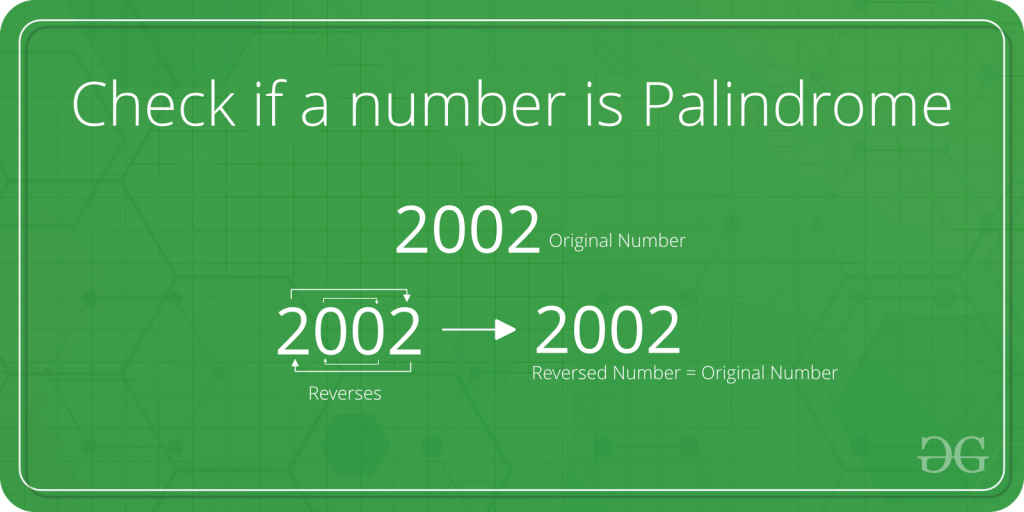
**Palindrome Numbers**

Given an integer, write a function that returns true if the given number is palindrome, else false. For example, 12321 is palindrome, but 1451 is not palindrome.



A simple approach to check if a number is Palindrome or not . This approach can be used when the number of digits in the given number is less than 10^18 because if the number of digits of that number exceeds 10^18, we can’t take that number as an integer since the range of long long int doesn’t satisfy the given number.

To check whether the given number is palindrome or not we will just reverse the digits of the given number and check if the reverse of that number is equal to the original number or not . If reverse of number is equal to that number than the number will be Palindrome else it will not a Palindrome.

C++Java

// C++ program to check if a number is Palindrome

#include <iostream>

using namespace std;

// Function to check Palindrome

bool checkPalindrome(int n)

{

int reverse = 0;

int temp = n;

while (temp != 0) {

reverse = (reverse \* 10) + (temp % 10);

temp = temp / 10;

}

return (reverse

== n); // if it is true then it will return 1;

// else if false it will return 0;

}

int main()

{

int n = 7007;

if (checkPalindrome(n) == 1) {

cout << "Yes\n";

}

else {

cout << "No\n";

}

return 0;

}

**Output**

Yes

**Time Complexity :**O(log(n)) or O(Number of digits in a given number)

**Auxiliary space**: O(1) or constant