name of the next victim in the shape of the all too familiar 'W' at the scene of the previous crime. Since Contractors do not think of such odd ways of arranging letters, it is up to you, a programmer who has been captured by the Syndicate, to make a program to re-arrange the names into a ‘W’ shape for them. |
|  |  |
| Input: | The first line of the input will be an integer *n* that represents the number of lines to follow. Each of the next *n* lines will contain the first name of the next victim. |
|  |  |
| Name of Data File: | adv55.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | Output the name, in all caps, in the shape of a ‘W’. The ‘W’ will be 5 characters tall, and the name will be written continuously as shown below. Each segment of the W is 5 characters long, printed in staggered manner. There must be one blank line between each ‘W’.  The output is to be formatted exactly like the sample output below. |
|  |  |
| Assumptions: | All input will be valid. |
| Sample Input | 3  NISH  FRANK  VIVASWATH |
| Sample Output | N N N  I H I H  S S S S  H I H I  N N  F N R  R A K F  A R F K  N F R N  K A  V H T  I T V A  V A I W  A W V S  S A |

| Problem 5.6 | Do the who and the what now? |
| --- | --- |
|  |  |
| General Statement: | Gene Starwind is given a sentence, written in a secret code, that he must decipher in order to unlock one of his Caster Shells and blow his enemies to oblivion. He is also given *n* keys to break the code, numbered 1 through *n*; he must figure out which key to use to break the code. The length of the first word of the sentence given divided by the length of the last word of the sentence determines which key will be used. For example, if the lengths are divided and the quotient is 1, then the 1st key given in the input is the correct key. Each letter in the key corresponds to a letter in the key. Thus, the first letter in the key is equivalent to ‘A’, the second to ‘B’, and so on. Then he must decode the sentence. Because he is incapable of understanding any of this, it is up to you, Gilliam II, the artificial intelligence system in Gene's ship, the Outlaw Star, to figure this out and save the crew. |
|  |  |
| Input: | The first line of input will be a number *n* indicating the number of data collections to follow. For each data collection, the first line is the sentence, written in a secret code. On the next line is *x*, the number of keys, and the next *x* lines arethe keys. Each key will be on one line in all caps, with no spaces in between each letter. After each data collection will be a line of space. |
|  |  |
| Name of Data File: | adv56.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | The decoded sentence is outputted. The output is to be formatted exactly like the sample output given below. |
|  |  |
| Assumptions: | All input and output will be in uppercase |
|  |  |
| Sample Input:  CHANGED | 2  LYMS YM VBBP  3  AEVNUQRSYZOPKLBCDWMTFGHIJX  YZHIVWSOXJNQKLMEDUFGABCPRT  CUMNOVWHABZPQDFGLXYRSTIJKE  OEIYMBGX AOJGPOG JA DJQG  4  XBQRSUOYZKIJDCELMFGPAVTHWN  CNOFGQRSJKLDIPEYZXABMWTUVH  AINOJKLSTUMBGHCDXYZFWERVPQ  HIEBDPJNRAQXTUFGSVWOCMKLYZ |
|  |  |
| Sample Output: | NISH IS COOL  COMPUTER SCIENCE IS LIFE |

| Problem 9.1 | AND… I MEAN OR! NO, XOR! AAH! BOOLEANS!!!  BZZT! \*CRASH\* |
| --- | --- |
|  |  |
| General Statement: | *The cardinal operators of Computer Science, AND (&), OR (|), XOR (^), and NOT (!), and parentheses have long dominated the domain of java programming.*  Joe Shimamura, a prototype 00 Cyborg, wishes to make a program to successfully evaluate the Boolean operators after his 8 other companions have overheated and shut down due to the sheer confusion caused by trying to evaluate Boolean equations in a testing session! Although Joe has the brainpower to figure out Boolean expressions, he cannot express it as a java program (since he is programmed in C--). As a researcher for the 00 experiment, write a program to solve the Boolean operations before Cyborg 009 malfunctions. |
|  |  |
| Input: | The first line of input is an integer *n* that represents the number of data collections that follow. Each data collection contains a Boolean expression of unspecified length; each data collection is on a separate line. The uppercase characters T and F will be used in the Boolean expression to represent the terms TRUE and FALSE, respectively. The order of operations is as follows:  () Evaluate the expression in parentheses first. Parentheses may be nested; evaluate from the inner most set of parentheses first.  ! Evaluate the term immediately following the NOT operator according to the rules !T=F and !F=T  & Evaluate the terms immediately preceding and following the AND operator according to the rules T&T=T, T&F=F, F&T=F, and F&F=F  ^ Evaluate the terms immediately preceding and following the XOR operator according to the rules T^T=F, T^F=T, F^T=T, and F^F=F  | Evaluate the terms immediately preceding and following the OR operator according to the rules T|T=T, T|F=T, F|T=T, and F|F=F  No characters beside TF()!&^| will appear in the lines of input. |
|  |  |
| Name of Data File: | adv91.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | Your program should produce *n* lines of output (one for each data collection). Each line should consist of only one uppercase character, either T or F, representing the result of the corresponding Boolean expression given in the input.  The output is to be formatted exactly like the sample output given below. |
| Assumptions: | The value for *n* will not exceed 1000.  The ! operator will immediately precede only the characters ‘(‘ ‘T’ or ‘F’.  The length of the inputted expression will not exceed 500 characters.  The input may or may not have an operator.  Nested parentheses may be used. |
|  |  |
| Sample Input: | 5  T&F|F&F|T&(F|!F)  F&T  !F  !T^F  ((!T)) |
|  |  |
| Sample Output: | T  F  T  F  F |

