The following section contains problems from YSU's 1991 contest. There are additional problems on YSU's anonymous ftp site as well as other sites listed in Appendix A.

# 1991 YSU-ACM High School Programming Contest

## Problem #1

**Description:** Given an integer $X$ from 1 to 40, print a inverted triangle with $X$ levels. Do this process five times.

**Input:** An integer from 1 to 40

**Output:** The inverted triangles, each separated by a blank line.

**Example**

| Input | Output |
|---|---|
| 5 | ***** |
| 2 | **** |
| 3 | *** |
| 1 | ** |
| 2 | * |
| | |
| | ** |
| | * |
| | |
| | *** |
| | ** |
| | * |
| | |
| | * |
| | |
| | ** |
| | * |

# 1991 YSU-ACM High School Programming Contest

## Problem #2

**Description:** Given any positive integer $X$, find $\log(X!)$ where $X!$ may be larger than the maximum representable number. Remember that since a computer is a finite machine, there is a maximum value for an integer.

**Note:**

$$\log(X) = \ln(X)/\ln(10)$$

$$\log(A * B) = \ln(A) + \log(B)$$

$$X! = 1 * 2 * 3 * ... * (X-1) * X$$

**Input:** Input will consist of five lines, each line containing one integer.

**Output:** The value of $X$ and the correct solution, one per line.

**Example:**

| **Input** | **Output** |
|---|---|
| 5 | $X = 5 \;\; \log(5!) = 2.079181246$ |
| 100 | $X = 100 \;\; \log(100!) = 64.78410487$ |

# 1991 YSU-ACM High School Programming Contest

## Problem #3

**Description:** You will read an array of up to 50 numbers from a file. The first integer, $N$, in the file is the number of elements in the array. The following $N$ integers are the elements of the array, starting with element A(.0.). You will then perform the following steps for four cycles:

**Step 1:** Read an integer $X$ from the file.

**Step 2:** If A(.$X$.) is 0, then print $X$ and stop. If A(.$X$.) is not 0, then repeat step 2 with A(.$X$.) as the new value of $X$.

**Input:** A text file containing a list of integers. The last four integers are the starting values of $X$.

**Output:** The index of the element equal to 0.

**Example:**

| **Input** | **Output** |
|---|---|
| 18 | 4 |
| 7 | 11 |
| 11 | 16 |
| 1 | 4 |
| 0 | |
| 3 | |
| 13 | |
| 9 | |
| 12 | |
| 4 | |
| 2 | |
| 0 | |
| 16 | |
| 17 | |
| 4 | |
| 8 | |
| 0 | |
| 10 | |
| 8 | |
| 3 | |
| 6 | |
| 8 | |
| 4 | |

# 1991 YSU-ACM High School Programming Contest

## Problem #4

**Description:** Given three five-character strings, each on a seperate line, find the values of the variables A, B, and C, such that the addition of lines one and two equals line three. You may find any correct solution to this expression.

**Note:** Each string contains at least one of the three variables.

**Input:** Input will be three lines of five characters.

**Output:** Output should be the strings, one per line, with the variables replaced by their appropriate values.

**Example:**

| **Input** | **Output** |
|---|---|
| C793A | 17936 |
| B83B5 | 28325 |
| 4ABAC | 46261 |

# 1991 YSU-ACM High School Programming Contest

## Problem #5

**Description:** Given a file of strings, read in a line and replace every other alphabetical character with it's complement, as shown below. This will continue until a line occurs that does not contain the specified pattern.

**Exception:** Characters within the specified pattern are to be left unchanged. The characters of the specified pattern, as well as those found within the pattern, are not to be counted in determining every other character.

**Pattern:** ('...') ... = every character between

**Complement:** The complement of a letter is the letter whose distance from the end of the alphabet equals the distance of the original letter from the the beginning of the alphabet.

**Note:** The IBM Mainframe Character Set contains additional characters within the alphabetic characters, 'A'..'Z'.

**Alphabet:** A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

**Complement:** Z Y X W V U T S R Q P O N M L K J I H G F E D C B A

**Input:** The text file to be alternated. The file can contain any character but letters will be in uppercase only, 'A'..'Z'.

**Output:** The processed lines, each on a separate line.

**Example:**

**Input:**
```
THIS IS THE ('FIRST') LINE IN THE SAMPLE FILE!!!
ANY CHARACTERS CAN BE IN THE FILE%^$&*.(")..
('THIS PART OF THE FILE SHOULD NOT BE CHANGED')
'(BUT THIS PART SHOULD 123)'.....('.....
NUMBERS, 1234567890, ARE OK.OK?
```

**Output:**
```
TSIH RS TSE ('FIRST') OIME IM GHV HANPOE FRLV!!!
AMY CSAIAXTVRH XAM YE IM GHV UIOE%^$&*.(")..
('THIS PART OF THE FILE SHOULD NOT BE CHANGED')
'(BFT TSIH KAIT SSOFLW 123)'.....('.....
```