

FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '93

1.1 Write a program to display the following six lines consisting of the acronym for GTE Data Services Inc.:

```

GTEDS GTEDS GTEDS GTEDS GTEDS GTEDS
GTEDS  GTEDS  GTEDS  GTEDS  GTEDS
GTEDS   GTEDS   GTEDS   GTEDS
GTEDS    GTEDS    GTEDS
GTEDS     GTEDS
GTEDS

```

1.2 Every couple of months, GTE Data Services hires qualified individuals for their New Recruit Development program. This program features 14 and 15 week curriculums designed to train programmers in the standards and procedures of GTE Data Services.

The course is "pass/fail" and successful students are placed in programmer positions in the company with a competitive starting salary. Entrance into the program is very competitive since GTEDS pools from as many as 300 candidates each period to select approximately 15 individuals by reviewing resumes, conducting telephone interviews, administering aptitude tests, and holding panel interviews.

Write a program to determine how many PROGRAMMERS have been placed in the company if N classes, starting with 15 students, have graduated with a total of M students dropping from the course. Examples:

```

INPUT: Enter N: 19
        Enter M: 9

```

```

INPUT: Enter N: 10
        Enter M: 5

```

```

OUTPUT: 276 PROGRAMMERS

```

```

OUTPUT: 145 PROGRAMMERS

```

1.3 GTE is the largest U.S.-based local-telephone company with domestic and international operations serving more than 20.7 million access lines in 40 states, Canada, and Latin America.

Write a program to display the formatted numeric figure of N million access lines, where N is input as a real number greater than 1 and less than 1000 and contains at most 4 digits. Examples:

```

INPUT: Enter N: 20.7

```

```

OUTPUT: 20,700,000 ACCESS LINES

```

```

INPUT: Enter N: 3.456

```

```

OUTPUT: 3,456,000 ACCESS LINES

```

1.4 The University of South Florida (USF) is ranked as the second largest university in the Southeast, having had a fall 1992 enrollment of over 34,000 students. Approximately half of these students are full-time and the other half are part-time. Approximately 57% of the student population is female. USF is located on five campuses: Tampa, St. Petersburg, Fort Myers, Lakeland, and Sarasota.

Write a program to accept as input the number of students attending each of the campuses and display the total number of STUDENTS enrolled at USF. Example:

```
INPUT: Enter # at Tampa: 27319
       Enter # at St. Petersburg: 3010
       Enter # at Fort Myers: 1329
       Enter # at Lakeland: 856
       Enter # at Sarasota: 1846
```

```
OUTPUT: 34360 STUDENTS
```

1.5 GTE Data Services selects qualified candidates to participate in the Information Systems Orientation Program (ISOP). Selection is made from all individuals whose current position is at least a Salary Grade Level 5 and who have the potential and desire to occupy at least the positions of Systems Supervisor and Systems Manager.

Write a program to accept the NAME of a selected candidate, their Salary Grade LEVEL, and whether they DESIRE to occupy at least the positions of Systems Supervisor and Systems Manager (YES or NO). Display a statement indicating whether or not the person entered is A POSSIBLE CANDIDATE FOR ISOP. Examples:

```
INPUT: Enter name: SCOTT
       Enter level: 6
       Enter desire: YES
```

```
OUTPUT: SCOTT IS A POSSIBLE CANDIDATE FOR ISOP
```

```
INPUT: Enter name: DOUG
       Enter level: 3
       Enter desire: YES
```

```
OUTPUT: DOUG IS NOT A POSSIBLE CANDIDATE FOR ISOP
```

```
INPUT: Enter name: JOE
       Enter level: 7
       Enter desire: NO
```

```
OUTPUT: JOE IS NOT A POSSIBLE CANDIDATE FOR ISOP
```

1.6 The New Recruit Development program features either a C/UNIX or MVS/COBOL curriculum based on the business needs of GTE Data Services. Some of the selection criteria used to hire employees for this program include:

- BA/BS in Computer Science, Math, Business, Engineering, MIS or CIS preferred
- Courses in any programming language
- Demonstrated interest in and commitment to information Management
- Experience in mainframe, mini or microcomputers
- Demonstrated leadership skills
- Strong GPA/Performance history
- Computer-related work experience
- Effective oral and written communication skills
- Career development potential
- Team player

If GTEDS is hiring for the C/UNIX curriculum then the following is a list of additional preferred skills: C, UNIX, ANSI SQL, OSF/MOTIF, SHELL PROGRAMMING.

