

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '86

1.1 Write a program that will first clear the screen and then center on the screen the sentence: **THIS IS THE EASIEST PROGRAM!**

1.2 Write a program that will accept two integers between -170 and 170, inclusive, and will print out the sum, difference (second from first), and product, each clearly labeled. Note: one or two spaces may separate the equal sign from each number output. Examples:

INPUT: Enter two numbers: **6, 8**

OUTPUT: **SUM = 14**
DIFFERENCE = -2
PRODUCT = 48

1.3 Consider the series

$$1 + (1/2)^2 + (1/3)^3 + (1/4)^4 + \dots$$

where there are infinitely many terms. Write a program to compute the sum of the series until the difference between the present sum and the sum obtained by including just one more term is less than a test value, E, given by the operator. Output must be printed in the form **#####**. Example:

INPUT: Enter test value E: **0.01**

OUTPUT: **1.287037**

1.4 Write a program for Ben's Towing Service for the purpose of writing checks. The payee's first name, middle name or initial, last name, and amount will be entered separately by the operator. The program should then print a check surrounded by a border of asterisks (39 x 11) with Ben's business address at the top left, as shown below. The middle name must be abbreviated to an initial whether the operator enters the full middle name or just an initial. The middle initial must have a period immediately following. The first and last names combined will not exceed 10 letters. The amount entered must appear immediately after a dollar sign. Example:

INPUT: Enter first name: **NORM**
Enter middle name: **N**
Enter last name: **SAND**
Enter amount: **1234.56**

OUTPUT: (appears on next page)

```

OUTPUT: *****
*
* BEN'S TOWING SERVICE
* 4563 WRECKER AVENUE
* WAVERLY, ARKANSAS 45632
*
* PAY TO THE ORDER OF NORM N. SAND
*
* THE SUM OF $1,234.56
*
*****
    
```

1.5 A jail has 100 cells. In celebration of the warden's birthday, some of the prisoners will be released today. First the warden opens all the cell doors. Then starting with the first door, he closes every other door. Then, starting with the first door, he goes to every third door, closing the open doors and opening the closed ones; That is, doors 1, 4, 7, etc. He repeats this process of opening closed doors and closing open doors for doors 1,5,9, etc.; i.e. for every 4th door starting with the first door. He continues doing this for every 5th door, every 6th door, etc., always starting with the first door, until he has done this for every 100th door. Prisoners whose door is open when he is done are permitted to leave. Write a program that determines who can leave, and display each cell number in the format shown below:

```

OUTPUT: CELL 2
        CELL 5
        CELL ###
         :
         :
        CELL ###
    
```

1.6 Jill is thinking about opening a retirement account. Write a program to help her figure how much she can accumulate. The operator should be asked to input the monthly investment, the end of the year lump sum investment and the rate of interest. Assume that the monthly investment is deposited on the first of each month and that interest is compounded at the end of each month. Also assume that the yearly investment is deposited at the end of each year after the month's interest has been added. Calculate the total amount (rounded to the nearest penny) in the account at the end of twenty years. Example:

```

INPUT: Enter Monthly investment: 100
       Enter end of year deposit: 600
       Enter annual rate of interest: 12
    
```

```

OUTPUT: AMOUNT AT END OF YEAR 20 IS $146715.74
    
```

1.7 Write a program which will teach your computer to print with a Southern accent. For purposes of this program, this means that for any sentence which is input (without a period), the last "g" of any word ending with ing or with ings must be dropped. Example:

INPUT: Enter sentence: **THOSE EARRINGS ARE TO MY LIKING**
OUTPUT: **THOSE EARRINS ARE TO MY LIKIN**

1.8 Assume that rabbits reproduce at the rate of 20% per month until the end of the month that overpopulation occurs, at which time they begin dying at the rate of 15% per month. Rabbits continue to die at this rate until their population is reduced by one third of the amount in which over population occurs, at which time they begin reproducing again. Write a program to simulate the rabbit population for 23 months, displaying each amount rounded to the nearest whole number. The program will accept as input the initial population and the number at which overpopulation occurs. Example:

INPUT: Enter initial population: **1826**
Enter point of over population: **2600**

OUTPUT: **POPULATION FOR MONTH 1 IS 2191**
POPULATION FOR MONTH 2 IS 2629
POPULATION FOR MONTH 3 IS 2235
POPULATION FOR MONTH 4 IS 1900
POPULATION FOR MONTH 5 IS 1615
POPULATION FOR MONTH 6 IS 1938
POPULATION FOR MONTH 7 IS 2325
POPULATION FOR MONTH 8 IS 2790
POPULATION FOR MONTH 9 IS 2372
POPULATION FOR MONTH 10 IS 2016
POPULATION FOR MONTH 11 IS 1714
POPULATION FOR MONTH 12 IS 2056
POPULATION FOR MONTH 13 IS 2468
POPULATION FOR MONTH 14 IS 2961
POPULATION FOR MONTH 15 IS 2517
POPULATION FOR MONTH 16 IS 2139
POPULATION FOR MONTH 17 IS 1819
POPULATION FOR MONTH 18 IS 2182
POPULATION FOR MONTH 19 IS 2619
POPULATION FOR MONTH 20 IS 2226
POPULATION FOR MONTH 21 IS 1892
POPULATION FOR MONTH 22 IS 1608
POPULATION FOR MONTH 23 IS 1930

1.9 The inhabitants of Eeland write normally with the exception of the fact that the letter E is always doubled when it is used. Write a program which will convert a normally written English sentence into an Eeland sentence. Example:

INPUT: Enter sentence: **SHE WORKS LIKE A BEE.**
OUTPUT: **SHEE WORKS LIKEE A BEE.**

1.10 Write a program which will accept a list of 12 whole numbers and a list of 11 whole numbers by entering one number at a time. The program will then print out the common elements of the two lists without repetition, with each number separated by 2 spaces and in the order that they appear in the first list. Example:

```
INPUT: Enter 1 of 12: 5
       Enter 2 of 12: 6
       Enter 3 of 12: 7
       Enter 4 of 12: 9
       Enter 5 of 12: 8
       Enter 6 of 12: 9
       Enter 7 of 12: 0
       Enter 8 of 12: 23
       Enter 9 of 12: 456
       Enter 10 of 12: 789
       Enter 11 of 12: 1
       Enter 12 of 12: 12
```

```
Enter 1 of 11: 1
Enter 2 of 11: 23
Enter 3 of 11: 32
Enter 4 of 11: 23
Enter 5 of 11: 8
Enter 6 of 11: 99
Enter 7 of 11: 9
Enter 8 of 11: 10
Enter 9 of 11: 12
Enter 10 of 11: 8
Enter 11 of 11: 1
```

```
OUTPUT: 9 8 23 1 12
```

2.1 Write a program that takes a sentence and right-justifies it on a line that has 65 columns. When the sentence is right-justified, it begins in column 1 and ends in column 65. The spacing between words should be approximately uniform. Example:

INPUT: Enter sentence: **THIS IS REALLY A SHORT SENTENCE.**

OUTPUT: (The sentence is to be right-justified on a 65 column line. Spacing between words is approximately uniform.)

THIS IS REALLY A SHORT SENTENCE.

2.2 Write a program that produces the following repeating pattern:

```

XXX--XXX--XXX--XXX--
---XX---XX---XX---XX
XXX--XXX--XXX--XXX--
---XX---XX---XX---XX

```

Note that on each line the X and - pattern is repeated four times. Also, the second line has -'s where the first line has X's and it has X's where the first line has -'s. The third and fourth lines repeat the pattern in the first two lines. Write the program so that the operator puts in the total number of X's and -'s used in the basic sequence, the number of X's used in the beginning basic sequence, and the number of rows to be printed (less than 20). Example:

INPUT: Enter total number of X's and -'s: **5**
Enter number of X's: **3**
Enter number of rows: **5**

