**Day-1:**

<https://leetcode.com/problems/palindrome-number>

Given an integer x, return true*if*x*is a*

***palindrome***

*, and*false*otherwise*.

**Example 1:**

**Input:** x = 121

**Output:** true

**Explanation:** 121 reads as 121 from left to right and from right to left.

**Solution:**

**class Solution {**

**public boolean isPalindrome(int x) {**

**if (x < 0) {**

**return false;**

**}**

**int temp = x, rev = 0;**

**while (x > 0) {**

**int rem = x % 10;**

**rev = rev \* 10 + rem;**

**x = x / 10;**

**}**

**return temp == rev;**

**}**

**}**

**Day 2**

**Question:**

[**https://leetcode.com/problems/find-the-index-of-the-first-occurrence-in-a-string**](https://leetcode.com/problems/find-the-index-of-the-first-occurrence-in-a-string)

**Solution:**

class Solution {

public:

    int strStr(string haystack, string needle) {

        int haystacklength=haystack.length();

        int needlelength=needle.length();

        if(needlelength==0)

        {

            return 0;

        }

        for(int i=0;i<=haystack.length-needle.length;i++)

        {

                for(int j=0;j<needlelength;j++)

                {

                    if(haystack.charAt(i+j)==needle.charAt(j))

                    {

                        break;

                    }

                }

                if(j==needlelength)

                {

                    return i;

                }

        }

return -1;

        public static void main(String []args)

        {

            String haystack1="sadbutsad";

            String needle1="sad";

            System.out.println("hence the  output is:"+strStr(haystack1,needle 1));

        }

    }

};

**DAY-3**

**Question:**

<https://leetcode.com/problems/climbing-stairs>

**Solution:**

class Solution

{

public int climbStairs(int n)

{

if(n==1)

{

return 1;

}

if(n==2)

{

return 2;

}

int n1=2;

int n2=1;

for(int i=3;i<=n;i++)

{

int current=n1+n2;

n2=n1;

n1=current;

}

return n1;

}

public static void main(String args[])

{

Solution ob=new Solution();

System.out.println(ob.climbStairs(5));

}

}

Output:

