1. In a small town the population is p0 = 1000 at the beginning of a year. The population regularly increases by 2 percent per year and moreover 50 new inhabitants per year come to live in the town. How many years does the town need to see its population greater or equal to p = 1200 inhabitants? (

   At the end of the first year there will be:

   1000 + 1000 * 0.02 + 50 => 1070 inhabitants

   At the end of the 2nd year there will be:

   1070 + 1070 * 0.02 + 50 => 1141 inhabitants (number of inhabitants is an integer)

   At the end of the 3rd year there will be:

   1141 + 1141 * 0.02 + 50 => 1213

   It will need 3 entire years.

   More generally given parameters:

   p0, percent, aug (inhabitants coming each year), p (population to surpass)

   the function nb_year should return n number of entire years needed to get a population greater or equal to p.

   aug is an integer, percent a positive or null number, p0 and p are positive integers (> 0)

   Examples:

   nb_year(1500, 5, 100, 5000) -> 15

   nb_year(1500000, 2.5, 10000, 2000000) -> 10

   Note: Don't forget to convert the percent parameter as a percentage in the body of your function: if the parameter percent is 2 you have to convert it to 0.02.

   def nbYear(p0, percent, aug, p) :

       // your code

2. The new "Avengers" movie has just been released! There are a lot of people at the cinema box office standing in a huge line. Each of them has a single 100, 50 or 25 dollars bill. An "Avengers" ticket costs 25 dollars.

   Vasya is currently working as a clerk. He wants to sell a ticket to every single person in this line.

   Can Vasya sell a ticket to each person and give the change if he initially has no money and sells the tickets strictly in the order people follow in the line?

   Return YES, if Vasya can sell a ticket to each person and give the change with the bills he has at hand at that moment. Otherwise return NO.
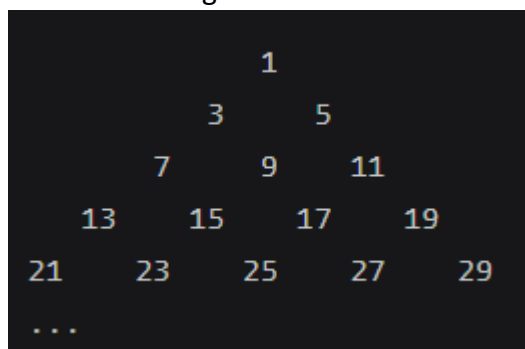
   ###Examples:
   tickets([25, 25, 50]) // => YES
   tickets([25, 100]) // => NO. Vasya will not have enough money to give change to 100 dollars
   tickets([25, 25, 50, 50, 100]) // => NO. Vasya will not have the right bills to give 75 dollars of change (you can't make two bills of 25 from one of 50)

   def tickets(peopleInLine):
     // your code

3. Given the triangle of consecutive odd numbers:

   ```
                 1
             3       5
          7       9       11
       13      15      17      19
    21      23      25      27      29
    . . .
   ```

   Calculate the row sums of this triangle from the row index (starting at index 1) e.g.:

   ```
   rowSumOddNumbers(1); // 1
   rowSumOddNumbers(2); // 3 + 5 = 8
   ```

   def rowSumOddNumbers(n) :
           // TODO

   And don't forget to print the triangle too (the size is base on the number of rows user input)