If GTEDS is hiring for the MVS/COBOL curriculum, then the following is a list of additional preferred skills: COBOL, JCL, MVS/ESA, TSO/ISPF, VSAM, ANSI SQL, DB2, IMS.

Write a program to enter the curriculum for which the recruiter is hiring and display all the preferred skills for that curriculum. Examples:

INPUT: Enter curriculum: **MVS/COBOL**

OUTPUT: **COBOL**
JCL
MVS/ESA
TSO/ISPF
VSAM
ANSI SQL
DB2
IMS

INPUT: Enter curriculum: **C/UNIX**

OUTPUT: **C**
UNIX
ANSI SQL
OSF/MOTIF
SHELL PROGRAMMING

1.7 Write a program to display the first N letters of the alphabet, where N is input as a number less than or equal to 26. Examples:

INPUT: Enter N: 6

OUTPUT: **ABCDEF**

INPUT: Enter N: 25

OUTPUT: **ABCDEFGHIJKLMNOPQRSTUVWXYZ**

1.8 Although money is not the primary reason why people remain computer programmers, an increase in salary does provide extra motivation. An employee's salary usually increases in proportion to his/her performance on the job. Write a program to accept a computer programmer's salary (greater than 20,000 and less than 90,000) and his/her "performance rating" and then display the new salary to be received (rounded to the nearest penny) based on the percentage rates listed below:

<u>Performance</u>	<u>Increase</u>
EXCELLENT	10%
ABOVE AVERAGE	7%
GOOD	5%

Examples:

INPUT: Enter salary: 25000.50
Enter rating: **EXCELLENT**

OUTPUT: **NEW SALARY = \$27500.55**

INPUT: Enter salary: 50000
Enter rating: **ABOVE AVERAGE**

OUTPUT: **NEW SALARY = \$53500.00**

INPUT: Enter salary: 30500.19
Enter rating: **GOOD**

OUTPUT: **NEW SALARY = \$32025.20**

1.9 When a GTE customer requests either telephone service, modifications to an existing service, or disconnection of service, the telephone company enters a "Service Order" into their computer system called SORCES (Service Office Record and Computer Entry System). This service order is then sent to the Customer Billing Services System (CBSS), which is an advanced and highly flexible customer billing system that produces detailed bills that are easy to read and understand.

Under Mike's direction, Russ and Scott supervise the development and maintenance of the Service Order sub-application programs within CBSS. Six of the most commonly used types of orders that these programs receive from SORCES are listed below:

- I = INSTALL - an initial order to establish service for a new customer.
- C = CHANGE - an order to make a change to existing service, such as adding a pricing plan, or changing a phone number, or having the desk phone removed and a wall phone added.
- R = RECORDS - an order to create a non-service affecting change to the customer's account. Examples include: name changes, address corrections, and changing the day when a customer receives his bill.
- O = OUT - an order issued to disconnect service.
- F = FROM - an order to disconnect service at an old address (similar to an 'O' order) and is the first half of an F & T combination.
- T = TO - an order to install service at a new location (similar to an 'I' order) and is the second half of an F & T combination.

Write a program to display the Service Order abbreviation if given the descriptive order, and to display the descriptive order if given the abbreviation for an order. Examples:

INPUT: Enter order: **INSTALL**
OUTPUT: **I**

INPUT: Enter order: **F**
OUTPUT: **FROM**

INPUT: Enter order: **I**
OUTPUT: **INSTALL**

INPUT: Enter order: **CHANGE**
OUTPUT: **C**

1.10 The average high school grade point average of the freshman class at the University of South Florida is 3.24. At USF the GPA is derived by dividing the total grade points by the total credits.