OUTPUT: **XXX--XXX--XXX--XXX--**
---XX---XX---XX---XX
XXX--XXX--XXX--XXX--
---XX---XX---XX---XX
XXX--XXX--XXX--XXX--

2.3 You just got a job with a secret agency as a message coder/decoder. Write a program which gives the operator the choice of: 1) coding a message from English into a secret code; or 2) decoding a message from secret code into English. The master translation string is "ZXCVBNMQASDFGHJKLQWERTYUIOP " for each of the corresponding letters of the alphabet. Notice that the space is the last character of the translation string which always translated to a space. Only these characters will be used. The program is to display a menu of three choices as shown below and will continue to prompt for input until a '3' is entered. Example:

```
OUTPUT: 1) ENCODE
        2) DECODE
        3) END
INPUT:  Choose: 1
        Enter message: HERE IS A MESSAGE
OUTPUT: ABWB SE Z HBEEZMB

        1) ENCODE
        2) DECODE
        3) END
INPUT:  Choose: 2
        Enter message: ABWB SE Z HBEEZMB
OUTPUT: HERE IS A MESSAGE

        1) ENCODE
        2) DECODE
        3) END
INPUT:  Choose: 3
OUTPUT: (program terminates)
```

2.4 The mode of a set of numbers is the most frequently used number. Write a program in which the operator is asked to input 15 numbers. The program then finds and prints the unique mode if only one mode; otherwise have the computer print the message NO UNIQUE MODE. Examples:

INPUT: Enter number 1: 7	INPUT: Enter number 1: 1
Enter number 2: 1	Enter number 2: 2
Enter number 3: 4	Enter number 3: 3
Enter number 4: 9	Enter number 4: 4
Enter number 5: 5	Enter number 5: 5
Enter number 6: 4	Enter number 6: 6
Enter number 7: 5	Enter number 7: 7
Enter number 8: 7	Enter number 8: 8
Enter number 9: 1	Enter number 9: 5
Enter number 10: 5	Enter number 10: 1
Enter number 11: 22	Enter number 11: 2
Enter number 12: 33	Enter number 12: 3
Enter number 13: 44	Enter number 13: 4
Enter number 14: 55	Enter number 14: 5
Enter number 15: 4	Enter number 15: 6

OUTPUT: **NO UNIQUE MODE** OUTPUT: **MODE IS 5**

2.5 A bank needs a program to handle savings accounts. The original balance in an account will be entered at the terminal. The annual interest rate on deposits will be 7%. The rest of the program will be menu driven. Choices will be: 1. MAKE A DEPOSIT, 2. MAKE A WITHDRAWAL, 3. CREDIT INTEREST, or 4. END the program. The amount of a deposit or a withdrawal will be entered at the terminal. After each transaction, the program must print the balance before the transaction, the type of transaction (deposit, withdrawal, or interest) with the amount of the transaction, and the new balance after the transaction (always above \$1500). When crediting interest, the current balance is the amount on which to figure the month's interest. When the "end" choice is selected, the final balance must be printed before terminating. Example:

```
INPUT: Enter original balance: 2345.15
OUTPUT: 1. MAKE A DEPOSIT
        2. MAKE A WITHDRAWAL
        3. CREDIT INTEREST
        4. END
INPUT: Enter option: 1
        Enter amount to deposit: 100
OUTPUT: BALANCE BEFORE TRANSACTION $2,345.15
        MAKE A DEPOSIT
        NEW BALANCE $2,445.15
        1. MAKE A DEPOSIT
        2. MAKE A WITHDRAWAL
        3. CREDIT INTEREST
        4. END
INPUT: Enter option: 2
        Enter amount to withdraw: 50
OUTPUT: BALANCE BEFORE TRANSACTION $2,445.15
        MAKE A WITHDRAWAL
        NEW BALANCE $2,395.15
        1. MAKE A DEPOSIT
        2. MAKE A WITHDRAWAL
        3. CREDIT INTEREST
        4. END
INPUT: Enter option: 3
OUTPUT: BALANCE BEFORE TRANSACTION $2,395.15
        CREDIT INTEREST OF $ 13.97
        NEW BALANCE $2,409.12
        1. MAKE A DEPOSIT
        2. MAKE A WITHDRAWAL
        3. CREDIT INTEREST
        4. END
INPUT: Enter option: 4
OUTPUT: FINAL BALANCE $3,409.12
```

