
FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '91

1.1 Write a program to clear the screen and display the following:

```

COMPUTER CONTEST 1991
O           9
M           9
P           1
U
T           T
E           S
R           E
            T
C           N
O           O
N           C
T
E           R
S           E
T           T
            U
1           P
9           M
9           O
1991 TSETNOC RETUPMOC

```

1.2 Write a program to display two random integers and their sum. Output must be displayed in the form: $X + Y = ZZ$, where X and Y are random integers between -9 and 9 inclusive. At least one space must appear between symbols and numbers. Output examples:

$3 + -9 = -6$ or $-5 + -2 = -7$ or $7 + 8 = 15$

1.3 Since 1980, the only high school team to complete all 30 programming problems at the Florida High School Computer Contest was Coral Springs High School in 1984, finishing with a time of 2 hours and 21 minutes. As a result of finishing 10 one-point programs, 10 two-point programs, and 10 three-point programs, they had a maximum team score of 60 points: $(10 \times 1 + 10 \times 2 + 10 \times 3 = 60)$. Write a program to determine the team score for a school, given the team name and the number of 1-point, 2-point, and 3-point programs completed. Example:

```

INPUT: Enter team name: CORAL SPRINGS HS
      Enter # of 1 point programs: 7
      Enter # of 2 point programs: 3
      Enter # of 3 point programs: 1

```

```

OUTPUT: CORAL SPRINGS HS SCORED 16 POINTS

```

1.4 Write a program to display the screen format for a spreadsheet. The letters A through T are to appear on the first line, each separated by 1 space. The numbers 1 through 20 are to appear on the left most part of the next 20 lines. (Note: the letter A starts in the 4th column) Example:

```
  A B C D E F G H I J K L M N O P Q R S T
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

1.5 Write a program to determine the number of teams participating in the Florida High School Computer Contest. The number of students participating will be given as input. Assume that every team has 4 students. Example:

```
INPUT: Enter number of students: 168
OUTPUT: 42 TEAMS
```

1.6 Write a program to accept a word with distinct letters, and then accept a letter in the word. The program must then print the word vertically and horizontally, intersecting at the chosen letter. Example:

```
INPUT: Enter word: COMPUTER
       Enter letter: M
```

```
OUTPUT:  C
         O
        COMPUTER
         P
         U
         T
         E
         R
```

1.7 A 19 digit account key identifies specific Dealer Floor Plan information. Each set of digits represents a particular number, as follows:

Digits 1 - 3 = Organization number
Digits 4 - 6 = Branch number
Digits 7 - 10 = Dealer number
Digits 11 - 13 = Class number
Digits 14 - 19 = Unit serial number

Write a program to display the five fields followed by the actual number taken from the digits in an input 19 digit account key. Example:

INPUT: Enter account key: **0010071234001228334**

OUTPUT: **ORGANIZATION 001**
BRANCH 007
DEALER 1234
CLASS 001
UNIT 228334

1.8 JCL (Job Control Language) allows a programmer to communicate with a computer operating system. The operating system maintains control over what the computer does. A job is a unit of work that the computer is to do. A job stream consists of JCL statements. The three main JCL statements, necessary to every job, consist of the following characters: JOB, EXEC, and DD. The JOB statement must be the first statement in a job stream, and it identifies the job to the system. The EXEC statement identifies the program (or procedure) to execute. The DD (data definition) statement describes the data used by the program. A JOB STEP consists of an EXEC statement and its DD statement(s) that follow. A job may have any number of job steps. Write a program to determine the number of job steps that are in a job stream, given several lines of partial JCL code (JOB, EXEC, or DD). The last input line will be two slashes //, which marks the end of the job stream. Example:

INPUT: Enter line: **JOB**
Enter line: **EXEC**
Enter line: **DD**
Enter line: **EXEC**
Enter line: **DD**
Enter line: **DD**
Enter line: **DD**
Enter line: **DD**
Enter line: **EXEC**
Enter line: **DD**
Enter line: **//**

OUTPUT: **3 JOB STEPS**

1.9 Write a program to accept a one-line sentence and to replace all occurrences of the word MAN with the word PERSON and to replace all occurrences of the word MEN with PERSONS. The words MAN and MEN may be imbedded in other words. The output will be less than 80 characters in length. Examples:

INPUT: Enter sentence:
MEN AND WOMEN HAVE ENGAGED IN MENTAL MANIPULATIONS.

OUTPUT:
PERSONS AND WOPERSONS HAVE ENGAGED IN PERSONSTAL PERSONIPULATIONS.

1.10 Write a program to determine the winner of two teams participating in the FLORIDA HIGH SCHOOL COMPUTER CONTEST. The team with the most amount of points wins. In the event of a tie of points, the team with the lowest overall time is declared the winner. The overall time consists of the sum of the completion time of the last program finished plus 5 minutes for every penalty point accumulated. Input will be a team name, then its points, time (in the form HMM where H is hours and MM is minutes), and number of penalties. Examples:

INPUT: Enter team name: **TARAVELLA**
Enter points, time, penalties: **32, 155, 6**
Enter team name: **CORAL SPRINGS**
Enter points, time, penalties: **32, 212, 1**

OUTPUT: **CORAL SPRINGS WINS**

INPUT: Enter team name: **TARAVELLA**
Enter points, time, penalties: **33, 155, 6**
Enter team name: **CORAL SPRINGS**
Enter points, time, penalties: **32, 212, 1**

OUTPUT: **TARAVELLA WINS**

2.1 Write a program to produce a pyramid of N numbers where N is input as 6, 10, 15, 21, 28, 36, or 45. The format is shown below. Each single digit number has a leading zero, and each succeeding row has one more number than the previous row. Each number is separated by two spaces. Example:

INPUT: Enter N: 21

