PYTHON LAB PROGRAM 8

***8*) Aim: Demonstration of classes and methods with polymorphism and overriding**

a) Write a python program to find the whether the given input is palindrome or not (for both string and integer) using the concept of polymorphism and inheritance.

Code:

```
class Palindrome:
    def __init__(self, data):
        self.data = data

    def is_palindrome(self):
        return str(self.data) == str(self.data)[::-1]


class StringPalindrome(Palindrome):
    def __init__(self, data):
        super().__init__(data)

    def is_palindrome(self):
        return super().is_palindrome()


class IntegerPalindrome(Palindrome):
    def __init__(self, data):
        super().__init__(data)

    def is_palindrome(self):
        return super().is_palindrome()


def check_palindrome(input_data):
    if isinstance(input_data, int):
        obj = IntegerPalindrome(input_data)
    elif isinstance(input_data, str):
        obj = StringPalindrome(input_data)
    else:
        raise ValueError("Input should be an integer or string")

    return obj.is_palindrome()

st = input("Enter a string : ")
stObj = StringPalindrome(st)
if stObj.is_palindrome():
    print("Given string is a Palindrome")
else:
    print("Given string is not a Palindrome")

val = int(input("Enter a integer : "))
intObj = IntegerPalindrome(val)
if intObj.is_palindrome():
    print("Given integer is a Palindrome")
else:
```

```
print("Given integer is not a Palindrome")
```

OUTPUT:

Case1:

```
Enter a string : AbA
Given string is a Palindrome
Enter a integer : 121
Given integer is a Palindrome
```

Case2:

```
Enter a string : VTU
Given string is not a Palindrome
Enter a integer : 123
Given integer is not a Palindrome
```