| Problem 9.2 | The Lottery! |
| --- | --- |
|  |  |
| General Statement: | Dante loves the lottery! This lottery game, however, is a bit more complicated than the average, as it is a more complex (and rewarding) back-alley lottery!  For this lottery, the winning lottery ticket will consist of 6 numbers (not necessarily distinct) selected from the lottery set {1, 2 …49}.  Some bright competitors developed a strategy to win the lottery with a theory that they believe will increase their chance of winning. Their strategy is to select a subset S containing *k* numbers from the lottery set, provided that 12 > *k* > 6. Then, the competitors will determine each possible combination of 6 numbers from the subset S, called ‘tickets’.  For example, if *k* = 8 and S = 1,2,3,5,8,13,21,34, then there are 28 possible lottery tickets:  [1,2,3,5,8,13], [1,2,3,5,8,21], [1,2,3,5,8,34], [1,2,3,5,13,21], ... , [3,5,8,13,21,34].  Your job is to write a program for Dante that prints out all the distinct lottery tickets that can be created by choosing 6 numbers from the subset S so that he can guarantee a winning ballot in the lottery and fund his Devil May Cry business without doing any real work. |
|  |  |
| Input: | The first line of input is an integer *n* that represents the number of data collections that follow. For each data collection, the 1st line of input is a number *k*, representing the number of numbers in subset S. On the following line is subset S, with each number of the subset separated from each other by a single space. The numbers of S are in ascending order. |
|  |  |
| Name of Data File: | adv92.dat |
|  |  |
| Time Allocation: | 3 second |
|  |  |
| Output: | Output first the number of distinct lottery tickets; then, on the following lines, output each possible lottery ticket. Each possible lottery ticket should consist of 6 numbers, ordered from least to greatest with a single space between each number. A line of space should be outputted between each test case.  The numbers in each ticket must be sorted in ascending order and separated from each other by exactly one space. The tickets themselves must be sorted lexicographically (least to greatest). (i.e. the game {1,4,6,7,8} is less than the game {1,4,6,7,9})  The output is to be formatted exactly like the sample output given below. |
| Assumptions: | The number *k* is between 6 and 12, exclusive. All numbers in the subset S are in the lottery set. |
| Sample Input: | 2  7  1 3 4 7 11 18 29  8  1 2 3 5 8 13 21 34 |
| Sample Output: | 7  1 3 4 7 11 18  1 3 4 7 11 29  1 3 4 7 18 29  1 3 4 11 18 29  1 3 7 11 18 29  1 4 7 11 18 29  3 4 7 11 18 29  28  1 2 3 5 8 13  1 2 3 5 8 21  1 2 3 5 8 34  1 2 3 5 13 21  1 2 3 5 13 34  1 2 3 5 21 34  1 2 3 8 13 21  1 2 3 8 13 34  1 2 3 8 21 34  1 2 3 13 21 34  1 2 5 8 13 21  1 2 5 8 13 34  1 2 5 8 21 34  1 2 5 13 21 34  1 2 8 13 21 34  1 3 5 8 13 21  1 3 5 8 13 34  1 3 5 8 21 34  1 3 5 13 21 34  1 3 8 13 21 34  1 5 8 13 21 34  2 3 5 8 13 21  2 3 5 8 13 34  2 3 5 8 21 34  2 3 5 13 21 34  2 3 8 13 21 34  2 5 8 13 21 34  3 5 8 13 21 34 |