```

OUTPUT:          01
                02 03
                04 05 06
                07 08 09 10
                11 12 13 14 15
                16 17 18 19 20 21

```

2.2 Write a program to accept 5 decimal numbers and to display them lined up vertically with the decimal point in the same column. No more than 4 digits will be input on the right or the left of the decimal point. Display 9 dashes under the last number (4 dashes on the right side of the decimal column and 4 dashes on the left side). Display the sum of all 5 numbers under the dashes with its decimal point lined up with the other decimal points. Example:

```

INPUT: Enter #: 1.1234      OUTPUT:  1.1234
        Enter #: 3.456      OUTPUT:  3.456
        Enter #: 123.45     OUTPUT: 123.45
        Enter #: 23.5       OUTPUT:  23.5
        Enter #: 678.9166   OUTPUT: 678.9166
                               -----
                               830.4460

```

2.3 Write a program to convert a partial BASIC program statement into a proper COBOL statement. The COBOL statement will be identical to the BASIC statement except that the relational symbols will be replaced by words:

```

>          IS GREATER THAN
<          IS LESS THAN
=          IS EQUAL TO
<= or =<  IS NOT GREATER THAN
>= or =>  IS NOT LESS THAN
<> or ><  IS NOT EQUAL TO

```

Example:

INPUT: Enter statement: **IF A < B OR C >= D**

OUTPUT: **IF A IS LESS THAN B OR C IS NOT LESS THAN D**

2.4 Write a program to rank N teams in a league. Input will be a set of team names each followed by the number of wins and losses on the next line. Output will be the teams in the order of number of wins, with its rank in the league appearing to the left of the name. Each differing place is separated by 1 blank line. If more than 1 team is tied, the teams are displayed in alphabetical order with no blank lines separating them. The number of wins is displayed 16 columns after the column with the team rank. Example:

```
INPUT: Enter N: 6
      Enter team: STRIKES
      Enter wins, losses: 29, 43
      Enter team: PAR FOUR
      Enter wins, losses: 35, 37
      Enter team: WINNERS
      Enter wins, losses: 35, 37
      Enter team: RUNNERS
      Enter wins, losses: 35, 37
      Enter team: FORCE
      Enter wins, losses: 29, 43
      Enter team: HIGH ROLLERS
      Enter wins, losses: 42, 30
```

```
OUTPUT: 1 HIGH ROLLERS 42 , 30

        2 PAR FOUR      35 , 37
        2 RUNNERS      35 , 37
        2 WINNERS      35 , 37

        5 FORCE         29 , 43
        5 STRIKES      29 , 43
```

2.5 Write a program to ALWAYS guess a user's secret number within 7 guesses, where the number is between 1 and 127 inclusive. The user will respond either H--Higher, L--Lower, or R--Right on. Have the program display guesses of this form: GUESS X: YYY, where X is the guess # (1 - 7), and YYY is the number guessed. Example:

```
OUTPUT: GUESS 1: 64
INPUT: Enter H, L, or R: L
OUTPUT: GUESS 2: 32
INPUT: Enter H, L, or R: H
OUTPUT: GUESS 3: 48
INPUT: Enter H, L, or R: L
OUTPUT: GUESS 4: 40
INPUT: Enter H, L, or R: H
OUTPUT: GUESS 5: 44
INPUT: Enter H, L, or R: L
OUTPUT: GUESS 6: 42
INPUT: Enter H, L, or R: H
OUTPUT: GUESS 5: 43
INPUT: Enter H, L, or R: R
OUTPUT: (program terminates)
```

2.6 Write a program to display text in pyramid form. The first word is displayed centered on the top, with respect to column 20 of your screen. Each succeeding line contains the least amount of words that causes it to exceed the previous line by at least 2 characters. Each line is then centered with respect to column 20. If a line contains an even amount of characters, then center the line in such a way that one extra character appears on the left half of column 20 as opposed to the right half. The last remaining line is to be centered with respect to column 20 regardless of its length. All words are separated by 1 space. No more than 127 characters will be entered. Example:

INPUT: Enter text: **WRITE A PROGRAM TO DISPLAY TEXT IN PYRAMID FORM. THE FIRST WORD IS DISPLAYED CENTERED ON THE TOP.**

OUTPUT: **WRITE
A PROGRAM
TO DISPLAY TEXT
IN PYRAMID FORM. THE
FIRST WORD IS DISPLAYED
CENTERED ON THE TOP.**

2.7 Write a program to display a rectangle of asterisks centered on the screen. The length and width are input. Example:

INPUT: Enter length, width: **7, 5**

OUTPUT: (the following rectangle is centered on the screen)

```
*****  
*      *  
*      *  
*      *  
*****
```

2.8 Write a program to display a bar graph for the lengths of the computer contests from 1980-1991. The title of the graph is input and then displayed indented 4 characters on the first line. Twelve numbers (less than 32768) are input corresponding to the length of contests 1980, 1981, ... 1991. Output will be a bar graph with a maximum of 20 vertical asterisks appearing for the year with the largest number. Each vertical asterisk represents the value of the maximum length divided by 20. This increment value for each asterisk is displayed on the first line after the title. Three spaces separate the title and the phrase "ASTERISK = ####.##". The amount of asterisks for the other years is determined by dividing the length by the increment value and truncating the result. Example:

```
INPUT: Enter title: BASIC PROGRAMS
Enter # for 1980: 5475
Enter # for 1981: 5640
Enter # for 1982: 13848
Enter # for 1983: 10346
Enter # for 1984: 10932
Enter # for 1985: 18478
Enter # for 1986: 20093
Enter # for 1987: 19034
Enter # for 1988: 19140
Enter # for 1989: 20840
Enter # for 1990: 19459
Enter # for 1991: 17014
```

```
OUTPUT: BASIC PROGRAMS    ASTERISK = 1042.00
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
-----
80 81 82 83 84 85 86 87 88 89 90 91
```


2.9 Write a program to produce a file maintenance list for a store. Each day customers come into the store to either purchase, exchange, or return an item. The store assigns to each customer a unique customer ID, consisting of 2 letters followed by 2 digits. If the customer purchases an item, then the customer ID and the item name (A,B,...Y, or Z) is entered in the file. The file is duplicated each night. Customers may exchange or return an item on the next day. If a customer exchanges an item, the item name is changed in the file corresponding to the customer. If a customer returns an item, his record entry is deleted from the file. Both files are sorted alpha-numerically according to the ID. Given the contents of yesterday's file and the contents of today's file, display the records that were added, changed, and deleted, under the following headings: ADDED, CHANGED, DELETED. Display the records in order of ID within each section. Each record is to be displayed with a space between the ID and the item. If the record was changed, display the old contents followed by a space and the new contents. Each section is to be separated by a blank line. Display the total number of each function at the end. Example:

```
INPUT: Enter # of entries in yesterday's file: 4
       Enter ID: AB12
       Enter item: C
       Enter ID: CH39
       Enter item: R
       Enter ID: CH40
       Enter item: D
       Enter ID: CR11
       Enter item: A
```

```
INPUT: Enter # of entries in today's file: 3
       Enter ID: AB12
       Enter item: C
       Enter ID: CH40
       Enter item: T
       Enter ID: CR13
       Enter item: A
```

