Programming Assignment 5

Instructions: *Solve the following problems in C. Some example input and output cases are given for better understanding. Your program is expected to provide correct output on* ***any valid input.*** *Ensure that your input and output formats exactly match the examples. The words 'SAMPLE INPUT: ' and 'SAMPLE OUTPUT: ' in the examples are* ***NOT*** *part of the input and output.*

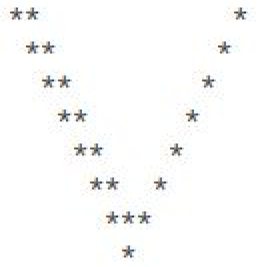
Note: *Some pattern printing problems need trailing spaces to be printed in each row while some do not. So check the problem statement carefully.*

Problem Set 1

Instructor: Om Prakash Patel
Timeslots:
- Tuesday: 4:30 PM -5:00 PM
- Thursday: 2:30 PM -3:00 PM
Venue: First floor faculty office

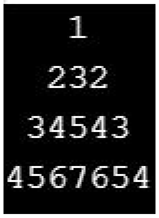
1. Write a program that will create a letter V of any size N < 20 as shown below. (Print the trailing spaces as well in each row).

   SAMPLE INPUT: 8
   SAMPLE OUTPUT:



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

2. Write a program to print the following pattern (the trailing spaces in each row should also be printed).



```
   1
  232
 34543
4567654
```

3. Often, we use shorthands such as a-f, 16023-16054 or A-Z while jotting down our notes. Write a program which expands a given shorthand. Assumption: a shorthand whether number or alphabets, (p-q) has an implicit ordering and q>=p. Input is type (0-alphabet, 1-Digit) AND shorthand. If a shorthand is given starting with a capital letter and ending with a small letter, ignore the non-alphabets which come in between them (check the fifth test case below).

   Input format: <0 or 1><space><shorthand>

   SAMPLE INPUT: 0 a-d
   SAMPLE OUTPUT: a b c d

   SAMPLE INPUT: 1 11-17
   SAMPLE OUTPUT: 11 12 13 14 15 16 17

   SAMPLE INPUT: 1 5-0
   SAMPLE OUTPUT: ERROR

   SAMPLE INPUT: 1 0-a
   SAMPLE OUTPUT: ERROR

   SAMPLE INPUT: 0 C-a
   SAMPLE OUTPUT: C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c

4. The value of π can be calculated with the series:

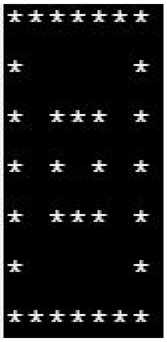
$$\pi = 4 + \tfrac{4}{3} - \tfrac{4}{5} + \tfrac{4}{7} - \tfrac{4}{9} + \ldots$$

   Write a program to compute the NUMBER OF TERMS in series required to obtain the value of PI approximated to a user specified *error percentage*. The reference value of PI is 3.141592. For instance, if the error percentage is 10, then the approximated value would fall in the range (3.141592 - 0.1*3.141592, 3.141592 + 0.1*3.141592)

   SAMPLE INPUT: 10
   SAMPLE OUTPUT: 4

   SAMPLE INPUT: 1
   SAMPLE OUTPUT: 32

5. Write a program to print the following pattern.



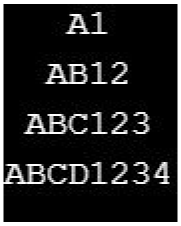
```
*******
*     *
* *** *
* * * *
* *** *
*     *
*******
```

Problem Set 2

Instructor: Sunny Rai
Timeslots:
- Monday: 4:30-5PM
- Thursday: 3:30-4PM

Venue: Second floor faculty office

1. Write a program to print the following pattern (the spaces on the right side of the last character in each row should also be printed).



```
   A1
  AB12
 ABC123
ABCD1234
```

2. There exists an interesting number "Kaprekar Constant". The number is notable for following rules:
   (a) Take any four digit number, using at least two different digits.
   (b) Arrange the digits in descending order and then in ascending order to get two four digit numbers.
   (c) Subtract the smaller number from the bigger number.
   (d) Go back to Step-2 and repeat till it reaches a fixed point (no change in digits).

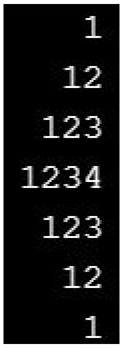
   For example: 6143
   6431-1346 = 5085
   8550-0558 = 7992
   9972-2799 = 7173
   7731-1377 = 6354
   6543-3456 = 3087
   8730-0378 = 8352
   8532-2358 = 6174
   7641-1467= 6174 // Kaprekar Constant

   Write a program to find the number of steps required to reach Kaprekar Constant for a given 4 digit number.

   SAMPLE INPUT: 4444
   SAMPLE OUTPUT: INVALID

   SAMPLE INPUT: 6143
   SAMPLE OUTPUT: 7

3. Write a program to print the following pattern.



```
   1
  12
 123
1234
 123
  12
   1
```

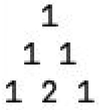
4. The 3n+1 Problem – Consider the following algorithm to generate a sequence of numbers. Start with an integer n. If n is even, divide by 2. If n is odd, multiply by 3 and add 1. Repeat the process with the new value of n, terminating when n = 1. For example, the following sequence of numbers will be generated for n=22: 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1. It is conjectured (but not yet proven) that this algorithm will terminate at n=1 for every integer n. Still, the conjecture holds for all integers up to at least 1,000,000. For an input n, the cycle-length of n is the number of numbers generated up to and including 1. In the example above, the cycle length of 22 is 16. Given an input integer n where 1<=n<=1,000,000, find its cycle length.

SAMPLE INPUT: 22
SAMPLE OUTPUT: 16

SAMPLE INPUT: 9
SAMPLE OUTPUT: 20

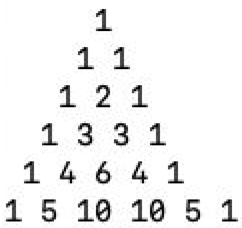
5. Write a program to print Pascal's triangle for a given number of rows. (Do not print the trailing spaces in each row except for the space after each number).

SAMPLE INPUT: 3
SAMPLE OUTPUT:



```
  1
 1 1
1 2 1
```

SAMPLE INPUT: 6
SAMPLE OUTPUT:

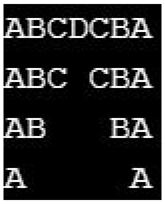


```
     1
    1 1
   1 2 1
  1 3 3 1
 1 4 6 4 1
1 5 10 10 5 1
```

Problem Set 3

Instructor: Rajesh Tavva
Timeslots:
- Monday: 4:30-5PM
- Thursday: 4:30-5PM
Venue: Library (faculty reading section)

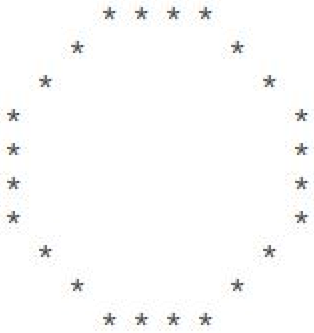
1. Write a program to print the following pattern.



```
ABCDCBA
ABC CBA
AB   BA
A     A
```

2. Write a program that will generate and print a STOP sign of size N, where N>=3. The stop sign should be an octagon with N stars on each side. (There should be a space after each '*', print the trailing spaces also in each row).

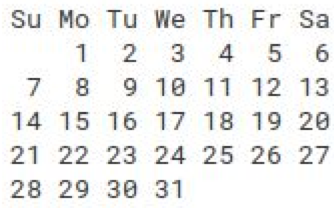
SAMPLE INPUT: 4
SAMPLE OUTPUT:



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

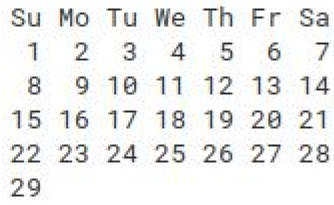
3. Write a program to print the calendar of a given month. Your input is the date of the first Sunday in the month and the total number of days in the month. Print in the following format. (Do not print the trailing spaces in the last line except for the space after the number).

SAMPLE INPUT: 7 31
SAMPLE OUTPUT:



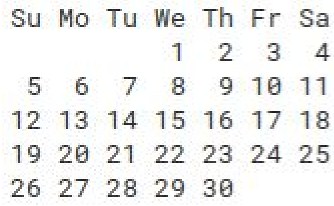
```
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

SAMPLE INPUT: 1 29
SAMPLE OUTPUT:



```
Su Mo Tu We Th Fr Sa
 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
```

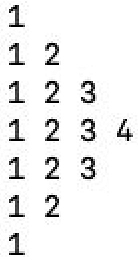
SAMPLE INPUT: 5 30
SAMPLE OUTPUT:



```
Su Mo Tu We Th Fr Sa
          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
```

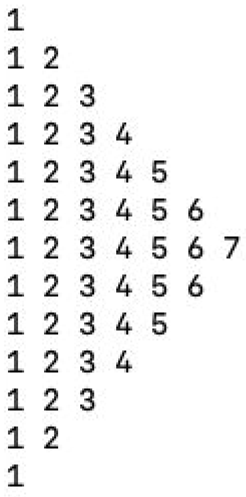
4. Write a program to print the following pattern. Input that is, the largest number in the pattern is provided by the user. (Print a space after each number, but DO NOT print the empty spaces to fill the entire rows in the matrix; the last line of the matrix should not be a newline, it should be ‘1 ’)

SAMPLE INPUT: 4
SAMPLE OUTPUT:



```
1
1 2
1 2 3
1 2 3 4
1 2 3
1 2
1
```

SAMPLE INPUT: 7
SAMPLE OUTPUT:



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

5. Factovisors - The factorial function, n! is defined as follows for all non-negative integers n: 0!=1; n!=n*(n-1)! for n>0. We say a divides b if there exists an integer k such that k * a= b. Given two non-negative integers, m and n, both less than or equal to 231, output a line stating whether or not m divides n!, in the format shown below. The first line of the input denotes the number of pairs of integers given as input.

SAMPLE INPUT:

```
3
9 6
9 5
100 10
```

SAMPLE OUTPUT:

```
9 divides 6!
9 does not divide 5!
100 divides 10!
```