1) We have bunnies standing in a line, numbered 1, 2, ... The odd bunnies (1, 3, ..) have the normal 2 ears. The even bunnies (2, 4, ..) have 3 ears, because they each have a raised foot. Recursively return the number of "ears" in the bunny line 1, 2, ... n (without loops or multiplication).

Example:

```
Please enter the number of lines (n=): 2
bunnyEars2(0) → 0
bunnyEars2(1) → 2
bunnyEars2(2) → 5
```

2) In this question, you will write a program to find super digit of a number ***X*** using the following rules:

- If ***X*** has only 1 digit, then its super digit is ***X***.
- If ***X*** has more than 1 digit, then its super digit is equal to the super digit of the digit-sum of ***X***.

Example:

```
superDigit(1245) = superDigit(1+2+4+5)
                 = superDigit(12)
                 = superDigit(1+2)
                 = superDigit(3)
                 = 3.
```

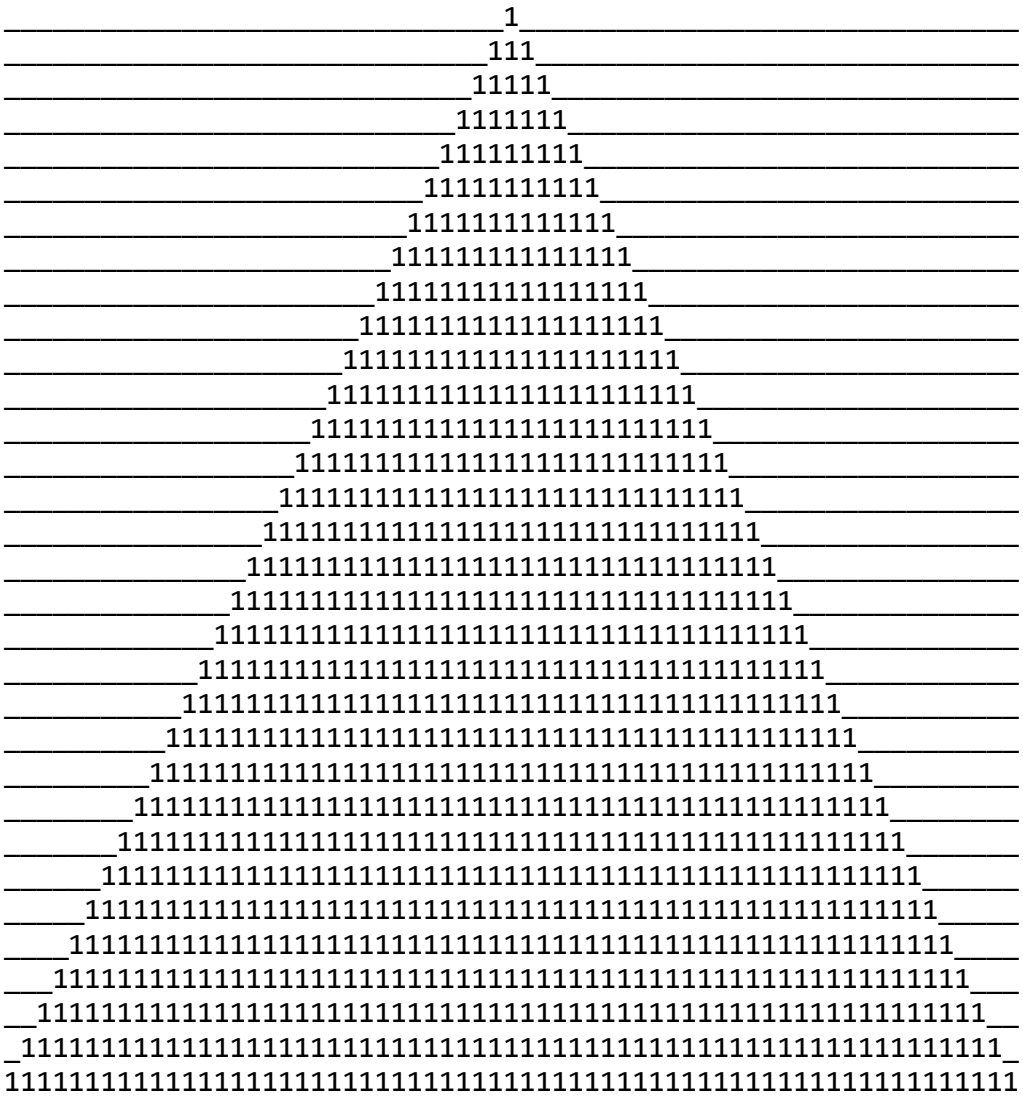
- The number ***X*** will be given to your program with two numbers (***n*** and ***k***) and you will construct the number ***X*** as the number ***n*** concatenated ***k*** times.

Example:

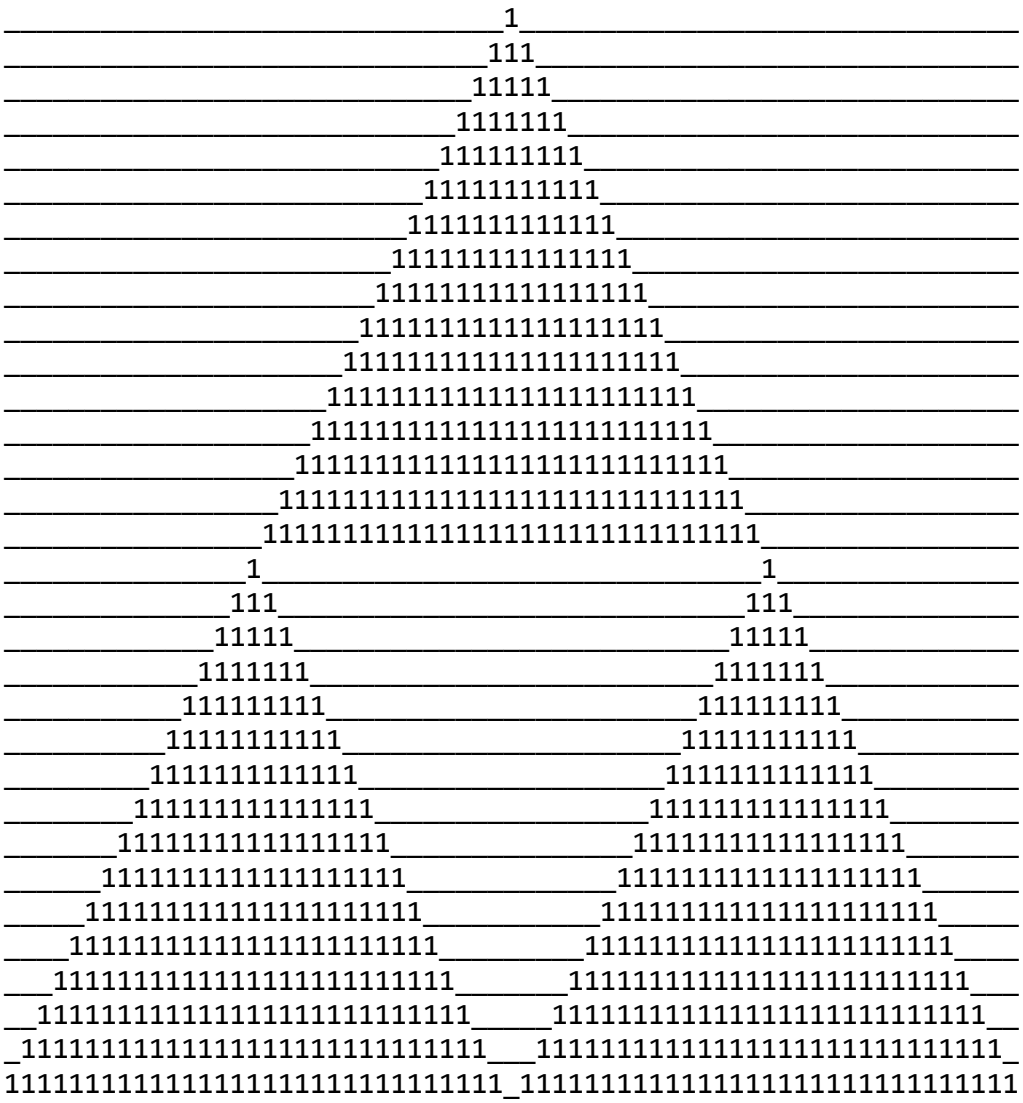
```
Please enter a number (n=) : 123
Please enter repetition factor (k=) : 3
Super digit of number 123123123 is 9.
```

3) In this question, you will print identical triangles nested each other recursively. You will construct the triangles by using two digits **'_'(underscore)** and **'1'**. There will be number of iterations given as an input to your program.

