ASSIGNMENT - 4

MS Connect

Wipro Technologies

Objectives

- Get comfortable with .NET.
- Start thinking more carefully.
- Solve some problems in C#

Reasonable

- Communicating with colleagues about problem problems in English (or some other spoken language).
- Discussing the assignment material with others in order to understand it better.
- Helping a colleagues identify a bug in his or her code, as by viewing, compiling, or running his or her code, even on your own computer.
- Incorporating snippets of code that you find online or elsewhere into your own code, provided that those snippets are not themselves solutions to assigned problems and that you cite the snippets' origins.
- Sending or showing code that you've written to someone, possibly a colleagues, so that he or she might help you identify and fix a bug.

Rules

- Draw the flow chart for each solutions.
- Write the Algorithm for each solution.
- Document your solutions in word file.
- Write proper comments for each line in your source code.
- Document the output of your program.
- Your program should address the problem, there should NOT be any deviations in output.

1. ASCII Weaving

Two integers:

- A non-negative integer W in the range 0 to 2^64-1, specifying the weave.
- A positive integer S in the range 1 to 255, specifying the side length.

These can be taken in whichever order suits you.

Output

An S by S ASCII representation of the requested weave (S newline separated strings of S characters with an optional trailing newline). The weave is defined by the weave number W as follows:

Convert W to binary and split into 8 bytes. The first (most significant) byte defines the top row, from left (most significant bit) to right. The next byte defines the next row, and so on for 8 rows. The weave number defines an 8 by 8 square which should be tiled over the required area starting from the top left. That is, its top left corner should correspond to the top left corner of the area to be covered.

Every 0 should be displayed as a | and every 1 should be displayed as a -

Examples

Input

```
0 8
```

Ouput

```
||||||||
||||||||
||||||||
||||||||
||||||||
||||||||
||||||||
||||||||
```

Input

```
 3703872701923249305 8
```

Output

```
||--||--
|--||--|
--||--||
-||--||-
||--||--
|--||--|
--||--||
-||--||-
```

Input

```
3732582711467756595 10
```

Output

```
||--||--||
--||--||--
--||--||--
||--||--||
||--||--||
--||--||--
--||--||--
||--||--||
||--||--||
--||--||--
```

Input

```
16141147355100479488 3
```

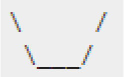
Output

```
---
|||
---
```

2. Water Bowl

You should write a program or function which receives the volume of a bowl and the volume of the water in it as input and outputs or returns an ASCII representation of a bowl with water in it with the desired volumes.

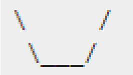
A bowl has the following structure:



```
\     /
 \___/
```

The bowl has at least one _ character. The count of \'s and /'s are also positive and they are equal due to symmetry.

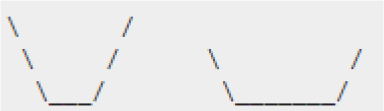
The volume of the bowl is the total number of _ and space characters between the \'s and /'s plus one for every pair of \ and /. This means the above bowl has the volume of 10:



```
\     /
 \___/
```

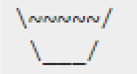
=> xxxxx x (the last one is for the \/ pair)
xxx x (the last one is for the \/ pair)

Note that two different bowl could have the same volume. E.g. both the following bowls have a volume of 18:



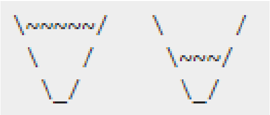
```
\       /
 \     /     \         /
  \___/       \_______/
```

We can pour some water into the bowl. The water is represented as a row of ~ characters instead of spaces inside the bowl. The bottom row has no spaces so it can not contain the ~'s. This means our example can be filled with water in only one way:



```
\~~~~~/
 \___/
```

Other bowls could be filled in multiple ways:



```
\~~~~~/     \     /
 \   /       \~~~/
  \_/         \_/
```

The volume of the water in a bowl is the volume of the bowl rows below the ~ characters. The above examples have water volumes of 4, 6 and 2 respectively.