| Problem 9.3 | Bases?! Inconceivable! |
| --- | --- |
|  |  |
| General Statement: | According to Haruhi Suzumiya, numbers are simply too annoying to group by base. To ameliorate this problem, she has ordered you to make a program that denotes the smallest possible base a number can have. Don’t disappoint her, for she will unleash beings from parallel universes to punish you! |
|  |  |
| Input: | The first line of input will be an integer *n* that represents the number of data collections to follow. Each of the following lines will contain a number *x* of an unspecified base. |
|  |  |
| Name of Data File: | adv93.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | You are to output the lowest possible number base of the given numbers in the following form: BASE x , where x is the lowest possible base. For example, the number 2134 could be at least in base 5. A7 could be at least in base 11. On the next line is the same number converted to base 10. The output is to be formatted exactly like the sample output below. |
|  |  |
| Assumptions: | Bases will never exceed 16 or be smaller than 2. All integers will be greater than or equal to 0. All letters will be capitalized. |
| Sample Input: | 4  123  5090  A7  1010 |
| Sample Output: | BASE 4  27  BASE 10  5090  BASE 11  117  BASE 2  10 |

| Problem 9.4 | What’s your function? |
| --- | --- |
|  |  |
| General Statement: | Dr. Riddles and Kido are trying to explain the concept of derivatives to their college Calculus class. Since Mr. Riddles knows all, and because he wears a monocle, he knows exactly how to do derivatives. But because his class will not acknowledge his prominent mustache and his dashing good looks, he does not feel like teaching such commoners. He has chosen you, because you are a student of his and are currently in need of extra credit, to create a program to solve derivative equations. |
|  |  |
| Input: | The first line of the input will be an integer *x* that represents the number of data collections to follow. On each of the next *x* lines, the number *n* and the function f(x) will be given, separated by a single space. *n* represents how many times you need to take the derivative of f(x). |
|  |  |
| Name of Data File: | adv94.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | For each function given, output the 1st derivative of the function on the 1st line, then the 2nd derivative on the next line, the 3rd derivative on the next line, and so on until you output the *n*th derivative.  The function should be ordered by degree, from greatest to least, with constant terms placed at the end.  Output a line of space between each data collection.  **Derivative hints:**  To find the derivative of a function:   1. Take the exponent of each x term and multiply it with the coefficient of that same x term. 2. Then reduce the exponent of that x term by 1.   For example, if you were to take the derivative of 3x^2+7x, then you would multiply 2 by 3, which is 6. Reduce the exponent, 2, by 1 to get 1. Repeat the process for the x term 7x to get 7. Thus, the derivative of 3x^2+7x is 6x+7. **\*\*\*(Warning: This explanation is for programming purposes only!)**  The 1st derivative of a function is when you take the derivative of f(x) for the first time. The 2nd derivative is when you take the derivative of the derivative of f(x), and so on.  Using the previous example, the 2nd derivative of 3x^2+7x would be 6, since 6 is the derivative of 6x+7, which was the derivative of 3x^2+7x. The 3rd derivative would be 0, the 4th would be 0, and ect. |
| Assumptions: | 1*. n* will be greater than 0 and an integer.  2. The degree of f(x)will be between -10 and 20 inclusive.  3. Coefficients of x terms may be negative or positive, ranging from –∞  to ∞.  4. f(x)may or may not have an x term; it may be simply a constant.  5. Each x term in f(x)will have distinct powers (i.e. there will not be 2  or more x terms with an exponent of 2).  6. The function f(x)in the input will be ordered from greatest to least  (by x terms), with the constant term at the end.  7. If the x term does not have a ‘^’ symbol, it is implied to have a power  of one. Therefore, when taking the derivative of an x term with a  power of 1, the x term will be eliminated. (e.g. the derivative of 6x is  6, not 6x^0)  8. For constant terms, it is implied that the x term is x^0. Thus, the  derivative of a constant term will be 0, since 0 times any number is 0.  9. If there is a constant term equal to 0, do not include it in the output.  10. If the exponent of an x term is negative and the coefficient of that x  term is positive, then the coefficient of the derivative of that x term  will become negative, since a negative times a positive is negative.  (e.g. the derivative of 2x^-2 is -4x^-3, and the derivative of  3x^-1 is 3x)  11. Output must be formatted exactly like the sample output below. |
| Sample Input: | 2  1 f(x)=x^2+3x-7  3 f(x)=5x^4-6x^2-7x^-1+9 |
| Sample Output: | 2x+3  20x^3-12x+7x^-2  60x^2-14x^-3-12  120x+42x^-4 |