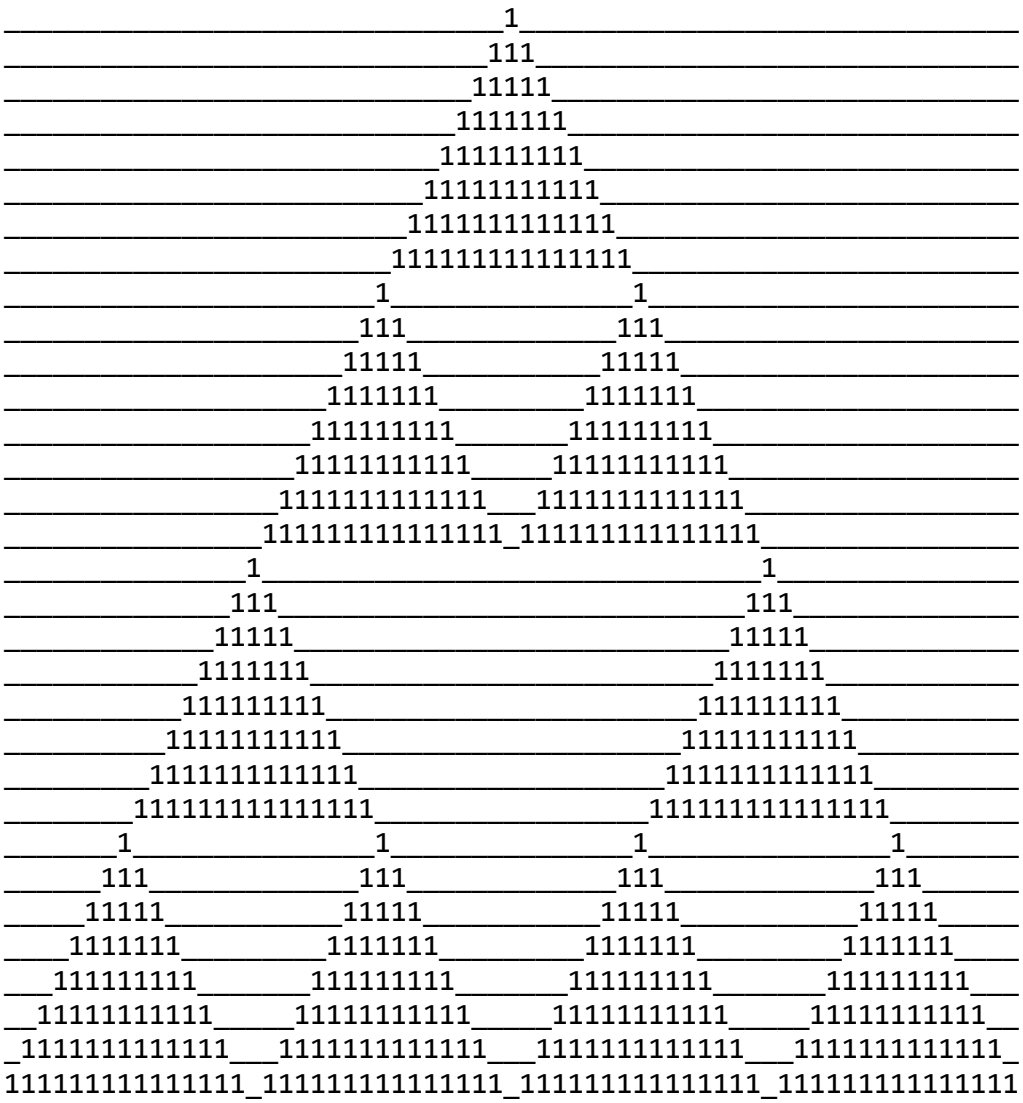
- **If the number of iterations is given as 0**, you will print a simple triangle by using 32 rows and 63 columns in a matrix as the following:



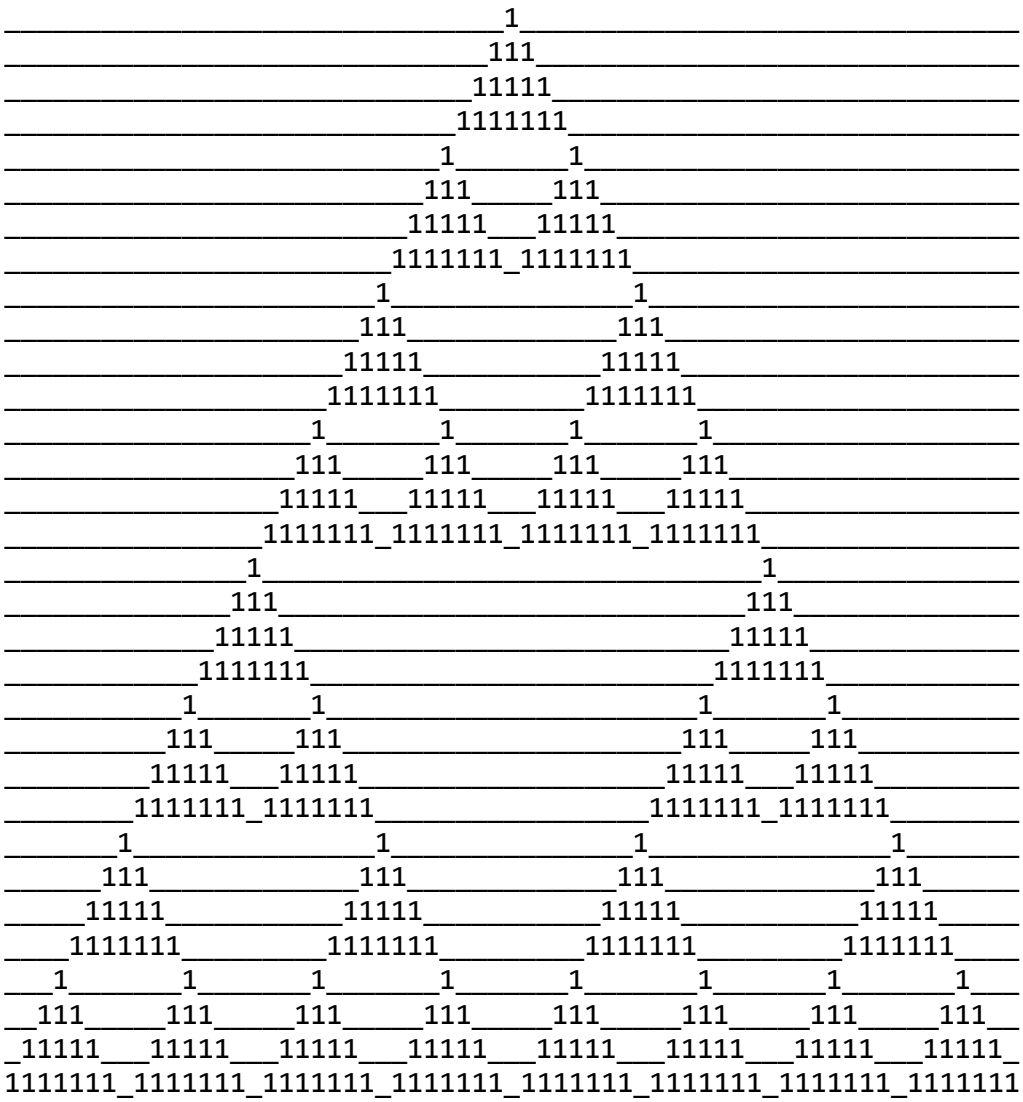
- **If the number of iterations is given as 1**, you will create 3 triangles by calculating their 3 corner points using the original triangle in the previous iteration. It should be noted that the original triangle at iteration 0 will be decomposed three identical triangles as the following:



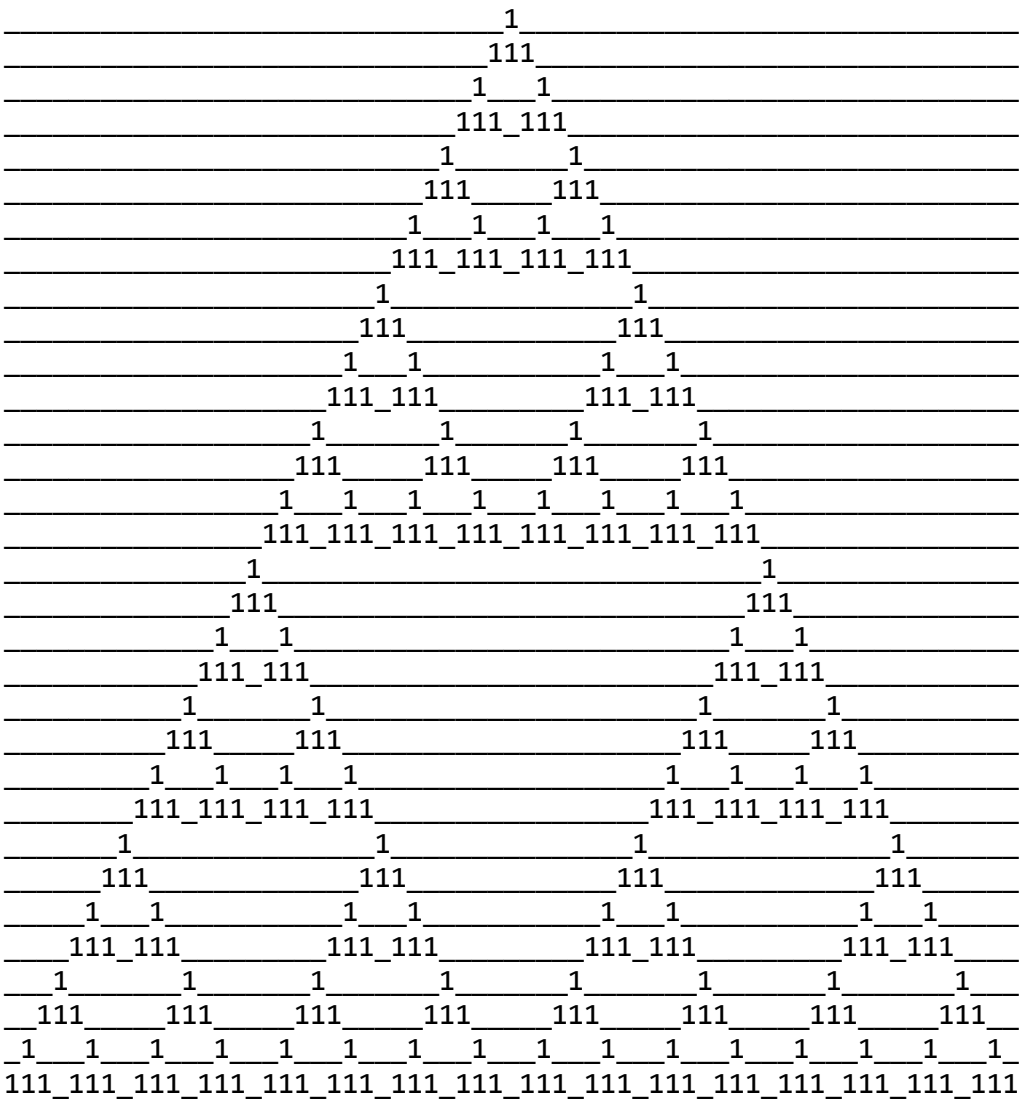
- **If the number of iterations is given as 2**, you will again create 3 triangles for each triangle in the previous iteration by calculating their 3 corner points using the triangles in the previous iteration. It should be noted that each triangle at iteration 1 will be decomposed three identical triangles as the following:



- **If the number of iterations is given as 3**, you will need to print triangles as the following:



- **If the number of iterations is given as 4**, you will need to print triangles as the following:



- The number of iterations given to your program will be less than 5.
- The output will consist of 32 rows and 63 columns, and will be composed of **ones (1)** and **underscores (_)** as in the triangles above.
- Solutions using iterations will not be graded for ALL questions.
- You have to solve the problems by using RECURSION.
- **It should be noted that <u>selected parts</u> will be graded in your homework.**