Input

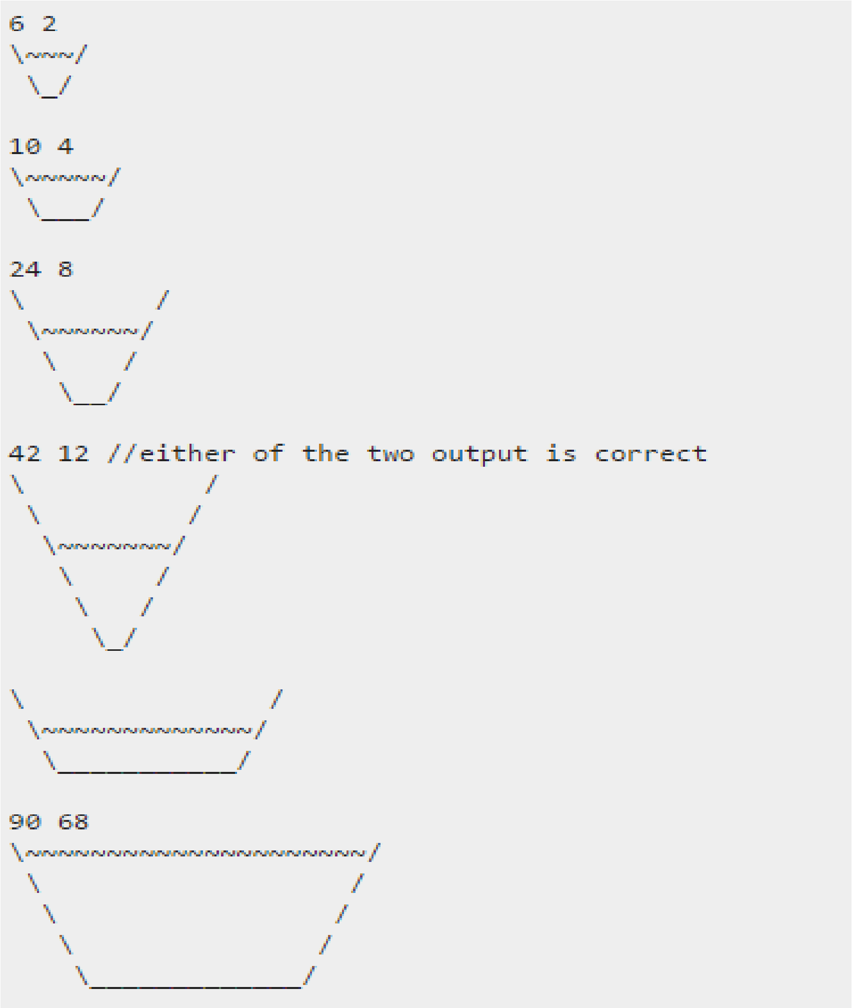
- Two positive integers, the volume of the bowl and the volume of the water.
- You can choose the order of the two numbers.
- The two integer can be inputted in any common list format (list, tuple, array, etc.) or as two separate integers.
- At least one valid bowl-water configuration is guaranteed for the input values.

Output

- The ASCII representation of a bowl with water where the bowl and water volumes match the input.
- If you choose to return the result instead of printing, it should be returned as single string (or your language's closest alternative).
- Any trailing whitespace is allowed.
- No unnecessary leading whitespace is allowed.
- If there are multiple correct configurations you are free to choose which one you output but you can only output one of them.