2.6 Write a program in which the operator may input any two positive integers (with at most 38 digits each), and the program then prints the sum of these two numbers. Example:

INPUT: Enter first number: **54387698325431256754**
Enter second number: **76589787321212546786**

OUTPUT: **SUM IS 130977485646643803540**

2.7 Write a menu driven program which can perform decimal/metric conversions in each direction with the choices below. Each of the 12 choices should allow an amount to be entered and then print the equivalent amount in the other system of measure. Each amount displayed must be in the form ###.###. A 13th choice will end the program. The menu must be presented in two columns with six choices in the first column and the other six in the second column (with the two-digit numbers beginning in column 27) as shown below. Note:

1 inch = 2.54 centimeters
1 foot = 0.3048 meters
1 mile = 1.6093 kilometers
1 ounce = 28.35 grams
1 pound = 0.4536 kilograms
1 gallon = 3.7854 liters

Example:

OUTPUT: 1 CENTIMETERS TO INCHES 7 GRAMS TO OUNCES
2 INCHES TO CENTIMETERS 8 OUNCES TO GRAMS
3 METERS TO FEET 9 KILOGRAMS TO POUNDS
4 FEET TO METERS 10 POUNDS TO KILOGRAMS
5 KILOMETERS TO MILES 11 LITERS TO GALLONS
6 MILES TO KILOMETERS 12 GALLONS TO LITERS
13 END

INPUT: Enter option: 2
Enter number of INCHES: 100

OUTPUT: **THIS IS EQUIVALENT TO 254.000 CENTIMETERS**

OUTPUT: 1 CENTIMETERS TO INCHES 7 GRAMS TO OUNCES
2 INCHES TO CENTIMETERS 8 OUNCES TO GRAMS
3 METERS TO FEET 9 KILOGRAMS TO POUNDS
4 FEET TO METERS 10 POUNDS TO KILOGRAMS
5 KILOMETERS TO MILES 11 LITERS TO GALLONS
6 MILES TO KILOMETERS 12 GALLONS TO LITERS
13 END

INPUT: Enter option: 10
Enter number of POUNDS: 10

OUTPUT: **THIS IS EQUIVALENT TO 4.536 KILOGRAMS**

OUTPUT: (menu displays again)

INPUT: Enter option: 13

OUTPUT: (program terminates)

2.8 Write a program to generate a mortgage amortization where the user inputs the principal, the interest rate, the term in years, and the number of the month during which the first payment is made. Have the program print the amount of interest paid for each month and the outstanding principal for that month. At the end of each year the program is to print the total amount of interest paid during that year and pause until a key has been pressed so that all the output may be viewed on the screen before proceeding to the next set of amounts. At the end of the loan, have the program print the total interest paid and the calculated monthly payment. Each amount must be rounded to the nearest penny and displayed in the form ###.## for monthly interest and #####.## for all others. Note: The formula for monthly payment is:

$$PA = (R*(1+R)^{(Y*12)})/((1+R)^{(12*Y)}-1)*P$$

where R is decimal interest,
Y is term in years,
and P is principal.

Note: The program will not compute accurate amounts if the power symbol (^) is used. The final principal must be -0.00 or 0.00.

A partial listing of a sample output is given below:

```

INPUT: Enter principal: 10000
       Enter % rate of interest: 10
       Enter term in years: 4
       Enter # of month in year for first payment: 4
OUTPUT: INTEREST      PRINCIPAL
        $ 83.33      $ 9829.71
        $ 81.91      $ 9658.00
        $ 80.48      $ 9484.85
        $ 79.04      $ 9310.27
        $ 77.59      $ 9134.23
        $ 76.12      $ 8956.72
        $ 74.64      $ 8777.73
        $ 73.15      $ 8597.26
        $ 71.64      $ 8415.27

        YEAR'S INTEREST $ 697.91

        $ 70.13      $ 8231.78
                :
                :
        YEAR'S INTEREST $ 229.58

        $ 6.24      $ 500.98
        $ 4.17      $ 251.53
        $ 2.10      $ -0.00

        YEAR'S INTEREST $ 12.51
        TOTAL INTEREST  $ 2174.04
        MONTHLY PAYMENT $ 253.63