```
OUTPUT: ADDED
        CR13 A

        CHANGED
        CH40 D T

        DELETED
        CH39 R
        CR11 A

TOTAL ADDED = 1
TOTAL CHANGED = 1
TOTAL DELETED = 2
```

2.10 Write a program to display the contents of one of Doug Woolley's computer diskettes for the Florida High School Computer Contest, given the year of the contest. The contests from 1983-1991 have had 10 1-point problems, 10 2-point problems, and 10 3-point problems. In 1980, there were 10 1's, 10 2's, and 12 3's. In 1981, there were 5 1's, 5 2's, and 5 3's. In 1982, there were 10 1's, 12 2's, and 8 3's. Each year has its own disk. For each year, there is one BASIC program for each problem and one Pascal program for each problem. The names are of the form:

XXX#TYY.ZZZ

where XXX is ONE, TWO or THR; # is a problem number (1 through 5, 8, 10, or 12 depending on the year); T is T for test; YY is year of contest (80 through 91); ZZZ is BAS or PAS for BASIC or Pascal.

In addition, each diskette has 12 files of the form:

FHSYY-#.ZZZ

where FHS signifies Florida High School, YY is the year (80 through 91); # is the problem number (1, 2, or 3); ZZZ is the extension (PRB, JDG, PG1, or PG2), representing a compilation of problems, judging criteria, BASIC programs, and Pascal programs respectively. Given a year as input, display 20 files at a time, waiting for a key press before displaying the next 20. The files are to be grouped primarily by their extension in the following order: PRB, JDG, PG1, PG2, BAS, PAS. The files are then grouped secondarily within some of these sub-groups by prefix order: ONE, TWO, THR. Every sub-group is arranged in order of problem numbers. Example:

INPUT: Enter year: **1991**

OUTPUT: **FHS91-1.PRB**
FHS91-2.PRB
FHS91-3.PRB
FHS91-1.JDG
FHS91-2.JDG
FHS91-3.JDG
FHS91-1.PG1
FHS91-2.PG1
FHS91-3.PG1
FHS91-1.PG2
FHS91-2.PG2
FHS91-3.PG2
ONE1T91.BAS
ONE2T91.BAS
ONE3T91.BAS
ONE4T91.BAS
ONE5T91.BAS
ONE6T91.BAS
ONE7T91.BAS
ONE8T91.BAS

INPUT: (press any key)

OUTPUT: (continued on next page)