Examples

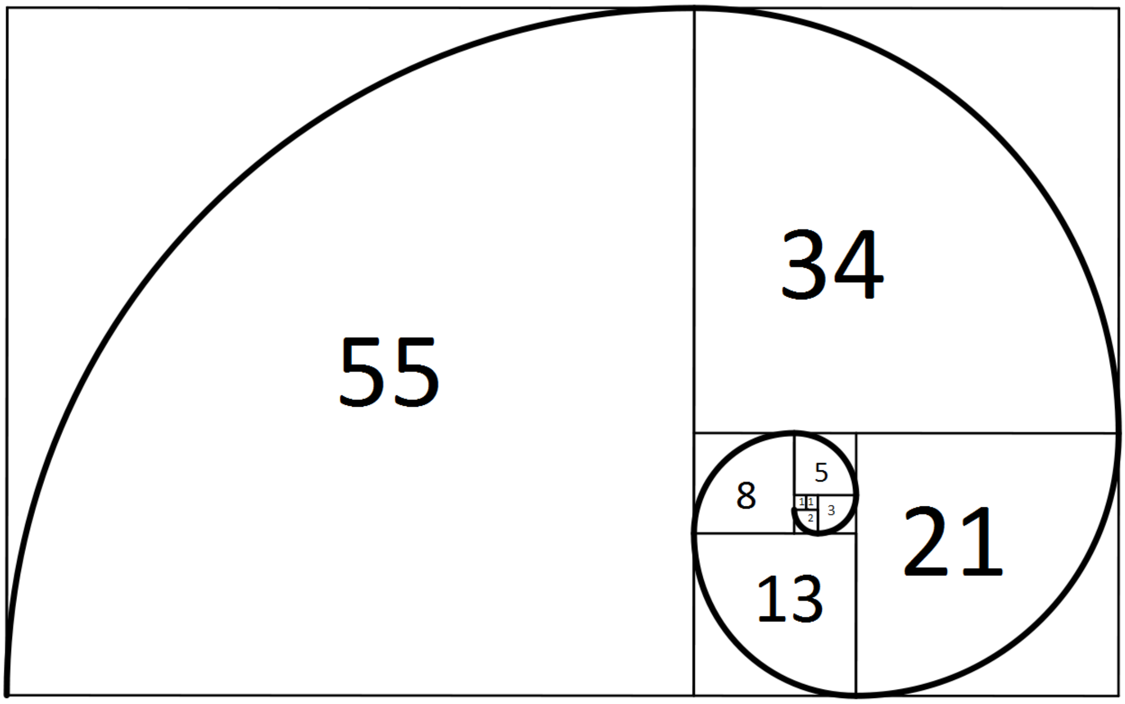
Each input integer pair is followed by its one or more possible outputs.



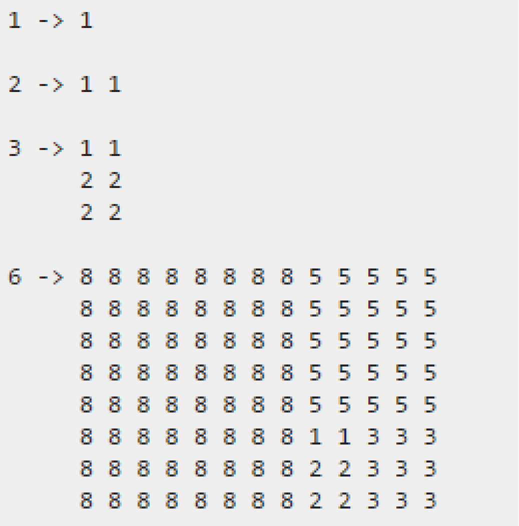
```
6 2
\~~~/
 \_/

10 4
\~~~~~/
 \___/

24 8
\        /
 \~~~~~~/
  \    /
   \__/

42 12 //either of the two output is correct
\           /
 \         /
  \~~~~~~~/
   \     /
    \   /
     \_/

\               /
 \~~~~~~~~~~~~~/
  \___________/

90 68
\~~~~~~~~~~~~~~~~~~~~~/
 \                   /
  \                 /
   \               /
    \_____________/
```

3. Fibonacci spiral with numbers

Your goal is to generate a Fibonacci spiral with numbers.



Example Input / Output



```
1 -> 1

2 -> 1 1

3 -> 1 1
     2 2
     2 2

6 -> 8 8 8 8 8 8 8 8 5 5 5 5 5
     8 8 8 8 8 8 8 8 5 5 5 5 5
     8 8 8 8 8 8 8 8 5 5 5 5 5
     8 8 8 8 8 8 8 8 5 5 5 5 5
     8 8 8 8 8 8 8 8 5 5 5 5 5
     8 8 8 8 8 8 8 8 1 1 3 3 3
     8 8 8 8 8 8 8 8 2 2 3 3 3
     8 8 8 8 8 8 8 8 2 2 3 3 3
```

Input the input can be taken through Console or function argument. It will be a single number

Output The output can be from console or a function's return value. It should be a single string.