```

2.9 The value of sine x is given by the series

$$x - x^3/3! + x^5/5! - x^7/7! + x^9/9! - x^{11}/11! + \dots$$

where x is measured in radians and the series has an infinite number of terms. Write a program that calculates the sine of N degrees, once N is entered as a number greater than 0 and less than 360. Perform the calculation so that x is added to $-x^3/3!$ and the sum is added to $x^5/5!$ and so on. The last term to be added will be $-x^{11}/11!$. Have the computer print the sum followed by the actual value of sine of N degrees as calculated by the computer's internal sine function. Each value is to be displayed in the form **#.#####**.

Note: Using the power sign (^) could give an inaccurate result.
Note: When N is greater than 180 use the reference angle of $360 - N$ for accurate results, but be careful with the sign.
Note: π is approximately equivalent to 3.1415926535.

Examples:

INPUT: Enter N degrees: **125**

OUTPUT: **PARTIAL SUM = 0.8191481**
ACTUAL SINE = 0.8191520

INPUT: Enter N degrees: **270**

OUTPUT: **PARTIAL SUM = -0.9999999**
ACTUAL SINE = -1.0000000

2.10 The seven Roman numerals and their corresponding values in the Arabic system are $M = 1000$, $D = 500$, $C = 100$, $L = 50$, $X = 10$, $V = 5$, and $I = 1$. The Roman system is generally the sum of the values of the digits as long as the digits decrease to the right. That is, MDLXXVII would be 1577. Whenever a digit of smaller value appears to the left, it is subtracted from the digit to its immediate right. Only C, X, and I may subtract. C will subtract from M or D, X from C or L, and I from X or V. The number MCDXLIX is 1449. Write a program to accept as input a Roman numeral and then translate the Roman numeral to an Arabic numeral. Examples:

INPUT: Enter Roman Numeral: **MMCMXCIX**
OUTPUT: **ARABIC = 2999**

INPUT: Enter Roman Numeral: **CCCXXXVIII**
OUTPUT: **ARABIC = 338**

3.1 Write a program that produces monthly calendars for the year 1986, one month at a time, starting with January. For each calendar the name of the month must be approximately centered above the calendar days. An abbreviation of the names of each day (starting with Sunday) must be on the top of each column of day numbers. Only the month of January needs the heading '1986'. After one monthly calendar is displayed, allow the user to press any key to clear the screen and to display the next month. Each month should have a neat form similar to the calendar of January as shown below.

Hint: remember the rhyme, 30 days have September, April, June, and November. All the rest have 31 except February which has 28 and on a leap year 29. Partial listing of example:

1986

JANUARY

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

FEBRUARY

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	

3.2 A fundamental theorem of mathematics says that every 5th degree polynomial has at least one root (a value of the variable at which the polynomial equals 0).

Write a program which will find a root for any 5th degree polynomial in the variable x (whose root lies between -10 and 10), once given the values of the coefficients of the polynomial:

$$Ax^5 + Bx^4 + Cx^3 + Dx^2 + Ex + F = 0$$

The root value x must be displayed in the form #.##### with the 5th digit on the right of the decimal being rounded. Example:

INPUT: Enter coefficients A,B,C,D,E,F: 1, -5, 8, 5, -4, 3
 OUTPUT: **ROOT IS -0.93629**

3.3 Write a program which changes a base A numeral into a base B numeral, where A and B are any integers between 2 and 36, inclusive. Examples:

```
INPUT: Enter base A: 12
      Enter base B: 11
      Enter original number: A14B
```

```
OUTPUT: A14B BASE 12 EQUALS 12154 BASE 11
```

```
INPUT: Enter base A: 36
      Enter base B: 27
      Enter original number: 1Y
```

```
OUTPUT: 1Y BASE 36 EQUALS 2G BASE 27
```

3.4 A number of DATA statements contain customer accounts. For each customer, there are four fields: social security number in the form XXXXXXXXXX, name, address, and account balance. Store the customer data into associated arrays. Additional input will come from the terminal and will consist of three items: social security number, C for charge or P for payment, and amount of transaction.