Write a program to enter 5 class grades, each having the same number of credits, and display the GPA for the semester. The values of the grades are as follows:

A = 4,
B = 3,
C = 2,
D = 1,
F = 0,
I = 0,
M = 0,
W is a withdrawal from the class.

A grade of 'I' (incomplete) and 'M' (missing) are counted as a grade of 'F' until resolved; however, a grade of 'W' (withdrawal) is not tallied since the class has been dropped (one less class of credits is used for dividing the points). The GPA must be displayed with 3 digits to the right of the decimal and rounded to the nearest 0.001. Examples:

INPUT: Enter grade: **A**
Enter grade: **C**
Enter grade: **F**
Enter grade: **I**
Enter grade: **B**

OUTPUT: **GPA = 1.800**

Note: In this example $(4 + 2 + 0 + 0 + 3) / 5 = 1.800$.

INPUT: Enter grade: **C**
Enter grade: **M**
Enter grade: **A**
Enter grade: **W**
Enter grade: **D**

OUTPUT: **GPA = 1.750**

Note: In this example $(2 + 0 + 4 + 1) / 4 = 1.750$. The grade points are divided by 4 classes instead of 5 classes because 1 class was dropped, indicated by a grade of 'W'.

2.1 Write a program to randomly generate N numbers between X and Y inclusive, where N is input as an integer less than 11 while X and Y are input as integers between -99 and 99 inclusive. Display all numbers on one line with one space separating each number. Possible examples (outputs will vary):

```
INPUT: Enter N: 5
       Enter X, Y: -15, -4
```

Possible OUTPUT: -13 -5 -15 -5 -8

```
INPUT: Enter N: 10
       Enter X, Y: 55, -1
```

Possible OUTPUT: 23 34 0 16 55 2 6 53 17 28

```
INPUT: Enter N: 5
       Enter X, Y: -4, -15
```

Possible OUTPUT: -6 -11 -7 -4 -12

2.2 Write a program to help produce an organizational work-chart for N people within a department at GTEDS. Input will consist of technician names, each followed by his/her grade level title. The names and titles are to be displayed in descending order by title and then ascending order by name. The grade level titles are as follows (from lowest to highest): P, PA, SA, SE, SSE, ASE, SASE.

```
INPUT: Enter N: 6
       Enter name: PAULA
       Enter title: SA
       Enter name: MARION
       Enter title: ASE
       Enter name: KEVIN
       Enter title: P
       Enter name: DON
       Enter title: SE
       Enter name: DOUG
       Enter title: P
       Enter name: DERRIL
       Enter title: SE
```

```
OUTPUT: MARION - ASE
        DERRIL - SE
        DON - SE
        PAULA - SA
        DOUG - P
        KEVIN - P
```

2.3 Write a program to accept as input a COBOL declaration of a record and display the fields indented appropriately. Each field begins with a two digit level number. Position all fields beginning with level number 01 in the first column while each successive field is indented 4 spaces more than the previous field if the level number is greater than the previous level number; if the level number is less than the previous level number then the field is indented 4 spaces less than the previous field; and if the level number is the same as the previous level number then the field is lined up in the same column as the previous field. The first line input will begin with the level number 01, and the last line input will be a blank line. Examples:

```
INPUT: Enter field: 01 WS-NAME PIC X(15).
      Enter field: 01 WS-ADDRESS.
      Enter field: 05 WS-STREET PIC X(20).
      Enter field: 05 WS-CITY-ST.
      Enter field: 10 WS-CITY PIC X(20).
      Enter field: 10 WS-STATE PIC X(02).
      Enter field: 07 WS-ZIP-9.
      Enter field: 15 WS-ZIP PIC X(05).
      Enter field: 15 WS-ZIP-4 PIC X(04).
      Enter field: 10 FILLER PIC X(10).
      Enter field: 20 FILLER PIC X(15).
      Enter field: 01 WS-DATA PIC X(80).
      Enter field: (Press the Enter key)
```

```
OUTPUT: 01 WS-NAME PIC X(15).
        01 WS-ADDRESS.
          05 WS-STREET PIC X(20).
          05 WS-CITY-ST.
            10 WS-CITY PIC X(20).
            10 WS-STATE PIC X(02).
          07 WS-ZIP-9.
            15 WS-ZIP PIC X(05).
            15 WS-ZIP-4 PIC X(04).
          10 FILLER PIC X(10).
            20 FILLER PIC X(15).
        01 WS-DATA PIC X(80).
```