Extra whitespace at the very end of the line is not allowed. The output can contain digits, linefeeds (newlines), and spaces.

Orientation does not matter, this means rotations and reflections. As long as it follows a valid Fibonacci spiral pattern.

Numbers with different amounts of digits (e.g. 1 and 13) should be right-aligned with each other. A space may need to be added to the very beginning of a line so everything can line up.

```
1   1                          1   1
100 100  should actually be  100 100
```

4. Checkerboard

Write a program that takes a positive integer n as an argument, in console, or as user input, and output a checkerboard with ***NxN*** squares, along with a border that is 1 character thick.

Each square should be 2x2 characters. The squares should follow the normal alternating white-black (white first, as in top-left corner) pattern of a checkerboard. White squares should be made out of space () characters, and black squares should be made out of pound (#) characters.

The border should be made out of dashes (-) with a plus (+) on the border or perpendicular point of a square.

Input

Positive integer in representing the number of squares (dimensions in squares) to draw in the checkerboard, with each square being 2x2 characters.

Example Results

n=2

```
+--+--+
|  |##|
|  |##|
+--+--+
|##|  |
|##|  |
+--+--+
```

n=3

```
+--+--+--+
|  |##|  |
|  |##|  |
+--+--+--+
|##|  |##|
|##|  |##|
+--+--+--+
|  |##|  |
|  |##|  |
+--+--+--+
```

n=4

```
+--+--+--+--+
|  |##|  |##|
|  |##|  |##|
+--+--+--+--+
|##|  |##|  |
|##|  |##|  |
+--+--+--+--+
|  |##|  |##|
|  |##|  |##|
+--+--+--+--+
|##|  |##|  |
|##|  |##|  |
+--+--+--+--+
```

... and so on.

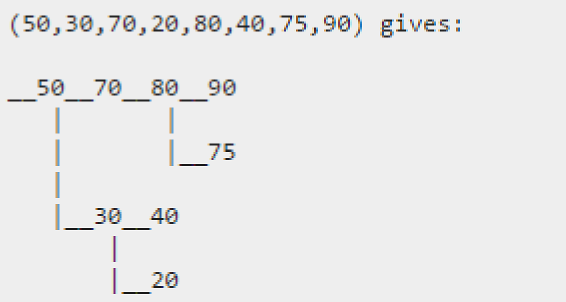
Notes

- Trailing spaces and new lines are acceptable.
- You may write either an entire program or a function.
- No leading spaces.
- Your program should display correct results for n=15.
- For less-known esoteric languages and similar, provide a link to the language.
- n=0 should produce +. (Optional, but highly recommended and encouraged.)

5. Binary search tree.

Brief

- Print an ASCII representation of a binary search tree.
- You should write your own minimum implementation logic of a BST (node, left, right, insert)



```
(50,30,70,20,80,40,75,90) gives:

__50__70__80__90
   |       |
   |       |__75
   |
   |__30__40
       |
       |__20
```

Detailed requirements

- Items are inserted into the tree in the order they are in the list
- All node values are right aligned in columns.
- Right nodes are joined using __
- Right nodes are represented along the horizontal and appear directly to the right of the parent node
- Right nodes are separated by a minimum of 2 underscores (__)
- Left nodes are joined using | and |__ on the next line
- Left nodes are represented along the vertical and appear under and to the right of the parent (ie, next column).
- Left nodes are separated by a a minimum of a blank line (discounting pipes (|) linking child to parent)
- The vertical pipes joining 2 left nodes should be in the column directly under the last character of the parent node (see samples)
- Column width is determined using something like max(col_values[]).toString().length() + 2 or find your way to do this
- Branching left always adds a new row to the tree diagram
- 2 separate (left) branches never appear on the same line (right__right__right is OK)
- The root node may optionally be prefixed with __ (Removing it just about doubled my test code!)
- Input is a set of positive and/or negative integers