| Problem 9.5 | Don’t be afraid to express-ion yourself! |
| --- | --- |
|  |  |
| General Statement: | Inuyasha is curious about how many times he has actually defeated Naraku. Upon questioning his friends, only the ominous and lecherous Miroku actually remembers the exact number of times. Being the pervert he is, he offers Inuyasha an ultimatum: solve the arithmetic equation to find out the number of times he has defeated Naraku, or use Kagome's camera to take photos of Sango in the bath. Of course, Inuyasha picks the former option and consults Kagome. Since Kagome is terrible at math, she consults her computer back in the future, and since Yahoo! answers and Google could not yield a proper answer, she has decided to make a program to figure out the equation instead. Help Kagome write a proper program to solve any of the equations Miroku proposes. |
|  |  |
| Input: | The first line of the input will be an integer that represents the number of data collections to follow. On the following line is the expression you must solve |
|  |  |
| Name of Data File: | adv95.dat |
|  |  |
| Time Allocation: | 1 second |
|  |  |
| Output: | The answer to the arithmetic expression, rounded to one decimal place. |
|  |  |
| Assumptions: | The only mathematical operations used will be addition, subtraction, multiplication, division, symbolized by +,-,\*,/, respectively. **Warning**: nested parentheses may be used. The numbers in the expression are integers greater than or equal to 0. Answers must be rounded to one decimal place. Answers may be negative or even undefined. If it is undefined, print out NaN. Numbers in the input range from -∞ to ∞ |
| Sample Input: | 2  4+6\*8/2-9\*4  6\*((6-4)/2)+8 |
| Sample Output: | -8.0  14.0 |

| Problem 9.6 | It’s A-maze-ing! |
| --- | --- |
|  |  |
| General Statement: | Seras Victoria is trapped in a 3 dimensional maze with various obstacles placed throughout the maze. Alucard, her master, is given the coordinates of her position and that of 3 doors, *A, B,* and *C*. Which ever one of these doors is closest to Seras is the one with the exit. He, and his master Sir Integra Hellsing, must direct Seras out of the maze. She may not move diagonally, and thus can only move up, down, left, right, forwards, and backwards. |
| Input: | The first integer *n* indicates how many data collections are to follow.  In each data collection:  On the first line of the data collection (i.e. the line after *n*) are the dimensions of the maze, given in the form *X*x*Y*x*Z*.  On the next line are Seras’ coordinates, given as *x y z*, each separated by spaces; Seras’ coordinates correspond to the (x, y, z) coordinate on a three dimensional coordinate plane.  On the next three lines are the coordinates of the doors *A, B,* and *C,* respectively, given in the same format as Seras’ coordinates.  On the next line is an integer *k* that represents the number of obstacles in the maze.  In the next *k* lines are the coordinates of *k* obstacles, given in the same format as the doors and Seras’ coordinates. |
|  |  |
| Name of Data File: | adv96.dat |
|  |  |
| Time Allocation: | 3 second |
|  |  |
| Output: | The distance from Seras’ position to the exit and the letter of that door. The distance should be an integer and, when outputted, should be separated from the letter of the correct door with a comma. (e.g. 3,B)  If more than one door has the same distance from Seras’ starting point, and that distance happens to be the shortest path length, then there is no exit; output “Sir Integra laughs at your feeble attempts at escape!” |
|  |  |
| Assumptions: | All coordinates are within the boundary of the maze. The dimensions of the maze will not exceed 50x50x50 and will be greater than 0x0x0. The coordinates of the door and Seras’ position will not overlap with those of the obstacles. Any of the doors and Seras’ location may be the same. All coordinates are given under the assumption that they are represented in a three-dimensional coordinate plane, with x, y, and z axis. |
| Sample Input: | 2  5x5x5  0 0 0  2 0 0  0 4 0  0 0 3  6  3 2 1  4 3 2  0 4 1  2 2 3  3 1 0  0 1 4  9x9x9  4 0 0  4 8 0  4 0 8  1 2 7  10  7 6 5  1 4 0  3 6 7  5 4 6  1 8 2  2 3 1  4 8 1  2 4 3  0 0 1  2 3 4 |
| Sample Output: | 2,A  Sir Integra laughs at your feeble attempts at escape! |