```
OUTPUT: ONE9T91.BAS
        ONE10T91.BAS
        TWO1T91.BAS
        TWO2T91.BAS
        TWO3T91.BAS
        TWO4T91.BAS
        TWO5T91.BAS
        TWO6T91.BAS
        TWO7T91.BAS
        TWO8T91.BAS
        TWO9T91.BAS
        TWO10T91.BAS
        THR1T91.BAS
        THR2T91.BAS
        THR3T91.BAS
        THR4T91.BAS
        THR5T91.BAS
        THR6T91.BAS
        THR7T91.BAS
        THR8T91.BAS
INPUT: (press any key)
OUTPUT: THR9T91.BAS
        THR10T91.BAS
        ONE1T91.PAS
        ONE2T91.PAS
        ONE3T91.PAS
        ONE4T91.PAS
        ONE5T91.PAS
        ONE6T91.PAS
        ONE7T91.PAS
        ONE8T91.PAS
        ONE9T91.PAS
        ONE10T91.PAS
        TWO1T91.PAS
        TWO2T91.PAS
        TWO3T91.PAS
        TWO4T91.PAS
        TWO5T91.PAS
        TWO6T91.PAS
        TWO7T91.PAS
        TWO8T91.PAS
INPUT: (press any key)
OUTPUT: TWO9T91.PAS
        TWO10T91.PAS
        THR1T91.PAS
        THR2T91.PAS
        THR3T91.PAS
        THR4T91.PAS
        THR5T91.PAS
        THR6T91.PAS
        THR7T91.PAS
        THR8T91.PAS
        THR9T91.PAS
        THR10T91.PAS
```

3.1 Write a program to simulate a baseball game of 9 innings. The standard baseball rules apply, but the bottom of the 9th inning is always played. Pitchers randomly throw strikes 40% of the time and the batters never swing at the ball. If 4 balls are thrown before 3 strikes are thrown, the batter walks to first base. When 4 batters from one team walk in one inning, 1 run is earned. Each batter that walks thereafter in the same inning earns a run for the team. 3 strikes make 1 out, and after 3 outs the next team bats. Because the program is random, executions will differ slightly. Examples:

	1	2	3	4	5	6	7	8	9	SCORE		
TEAM A	!	0	0	0	3	0	3	0	0	0	!	6
TEAM B	!	1	0	0	2	0	0	0	2	0	!	5

TOTAL # OF STRIKES: 235
 TOTAL # OF BALLS: 343
 TOTAL # OF WALKS: 77
 TOTAL # OF STRIKE OUTS: 54

	1	2	3	4	5	6	7	8	9	SCORE		
TEAM A	!	0	1	0	0	1	1	0	0	0	!	3
TEAM B	!	0	0	2	0	0	2	2	0	3	!	9

TOTAL # OF STRIKES: 251
 TOTAL # OF BALLS: 385
 TOTAL # OF WALKS: 88
 TOTAL # OF STRIKE OUTS: 54

3.2 Write a program to display the units digit of the following expression:

$$A^X + B^Y + C^Z$$

where A, B, C are input as numbers between 1 and 15 inclusive, and X, Y, Z are input as numbers between 100 and 999 inclusive. The symbol ^ means "raised to the power of". Examples:

INPUT: Enter A, X: 3, 997
 Enter B, Y: 7, 998
 Enter C, Z: 13, 999

INPUT: Enter A, X: 5, 103
 Enter B, Y: 4, 102
 Enter C, Z: 3, 101

OUTPUT: 9

OUTPUT: 4

3.3 Write a program to display all the digits of the result of X raised to the Y power, where X and Y are positive integers less than 100. The result will not exceed 200 digits, but it may wrap around the screen. Examples:

INPUT: Enter X, Y: **2, 99**
 OUTPUT: **633825300114114700748351602688**

INPUT: Enter X, Y: **45, 67**
 OUTPUT: **582422873843435732243403365900823690091753194215381
 550018222590908115610802697759140755806583911180496
 2158203125**

3.4 Write a program to assign user LOGON ID's to several people who would like access to their company's mainframe computer. Input will be several lines of full names (entered one line at a time) and followed by a line with the word END to indicate the end of input. Each input line will be of the format:

(first name)(1 space)(middle name or initial or nothing)
 (1 space)(last name)

If no middle name or initial is entered then only 1 space separates the first and last names. Output will consist of each full name followed by his/her LOGON ID. All IDs are to be lined up vertically, starting 19 columns after the first character in the name. The LOGON ID is generally of the form:

SD(3 initials of first, middle, last name)1

There are two exceptions. If no middle initial or name is entered, then the letter X is used as the middle initial. If more than one person has the same initials then the common names are sorted alphabetically by last name then first name. The first sorted name is assigned a middle initial of 1; the next name alphabetically is assigned a middle initial of 2; and so on. Example:

INPUT: Enter name: **DOUG E WOOLLEY**
 Enter name: **DAVE E WEAVER**
 Enter name: **ALICE DOCK**
 Enter name: **ANNE VINCENT**
 Enter name: **AL DATSON**
 Enter name: **DON ENGLAND CHANG**
 Enter name: **END**

OUTPUT: **DOUG E WOOLLEY** **SDD2W1**
DAVE E WEAVER **SDD1W1**
ALICE DOCK **SDA2D1**
ANNE VINCENT **SDAXV1**
AL DATSON **SDA1D1**
DON ENGLAND CHANG **SDDEC1**

3.5 Write a program to display the numbers 0 through 9, one at a time in enlarged block format (15 characters high by 11 characters wide). Each number must be displayed for approximately 1 second before clearing the screen and displaying the next number. The following are examples of the numbers:

```

*****      *      *****      *****
*           *      *           *           *
*           *      *           *           *
*           *      *           *           *
*           *      *           *           *
*           *      *           *           *
*           *      *           *           *
*           *      *           *           *
*           *      *           *           *
*           *      *           *           *
*           *      *           *           *
*           *      *           *           *
*           *      *           *           *
*           *      *           *           *
*****      *      *****      *****

```