The social security number is used to determine whose balance is affected; the balance is then updated; and the new balance is printed. The process is repeated until the social security number 000000000 is entered. After that, all data is to be printed in a report showing social security number, name, address, and the final balance for each customer. The customer listing must be arranged in decreasing order according to the account balance. Balances are displayed in the form ####.##. Use the DATA statements given below.

```
DATA 234567890,JOHN SMITH,
      "1234 ANYWHERE LANE, EXIST, KANSAS 66754",345.78
DATA 564783219,GAIL HUSTON,
      "543 SOUTH THIRD, BIG TOWN, TEXAS 88642",2365.89
DATA 873421765,TIM JONES,
      "2387 PALM PLACE, NOME, ALASKA 77643",6754.76
DATA 543876543,JILL RUPERTS,
      "4536 123RD STREET, TINY TOWN, MAINE 76765",45.18
DATA 345212342,AL BROWN,
      "PO BOX 234, TINSEL TOWN, CALIFORNIA 77654",3456.09
DATA 565656565,KERMIT TEU,
      "1234 LOST LANE, WIMPLE, WISCONSIN 66543",78.36
```

Example:

```
INPUT: Enter SSN: 564783219
      Enter C for charge or P for payment: C
      Enter amount of transaction: 10
```

```
OUTPUT: NEW BALANCE IS $2355.89
(Input/Output continued on next page)
```

(Input/Output continued)

INPUT: Enter SSN: **565656565**
 Enter C for charge or P for payment: **P**
 Enter amount of transaction: **10**

OUTPUT: **NEW BALANCE IS \$ 88.36**

INPUT: Enter SSN: **000000000**

OUTPUT: SSN	NAME	ADDRESS	BALANCE
873421765	TIM JONES	2387 PALM PLACE NOME ALASKA 77643	\$6754.76
345212342	AL BROWN	PO BOX 234 TINSEL TOWN CALIFORNIA 77654	\$3456.09
564783219	GAIL HUSTON	543 SOUTH THIRD BIG TOWN TEXAS 88642	\$2355.89
234567890	JOHN SMITH	1234 ANYWHERE LANE EXIST KANSAS 66754	\$ 345.78
565656565	KERMIT TEU	1234 LOST LANE WIMPLE WISCONSIN 66543	\$ 88.36
543876543	JILL RUPERTS	4536 123RD STREET TINY TOWN MAINE 76765	\$ 45.18

3.5 Write a program to print out the product of any two decimal numbers, each containing at most 20 digits and a decimal point. Example:

INPUT: Enter first number: **1.123456789**
 Enter second number: **0.11239**

OUTPUT: **PRODUCT = 0.12626530851571**

INPUT: Enter first number: **123456789.123456789**
 Enter second number: **1000000000.0000000001**

OUTPUT: **PRODUCT = 123456789123456789.123456789123456789**

3.6 A palindrome is a word, verse, or number that reads the same backwards or forwards. The number 12321 is a palindrome. Here is a method of forming palindromic numbers which is almost always successful: Begin with any positive integer. If it is not a palindrome, reverse its digits and add the two numbers. If the sum is not a palindrome, treat it as the original number and continue until a palindrome is generated. Write a program to find palindromes using this method; However, allow a limit of at most 23 additions. For example, 196 will not produce a palindrome, so a limit is necessary. If a number will not produce a palindrome after 23 additions, then print out the message: CANNOT GENERATE A PALINDROME. Note: the generated palindrome could be up to 15 digits long. In the example below, 5405 can generate a palindrome because $5405 + 5045 = 10450$ and $10450 + 5401 = 15851$.