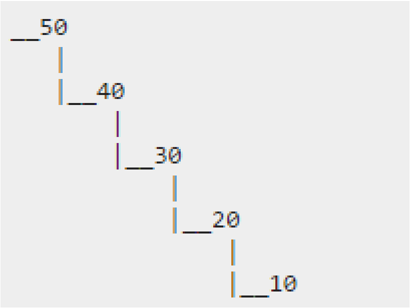
- You do not need to deal with duplicate values (drop/ignore them)
- I'm not placing many restrictions on trailing whitespaces:
- You may have trailing whitespaces after the last node on a line
- You may have trailing whitespace after a | on the empty lines
- No whitespace between nodes when when branching right

Sample Input & Output

Input

(50,40,30,20,10)

Output

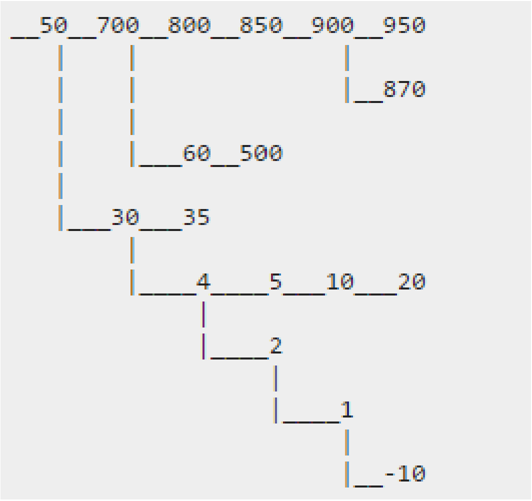


```
__50
   |
   |__40
       |
       |__30
           |
           |__20
               |
               |__10
```

Input

(50,700,30,800,60,4,5,10,20,2,500,850,900,950,870,35,1,-10)

Output



```
__50__700__800__850__900__950
   |    |              |
   |    |              |__870
   |    |
   |    |___60__500
   |
   |___30___35
        |
        |____4____5___10___20
             |
             |____2
                  |
                  |____1
                       |
                       |__-10
```

6. Dosa in Chutneys

You are a talented young chef who has just been offered the position of sous chef at the world's most prestigious Indian restaurant, Chutneys in Hyderabad. You have little experience with preparing Indian cuisine, but you're determined, so you set out to prove yourself. You decide to become the leading dosa expert. To do this, you must not only master the creation of dosas, but you must be able to scale them to be arbitrarily large.

1 cup rice
1 cup ukda chawal
1/2 cup urad dal
1/4 cup poha
1/4 tsp methi seeds
Salt to taste
Water as needed

This will make a dosa approximately 1/2 meter in length.

Challenge

Write a program or function that will tell the restaurant staff exactly what's needed to make a single dosa out of an integer multiple of the given recipe. Space is tight in the kitchen, so you want your code to be as short as possible.

Measuring spoons and cups come in the following standard sizes: 1/4, 1/3, 1/2, 2/3, 3/4, and 1. To avoid angering the kitchen staff, measurements must be reported in the largest unit in which the number can be written as mixed numbers using standard sizes only.

Measurements propagate to larger sizes per the following convention:

- 3 tsp == 1 tbsp
- 4 tbsp == 1/4 cup

So for a multiple of 12, 1/4 tsp methi seeds becomes 1 tbsp. However, for a multiple of 13, it becomes 3 1/4 tsp. Otherwise it wouldn't be represented in standard sizes.

The wait staff must be able to carry the dosa to the tables. To ensure that the dosa does not break in transit, you instruct them to carry the dosa in teams. Each person can carry at most one meter of dosa. So for a single or double recipe, only one person is needed to carry it. The wait staff is less effective if they're cut into fractional pieces, so an integer number of waiters is always required.

Input

Take a positive integer command line argument, or function argument. This number dictates the scaling factor for the recipe and can be as small as 1, but it's int.