```

*           *      *****      *****      *****
*           *      *           *           *           *
*           *      *           *           *           *
*           *      *           *           *           *
*           *      *           *           *           *
*           *      *           *           *           *
*           *      *           *           *           *
*****      *           *           *           *           *
*           *           *           *           *           *
*           *           *           *           *           *
*           *           *           *           *           *
*           *           *           *           *           *
*           *           *           *           *           *
*           *           *           *           *           *
*           *           *           *           *           *
*****      *****      *****      *****

```

```

*****      *****
*           *      *           *
*           *      *           *
*           *      *           *
*           *      *           *
*           *      *           *
*           *      *           *
*****      *****
*           *           *           *
*           *           *           *
*           *           *           *
*           *           *           *
*           *           *           *
*****      *****

```

3.6 Write a program to evaluate an arithmetic expression using positive integers less than 10 and the symbols + and -. Several pairs of parenthesis may also be used. Examples:

INPUT: Enter expression: **(8+4)-(8-(6+1))**

OUTPUT: **11**

INPUT: Enter expression: **7-9+(1-(6+2))-5+9**

OUTPUT: **-5**

INPUT: Enter expression: **((1+(1+2-3+4)-2+5)-3)**

OUTPUT: **5**

3.7 Write a program to determine the 2 dates that an employee is paid in a given month in 1991. Normally an employee is paid on the 15th and the last day of the month except if this date falls on a weekend or on a holiday. In such a case, the pay day is the first working day proceeding the normally expected pay day. Employees work Monday through Friday except on holidays. The program must first accept all the holidays for that year, input as MM, DD. After entering 0,0 to terminate holiday entries, the program accepts the month number for the pay days desired. Output must be the 2 pay days in that month in the form: day of week; month name; day. The program is to continue accepting month numbers and displaying pay days until 0 is entered for the month number. January 1, 1991 was a Tuesday. Example:

INPUT: Enter holiday MM, DD: **1, 1**
Enter holiday MM, DD: **1, 21**
Enter holiday MM, DD: **2, 28**
Enter holiday MM, DD: **3, 29**
Enter holiday MM, DD: **0, 0**

INPUT: Enter month #: **1**

OUTPUT: **TUESDAY JANUARY 15**
THURSDAY JANUARY 31

INPUT: Enter month #: **3**

OUTPUT: **FRIDAY MARCH 15**
THURSDAY MARCH 28

INPUT: Enter month #: **0**

OUTPUT: (program terminates)

3.8 Write a program to display all 3 x 3 magic squares (consisting of the digits 1 - 9) that contain a given digit in a designated position.

```
6 7 2      is a 3 x 3 magic square where the sum of
1 5 9      all the elements in each row, column, and
8 3 4      diagonal equal the same number, 15.
```

In this example, the digit 7 is in row 1, column 2, designated as (1,2). Given one digit as input (not 5) and the row and column this digit must appear, display all 3 x 3 magic squares that have that digit in the designated position. If no magic squares can be formed then display "NO SOLUTION". Hint: If a solution is possible, there will be two magic squares, each of which can be formed by rotating the outer numbers of a valid magic square and by interchanging the elements on each side of the 2nd row, 2nd column, or a diagonal (depending on the row and column of the number designated). The two magic squares may appear in either order. Examples:

```
INPUT: Enter digit: 9
      Enter row, col: 1, 2
```

```
OUTPUT: 2 9 4
        7 5 3
        6 1 8

        4 9 2
        3 5 7
        8 1 6
```

```
INPUT: Enter digit: 4
      Enter row, col: 3, 3
```

```
OUTPUT: 6 7 2
        1 5 9
        8 3 4

        6 1 8
        7 5 3
        2 9 4
```

```
INPUT: Enter digit: 4
      Enter row, col: 1, 2
```

```
OUTPUT: NO SOLUTION
```


3.10 Write a program to convert a base M numeral into a base N numeral where M and N are entered as 2, 4, 8, or 16. Both numerals may have as many as 64 digits. Output must not contain leading zeroes. Examples:

```
INPUT: Enter numeral: 123456789ABCDEF0123456789ABCDEF0  
      Enter base M: 16  
      Enter base N: 4
```

```
OUTPUT:  
102031011121320212223303132330001020310111213202122233031323300
```

```
INPUT: Enter numeral:  
      12345670123456701234567012345670123456701  
      Enter base M: 8  
      Enter base N: 16
```

```
OUTPUT: 14E5DC14E5DC14E5DC14E5DC14E5DC1
```