2.4 Write a program to translate a word into a number by substituting each letter with its position in the alphabet, and then display the amount of blocks of contiguous even digits and odd digits within the number. Example:

INPUT: Enter word: **CONTEST**

OUTPUT: **NUMBER = 315142051920**
BLOCKS = 4

INPUT: Enter word: **WAY**

OUTPUT: **NUMBER = 23125**
BLOCKS = 4

Note: the first example has 4 blocks: 3151, 420, 519 and 20;
the second example has 4 blocks: 2, 31, 2, and 5.

2.5 Write a program to enter N telephone numbers and display them with dashes in the appropriate positions: after the NPA (the first three digits), and after the NXX (the next three digits). Display a blank line between numbers with the same NPA but a different NXX, and display 2 blank lines between numbers with a different NPA. Numbers will be input in ascending order. Display the total amount of telephone numbers within an NPA on the last line displayed within that section. Examples:

INPUT: Enter N: **8**
Enter #: **1234567890**
Enter #: **1234568907**
Enter #: **1235678901**
Enter #: **1235679012**
Enter #: **1235679999**
Enter #: **2345678901**
Enter #: **3456789012**
Enter #: **3457890123**

OUTPUT: **123-456-7890**
123-456-8907

123-567-8901
123-567-9012
123-567-9999 TOTAL FOR NPA OF 123 = 5

234-567-8901 TOTAL FOR NPA OF 234 = 1

345-678-9012

345-789-0123 TOTAL FOR NPA OF 345 = 2

2.6 Write a program to calculate the cost of purchasing several grocery store products (A - Z), where a person has one or more coupons for products (A - Z). The first part of the input will consist of a set of products with their prices (more than \$1 and less than \$10), ended by entering a fictitious product (9). The second part of the input will consist of a set of coupons for products with their discounts (less than \$1), ended by entering a fictitious coupon (9). Output will be the TOTAL dollar amount of the bill (not including tax), after all applicable coupons have been used. If there is more than one coupon for the same product bought, then the largest coupon discount will be used. A coupon discount for a product may only be used once. Example:

```
INPUT: Enter product: F
       Enter price: 1.50
       Enter product: B
       Enter price: 1.75
       Enter product: C
       Enter price: 5.00
       Enter product: B
       Enter price: 1.75
       Enter product: 9

       Enter coupon: B
       Enter discount: 0.50
       Enter coupon: D
       Enter discount: 0.60
       Enter coupon: C
       Enter discount: 0.75
       Enter coupon: B
       Enter discount: 0.65
       Enter coupon: B
       Enter discount: 0.70
       Enter coupon: 9
```

OUTPUT: **TOTAL = \$7.90**

Note: $7.90 = 1.50 + (1.75 - 0.70) + (5.00 - 0.75) + (1.75 - 0.65)$

```
INPUT: Enter product: C
       Enter price: 8.50
       Enter product: B
       Enter price: 7.75
       Enter product: 9

       Enter coupon: B
       Enter discount: 0.50
       Enter coupon: 9
```

OUTPUT: **TOTAL = \$15.75**

2.7 There are three standard formats for the DATE data type in a particular COBOL system:

YYYY-MM-DD	ISO (International Standards Organization)
MM-DD-YYYY	AMERICAN
DD-MM-YYYY	EUROPEAN

Write a program to accept as input a date in one of the three formats and display the date in the other two formats in the order given in the above list. Examples:

INPUT: Enter format: **AMERICAN**
Enter date: **02-04-1993**

OUTPUT: **ISO = 1993-02-04**
EUROPEAN = 04-02-1993

INPUT: Enter format: **ISO**
Enter date: **1994-12-31**

OUTPUT: **AMERICAN = 12-31-1994**
EUROPEAN = 31-12-1994

INPUT: Enter format: **EUROPEAN**
Enter date: **16-10-1993**

OUTPUT: **ISO = 1993-10-16**
AMERICAN = 10-16-1993

2.8 Write a program to reverse the order of words in one or two sentences input, where each word within a sentence is separated by one space and each sentence is ended by a period. Two spaces will separate the first and second sentences. Examples:

INPUT: Enter sentence: **I AM HAPPY. SO ARE YOU.**

OUTPUT: **HAPPY AM I. YOU ARE SO.**

INPUT: Enter sentence: **I AM ENJOYING THIS COMPUTER CONTEST.**

OUTPUT: **CONTEST COMPUTER THIS ENJOYING AM I.**

2.9 Write a program to enter a 4 x 4 matrix of positive integers less than 10 and display the 4 smallest elements and their location(s) within the matrix. If a smallest element occurs more than once, list the locations in ascending order by rows and then by columns, each separated by a comma. Example:

```
INPUT: Enter row 1: 1, 3, 5, 7
      Enter row 2: 9, 8, 3, 1
      Enter row 3: 6, 7, 8, 9
      Enter row 4: 9, 1, 5, 5
```

```
OUTPUT: 1. SMALLEST = 1 OCCURS AT (1,1), (2,4), (4,2)
        2. SMALLEST = 3 OCCURS AT (1,2), (2,3)
        3. SMALLEST = 5 OCCURS AT (1,3), (4,3), (4,4)
        4. SMALLEST = 6 OCCURS AT (3,1)
```

2.10 GTE Data Services Inc. is a wholly owned subsidiary of GTE Corporation based in Stamford, Connecticut. GTEDS is one of the largest software development and information processing service companies in the United States, using 15 large mainframes to support approximately 125 major software systems. With corporate headquarters in Temple Terrace, Florida (next to Tampa), GTEDS is a part of GTE's Telephone Operations Group. More than 2,600 employees are located in the Tampa area, while approximately 5,000 are employed at GTEDS' four regional processing centers in the United States. **GTEDS was incorporated on Oct. 25, 1967** to provide information management and systems development services to customers nationwide. Since that time GTE Data Services has been providing low-cost, high-quality data processing, office automation and internal telecommunications product and services to GTE telephone operations in the United States, Canada and the Dominican Republic. Don A. Hayes was appointed president of GTE Data Services in July of 1988, and recently hosted GTEDS' 25th Anniversary Celebration at the University of South Florida (USF) Special Events Center.

Write a program to accept as input a month, day, and year (less than 2000), and then display the number of days GTEDS has been operating. Examples:

```
INPUT: Enter month: 10
      Enter day: 27
      Enter year: 1967
```

```
INPUT: Enter month: 2
      Enter day: 4
      Enter year: 1993
```

```
OUTPUT: 2 DAYS
```

```
OUTPUT: 9234 DAYS
```

3.1 Write a program that will allow the user to move and position the cursor (a pound sign) on the screen by using appropriate keys (designated by the programmer). The program then allows the user to press 1, 2, 3, or 4, and displays a square made up of **GTEDS** relative to the cursor's position (as shown below) with the number centered in the square. If the square to be displayed will not fit on the screen, then display the error message **OFF THE SCREEN** on the top line of the screen; otherwise, display the appropriate square as shown below, relative to the cursor:

```

#           #
G T E D S   G T E D S   G T E D S   G T E D S
T           D   T           D   T           D   T           D
E   1   E   E   2   E   E   3   E   E   4   E
D           T   D           T   D           T   D           T
S D E T G   S D E T G   S D E T G   S D E T G
#           #

```

Examples:

INPUT: (**Move cursor** to top right corner and press **2**)

```

OUTPUT:           #
G T E D S
T           D
E   2   E
D           T
S D E T G

