1. A number is called **digit-increasing** if it is equal to n + nn + nnn + ... for some digit n between 1 and 9. For example 24 is digit-increasing because it equals 2 + 22 (here n = 2) Write a function called **isDigitIncreasing** that returns 1 if its argument is digit-increasing otherwise, it returns 0.
The signature of the method is int isDigitIncreasing(int n)

Examples:

if n is	**then function returns**	**reason**
7	1	because 7 = 7 (here n is 7)
36	1	because 36 = 3 + 33
984	1	because 984 = 8 + 88 + 888
7404	1	because 7404 = 6 + 66 + 666 + 6666
37	0	because there is no digit n (1-9) such that n + nn = 37

---

2. An array is defined to be **twin paired** if its even-valued elements (if any) are in scending order and its odd-valued elements (if any) are in ascending order. The array {-6, 12, 1, 24, 3, 5} is twin paired because the even-valued elements (-6, 12, 24) are in ascending order and so are the odd-valued elements (1, 3, 5). However, the array {3, 2, 1} is not twin aired because the odd numbers are not in ascending order.

Write a function named *isTwinPaired* that returns 1 if its array argument is twin paired, therwise it returns 0. If you are programming in Java or C#, the function signature is int isTwinPaired(int[ ] a) If you are programming in C or C++, the function signature is int isTwinPaired(int a[ ], int len) where len is the number of elements in a.

Other twin paired arrays include:
{2, 4, 32},
{2, 2, 2, 1, 1, 1},
{1, 19, 23},
{1, 2},
{2, 1},
{8},
{17},
{ }

3. An **Olympic array** is defined to be an array in which every value is greater than or equal to the sum of the values less than it. The sum of the values less than the minimum value in the array is defined to be 0.
For example, {3, 2, 1} is an Olympic array because

1. 1 is the minimum value and by definition the sum of the values less than it is 0. Since 1 is greater than 0, it satisfies the condition.
2. There is only one value less than 2 and 2 is greater than it, so the value 2 satisfies the condition.
3. The values 1 and 2 are less than 3 and 3 is equal to their sum, so the value 3 satisfies the condition.

Hence all elements of the array satisfy the conditions and the array is an Olympic array. {2, 2, 1, 1} is also an Olympic array because the values less than 2 sum to 2. {1, 1000, 100, 10000, 2} is also an Olympic array. However, {1, 99, 99, 1000, 100, 10000, 2} is **not** an Olympic array because the sum of the numbers less than 100 (99+99+1) is greater than 100. Please be sure that your function detects that this is not an Olympic array! {1, 2, 1, 3, 2} is not an Olympic array because 3 is not greater than or equal to 1+2+1+2. {1, 2, -1, 2, 2} is not an Olympic array because -1 is the minimum value but it is not greater than or equal to 0. Write a function named *isOlympic* that returns 1 if its array argument is an Olympic array, otherwise it returns 0.
If you are writing in Java or C#, the function signature is int isOlympic (int[ ] a)
If you are writing in C or C++, the function signature is int isOlympic(int a[ ], int len) where len is the number of elements in the array.