Output

Print to console the list of ingredients scaled according to the input as well as the number of waiters required to carry the dosa. The ingredients must be listed in the order given above and in the format given below.

Examples

Input

2

Output

```
2 cups rice
2 cups ukda chawal
1 cup urad dal
1/2 cup poha
1/2 tsp methi seeds
Salt to taste
Water as needed 1 waiter
```

Note that "cup" changes to "cups" when the value is greater than 1. "tsp" does not change. "waiter," like "cup," becomes plural.

Input

5

Output

```
5 cups rice
5 cups ukda chawal
2 1/2 cups urad dal
1 1/4 cups poha
1 1/4 tsp methi seeds
Salt to taste
Water as needed
3 waiters
```

Non-integer values greater than 1 are represented as mixed numbers, i.e. an integer followed by a reduced fraction.

7. Sphinx

After a long battle, you have managed to defeat a Sphinx in a contest of riddles. Sphinx , impressed with your skill, wishes to give you a reward commensurate with your cleverness, and conjures into existence a strip of magical parchment divided into eight boxes, each containing a numeral.

"Crease the parchment," says Sphinx , "such that the boxes overlap, and those boxes will merge either through addition or multiplication. When one box remains, its value will be your reward in gold coins."

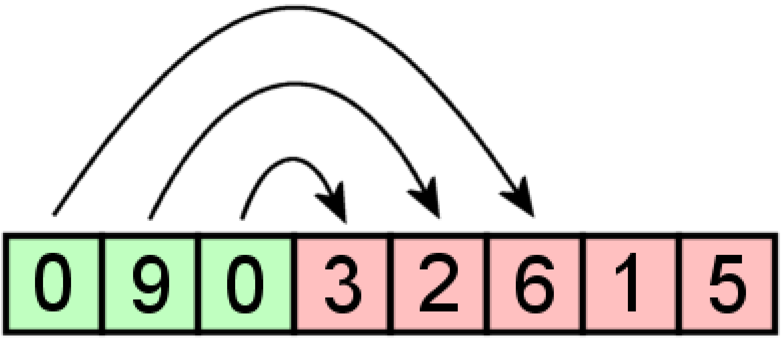
Task

You must write a program or function which takes as input a list/array/whatever of eight natural numbers, and returns/prints the maximum possible reward obtainable through a series of 'crease' operations.

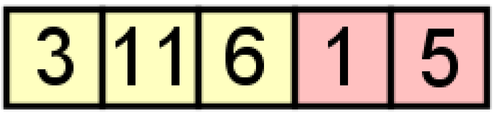
Mechanics

The 'crease' operation is performed on some number of cells, and with either + or * as the operator. The first n cells of the list are folded across and merged with their destination cells using the operator. Any cells which are not consumed in the merge operation are left unmodified.

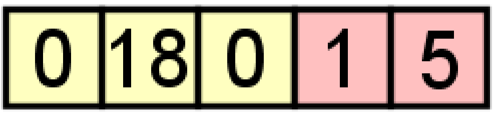
Here is an example of creasing using n=3 cells:



using either addition, which would result in this:



or multiplication, which would result in this:



0	18	0	1	5

Note: For simplicity, creasing with fewer than 1 cell is disallowed, as is creasing with a number of cells greater than or equal to the length of the list. However, a list can by creased by more than half its cell count.

A list of 8 cells can be creased by 5, resulting in a new list of length 5: [0,1,2,3,4,5,6,7] creased by 5 cells using the + operator would give [9,9,9,1,0].

Your Code should print the sequence of crease operations which leads to the maximal reward, multiply your score by 0.8. Example output might look like:

```
crease 5 +
crease 2 *
crease 2 +
crease 1 *
```

Examples

Test your code using these inputs and results, in the form of input - maximum reward:

[0, 1, 2, 3, 4, 5, 6, 7] - 7560
[0, 9, 0, 3, 2, 6, 1, 5] - 1944
[0, 1, 0, 3, 0, 2, 0, 4] - 36
[6, 0, 9, 1, 9, 0, 7, 3] - 11907
[0, 5, 2, 0, 1, 3, 8, 8] - 2560