```

INPUT: (**Move cursor** to bottom right corner and press **3**)

```

OUTPUT: G T E D S
T           D
E   3   E
D           T
S D E T G
#

```

INPUT: (**Move cursor** to top left corner and press **4**)

OUTPUT: **OFF THE SCREEN**
 (Note: message must appear on the top line of screen)

INPUT: (**Move cursor** to bottom left corner and press **1**)

OUTPUT: **OFF THE SCREEN**
 (Note: message must appear on the top line of screen)

3.2 Write a program to find the value of X that satisfies an input equation with one arithmetic symbol. All numbers in the equation will be positive integers less than 1000, and the arithmetic symbols that will be used are +, -, *, and /. The following examples illustrate the procedure for entering equations such as $X+5=9$, $7-6=X$, $150=X*5$, and $6=24/X$. Examples:

```
INPUT: Enter value: X
      Enter symbol: +
      Enter value: 5
      Enter symbol: =
      Enter value: 9
```

OUTPUT: **X = 4**

```
INPUT: Enter value: 7
      Enter symbol: -
      Enter value: 6
      Enter symbol: =
      Enter value: X
```

OUTPUT: **X = 1**

```
INPUT: Enter value: 150
      Enter symbol: =
      Enter value: X
      Enter symbol: *
      Enter value: 5
```

OUTPUT: **X = 30**

```
INPUT: Enter value: 6
      Enter symbol: =
      Enter value: 24
      Enter symbol: /
      Enter value: X
```

OUTPUT: **X = 4**

3.3 Write a program to accept as input a string of several unique digits (1 through 7), and display all possible combinations of the digits which sum to an input number (**given in base 8**). Each set of single digit addends are to be separated by a plus sign and must appear in the same order as given in the input string. Each line of addends may be in any order. Examples:

```
INPUT: Enter digits: 6527
      Enter sum: 15
```

```
OUTPUT: 6+5+2 = 15      or      OUTPUT: 6+7 = 15
        6+7 = 15          6+5+2 = 15
```

```
INPUT: Enter digits: 654321
      Enter sum: 20
```

```
OUTPUT: 6+5+3+2 = 20
        6+5+4+1 = 20
        6+4+3+2+1 = 20
```

(Note: The three lines may appear in a different order.)

3.4 Write a program to decompose a large positive integer into its prime factors. The input number will be less than 80 digits with all its prime factors less than 100. The prime factors must be displayed in ascending order, each followed by the power symbol (^) and the number of times the prime divides the number. If a prime divides the number only once, then do not display the power. Examples:

INPUT: Enter number: **10633823966279326983230456482242756608**

OUTPUT: **2^123**

INPUT: Enter number: **7351545822408260372903328000000000**

OUTPUT: **2^15 * 3^4 * 5^10 * 13^2 * 37 * 89^6 * 97^3**

3.5 Write a program to find words, associated with computers, within the following 12 X 11 array of letters:

		1	2	3	4	5	6	7	8	9	0	1
1	D	A	T	A	A	D	F	B	A	A	M	
2	J	A	R	B	J	C	E	D	F	O	I	
3	R	E	A	E	E	X	E	V	D	B	C	
4	J	E	S	U	S	D	E	E	R	N	R	
5	F	A	B	U	U	N	M	I	E	M	O	
6	L	L	M	N	S	O	I	P	T	K	C	
7	P	O	Q	R	S	I	T	R	U	O	H	
8	A	B	U	V	K	W	S	X	P	P	I	
9	S	O	Y	Z	C	P	U	L	M	L	P	
10	C	C	I	S	A	B	C	D	O	A	M	
11	A	E	F	G	R	H	I	J	C	R	M	
12	L	K	L	E	T	T	E	K	S	I	D	

Words may appear vertically or horizontally, forward or backward. Display the coordinates of the first letter and the last letter of a word given as input. Examples:

INPUT: Enter word: **COMPUTER**
 OUTPUT: **FIRST LETTER: (11, 9)**
LAST LETTER: (4, 9)

INPUT: Enter word: **CHIP**
 OUTPUT: **FIRST LETTER: (6, 11)**
LAST LETTER: (9, 11)

INPUT: Enter word: **DISKETTE**
 OUTPUT: **FIRST LETTER: (12, 11)**
LAST LETTER: (12, 4)

3.6 Write a program to display all integer values of X that satisfy two equations joined by either the logical operator **AND** or **OR**, where each equation is of the form: $X > \#$, or $X < \#$ (where $\#$ is a single digit whole number).

- If no integers solve the equations, display: **NO SOLUTION**
- If all integers solve the equations display: **ALL INTEGERS**
- If there is a finite solution:
 - If there are six or less integers that solve the equations then list all the integers in ascending order separated by commas; otherwise list the first three integers separated by commas and then three periods and then the last three integers.
- If there are an infinite number of positive integers that solve the equations, then display the smallest three integers separated by commas, and then three periods.
- If there are infinite number of negative integers that solve the equations, then display three periods followed by the three largest numbers that satisfy the equations, separated by commas.
- If there are both an infinite number of positive solutions and negative solutions, but there is a gap between them, then display three spaces between these two infinite solutions as directed above.

Examples:

```
INPUT: Enter equation 1: X>3
      Enter logical op: AND
      Enter equation 2: X<9
OUTPUT: 4,5,6,7,8
```

```
INPUT: Enter equation 1: X>3
      Enter logical op: OR
      Enter equation 2: X<0
OUTPUT: ...-3,-2,-1 4,5,6...
```

```
INPUT: Enter equation 1: X<1
      Enter logical op: OR
      Enter equation 2: X>0
OUTPUT: ALL INTEGERS
```

```
INPUT: Enter equation 1: X<2
      Enter logical op: AND
      Enter equation 2: X<5
OUTPUT: ...-1,0,1
```

```
INPUT: Enter equation 1: X>1
      Enter logical op: AND
      Enter equation 2: X<9
OUTPUT: 2,3,4...6,7,8
```

3.7 Write a program to accept as input two 3 x 3 matrices (with each element no larger than 2 digits in base 16) and prints the SUM and PRODUCT (first times the second). Each element in the two results are to be displayed in base 16 and are to be right justified within 6 columns. All the elements of first matrix are input before entering the elements of the second matrix. Each element at (ROW, COL) in the PRODUCT is obtained by summing the three products of paired elements from row number ROW in matrix 1 with the elements from column number COL in matrix 2. Example:

```
INPUT: Enter Mat1 (1,1): 1      Enter Mat2 (1,1): 1
       Enter Mat1 (1,2): 2      Enter Mat2 (1,2): E
       Enter Mat1 (1,3): 3      Enter Mat2 (1,3): F2
       Enter Mat1 (2,1): A1     Enter Mat2 (2,1): 2
       Enter Mat1 (2,2): B2     Enter Mat2 (2,2): 99
       Enter Mat1 (2,3): C3     Enter Mat2 (2,3): 8D
       Enter Mat1 (3,1): D4     Enter Mat2 (3,1): 3
       Enter Mat1 (3,2): E5     Enter Mat2 (3,2): FF
       Enter Mat1 (3,3): F6     Enter Mat2 (3,3): 9E
```

```
OUTPUT: SUM =      2      10      F5
           A3      14B     150
           D7      1E4     194
```

```
PRODUCT =      E      43D     3E6
             44E 1356D 17296
             580 1897F 1DE5D
```

Note: Element 17296, located at (2,3) in PRODUCT, is derived by summing the three products: A1 * F2, B2 * 8D, and C3 * 9E.

3.8 Write a program to find all sets of three 3-digit primes composed of the digits 1 through 9 such that their sum consists of four distinct digits in order of magnitude. Output must be of the following format:

```
### + ### + ### = ####
```

where ### represents the primes displayed in increasing order, and #### represents their sum. The seven sets of primes are to be displayed in order of magnitude by the first prime and then the second prime (if two sets have the same first prime). Two of the seven solutions are displayed below. Example:

```
149 + 257 + 863 = 1269
### + ### + ### = ####
### + ### + ### = ####
241 + 367 + 859 = 1467
### + ### + ### = ####
### + ### + ### = ####
### + ### + ### = ####
```

3.9 A binary tree is a tree where each node has at most 2 children. A binary search tree has the value of each node greater than (or equal to) the value of its left child and less than the value of its right child.

Write a program to produce a binary search tree of letters from data input. Input will be one or more words separated by a space, but only the letters will be used in building the tree. No more than 9 row levels of letters will be produced from the data input. Examples:

INPUT: Enter word(s): **INDUSTRY**

```

OUTPUT:
                I
      D-----+-----N
                    +-----U
                      S----+----Y
                        R--+-T
  
```

Note: In the above example, the first letter is centered. The second letter, N, is greater than I, so it is placed 16 positions to the right of I. The third letter, D, is less than I, so it is placed 16 positions to the left of I. The fourth letter, U, is greater than I and greater than N, so it is placed 8 positions to the right of N. The fifth letter, S, is greater than I and N, but less than U, so it is placed 4 positions to the left of U. The sixth letter, T, is greater than I and N, less than U, and greater than S, so it is placed 2 positions to the right of S. The seventh letter, R, is greater than I and N, but less than U and S, so it is placed 2 positions to the left of S. The last letter, Y, is greater than I, N, and U, so it is placed 4 positions to the right of U.

INPUT: Enter word(s): **SENIOR DIVISION**

```

OUTPUT:
                S
      E-----+-----V
    D-----+-----N
          I---+----O
        I--+-N   O--+-R
        I+       +S
        I+
  
```

Note: In the above example, the fourth occurrence of the letter I is placed 1 position to the left of the previous letter I. Every letter placed beyond the 5th level of nodes is placed either 1 position to the right or to the left of its parent.

3.10 Woolley's Paradox, published in the May 6, 1984 Miami Herald Neighbors section, "proves" that $2 = 4$ by using parallel equations and the concept of infinity. A paradox is a self-contradictory statement that at first seems to be true. The fallacy of the statement presented in the newspaper can be seen in solving the following computer program.

Write a program to determine the range of values for which $F(X)$ CONVERGES, as X approaches infinity for varying positive values of K . $F(X)$ is defined below as a recursive function:

$$F(X) = \begin{cases} \text{----} \\ K & \text{if } X = 1 \\ K^{F(X-1)} & \text{if } X > 1 \\ \text{----} \end{cases}$$

For $K = 0.0$,
as X approaches infinity, $F(X)$ will equal 0.0 when X is odd and will equal 1.0 when X is even; therefore, $F(X)$ does not converge for $K = 0.0$.

For $K = 0.25$,
as X approaches infinity, $F(X)$ will converge on 0.5.

For $K = 1.0$,
as X approaches infinity, $F(X)$ will equal 1.0.

For $K = 2.0$,
 $F(1) = 2.0$,
 $F(2) = 2.0^{F(1)} = 4.0$,
 $F(3) = 2.0^{F(2)} = 16.0$,
 $F(4) = 2.0^{F(3)} = 65536.0$, and
 $F(X)$ will diverge as X approaches infinity.

For practical purposes, the functional value at $X = 5000$ can be used for "the functional value as X approaches infinity," although it may not always be necessary to perform 5000 iterations to obtain the desired result.

Output must be of the form:

MINIMUM VALUE: F(X) = #.### OCCURS WHEN K = #.###
MAXIMUM VALUE: F(X) = #.# OCCURS WHEN K = #.#####

Note: #.### represents the actual minimum value $F(X)$ takes on for some value of K . On the first line, both $F(X)$ and K need to be rounded to the nearest 0.001.

Note: #.# represents the actual maximum value $F(X)$ takes on for some value of K . $F(X)$ must be rounded to the nearest 0.1. The value of K causing a maximum value for $F(X)$ is represented by #.#####, and must be rounded to the nearest 0.00001.