Example:

INPUT: Enter number: **5405**
OUTPUT: **15851 IS A PALINDROME**

INPUT: Enter number: **196**
OUTPUT: **CANNOT GENERATE A PALINDROME**

3.7 Write a program to solve a system of N equations with N unknowns. The program will first accept a value for N between 2 and 4 inclusive. For each equation the program should accept the values for the N coefficients followed by the constant. Each given system of equations will have one unique solution whose values will be integers. A possible algorithm to be used to solve an NxN system of equations is shown to the right of the example input/output. Examples:

INPUT: Enter N: 3	$2x + 4y + 6z = 4$ (Divide by 2)
Enter coefficients for row1	$2x + y + z = 3$
Col: 2	$-x + 2y + z = 2$
Co2: 4	
Co3: 6	1 2 3 2
Enter constant: 4	2 1 1 3 (Subtract 2*Row1)
Enter coefficients for row2	-1 2 1 2
Col: 2	
Co2: 1	1 2 3 2
Co3: 1	0 -3 -5 -1
Enter constant: 3	-1 2 1 2 (Subtract -1*Row1)
Enter coefficients for row3	
Col: -1	1 2 3 2
Co2: 2	0 -3 -5 -1 (Divide by -3)
Co3: 1	0 4 4 4
Enter constant: 2	
OUTPUT: (1, 2, -1)	1 2 3 2
	0 1 5/3 1/3
	0 4 4 4 (Subtract 4*Row2)
	1 2 3 2
INPUT: Enter N: 2	0 1 5/3 1/3
Enter coefficients for row1	0 0 -8/3 8/3 (Divide by -8/3)
Col: 1	
Co2: 1	1 2 3 2
Enter constant: 1	0 1 5/3 1/3 (Subtract 5/3*Row3)
Enter coefficients for row2	0 0 1 -1
Col: 2	
Co2: 3	1 2 3 2 (Subtract 3*Row3)
Enter constant: 6	0 1 0 2
	0 0 1 -1
OUTPUT: (-3, 4)	
	1 2 0 5 (Subtract 2*Row2)
	0 1 0 2
	0 0 1 -1
	1 0 0 1
	0 1 0 2
	0 0 1 -1

3.8 Write an EFFICIENT program to accept as input a word with L distinct letters (L is less than 8) and a positive integer K. Your program is to print out the Kth, 2*Kth, and 3*Kth elements of the alphabetized list of all the permutations of the word. Each permutation is to be separated by two spaces. For example, if the word CAT is entered with K=2 then the computer would form the list ACT, ATC, CAT, CTA, TAC, TCA, and output would be: ATC CTA TCA. NOTE: No team completed this program last year. Can anyone do it this year!

Example:

INPUT: Enter word: **CAT**
OUTPUT: **ATC CTA TCA**

3.9 Write an EFFICIENT program to solve the following cryptarithm puzzle where each unique letter represents a unique digit:

$$ABB - CB = DEF$$

CB is a 2-digit number and is subtracted from the 3-digit number ABB to obtain the 3-digit result DEF. Have the program print every unique solution, with each one numbered, and the total number of solutions printed at the end. The solutions do not have to appear in any particular order, but they must appear in the format below. A partial listing of an example:

OUTPUT: 411 - 21 = 390 **NUMBER 1**
511 - 21 = 490 **NUMBER 2**
611 - 21 = 590 **NUMBER 3**
711 - 21 = 690 **NUMBER 4**
:
:
:
477 - 97 = 380 **NUMBER ###**
577 - 97 = 480 **NUMBER ###**
677 - 97 = 580 **NUMBER ###**

TOTAL NUMBER OF SOLUTIONS = ###

3.10 Write a program that will find all positive two digit integers that can exist as the sum of the digits 0-9 in which each of the digits are used exactly once. Any two of the digits may be combined to form a 2-digit integer to be added with the other digits. Output must include every qualified number in ascending order followed by ONE OF THE SUM PROCESSES that qualifies the number. Order of the addends does not matter. A partial example:

OUTPUT: **45 = 0 + 1 + 2 + 3 + 4 + 5 + 7 + 8 + 9**
:
72 = 10 + 23 + 4 + 5 + 6 + 7 + 8 + 9
: