Bangladesh University of Business and Technology

Lab Report For AI

Course : Artificial Intelligence Lab (CSE 352)

Problem Name:

1. Fibonacci nth position and sum.
2. Fibonacci using recursion.
3. Count vowel,consonant, space.
4. Strong Number.
5. Sum nth position using function and recursive.
6. Reverse number.
7. Palindrome check number.
8. Palindrome check number using recursive.
9. Sum of even number in a range.

Submitted by:

Name : Md. Rashidul Haq
Id : 19202103356
Intake : 44
Sec : 4
Email:19202103356@cse.bubt.edu.bd

Submitted To:

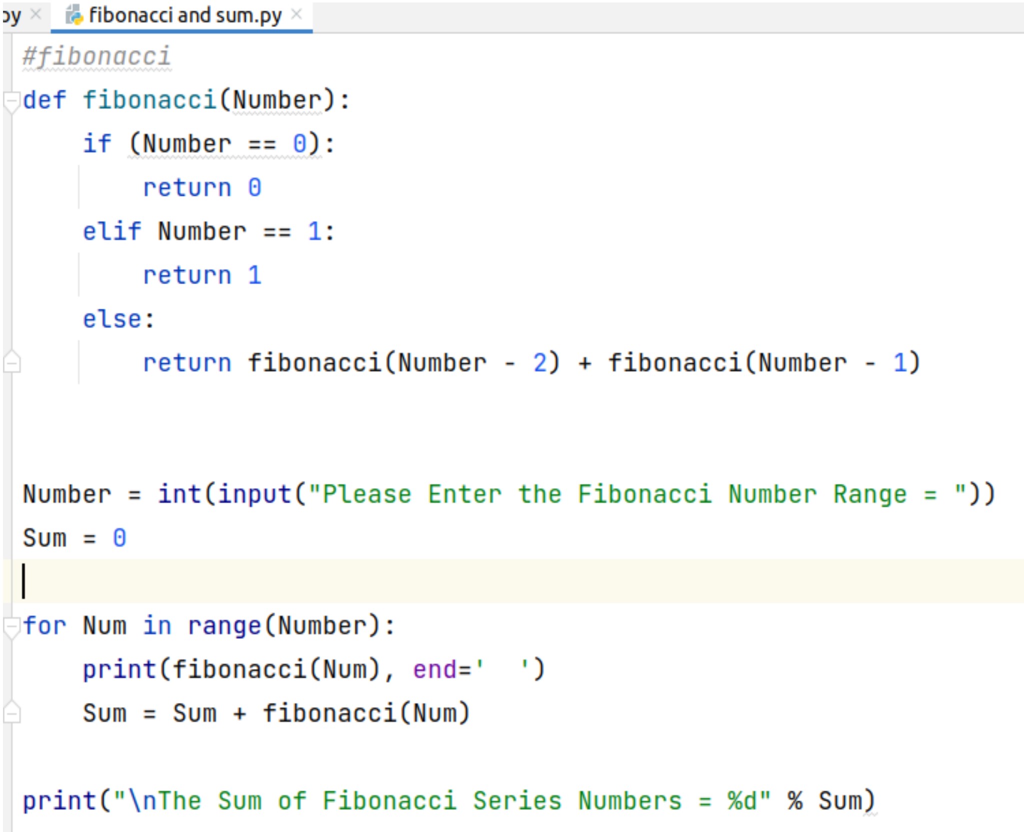
Humayra Ahmed
(Assistant Professor)

Bangladesh University of
Business & Technology (BUBT)

Problem 1:

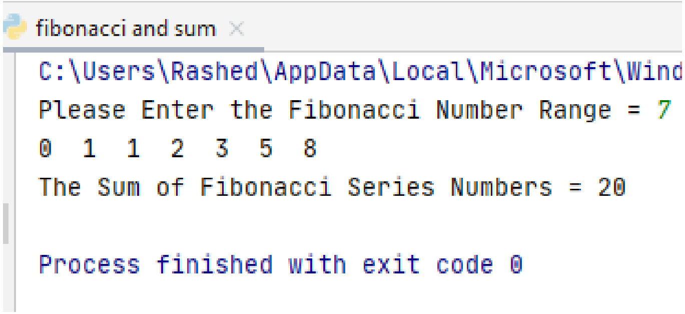
Problem name: Fibonacci nth position and sum

Code:



```python
#fibonacci
def fibonacci(Number):
    if (Number == 0):
        return 0
    elif Number == 1:
        return 1
    else:
        return fibonacci(Number - 2) + fibonacci(Number - 1)


Number = int(input("Please Enter the Fibonacci Number Range = "))
Sum = 0

for Num in range(Number):
    print(fibonacci(Num), end='  ')
    Sum = Sum + fibonacci(Num)

print("\nThe Sum of Fibonacci Series Numbers = %d" % Sum)
```

Output:



```
C:\Users\Rashed\AppData\Local\Microsoft\Wind
Please Enter the Fibonacci Number Range = 7
0  1  1  2  3  5  8
The Sum of Fibonacci Series Numbers = 20

Process finished with exit code 0
```

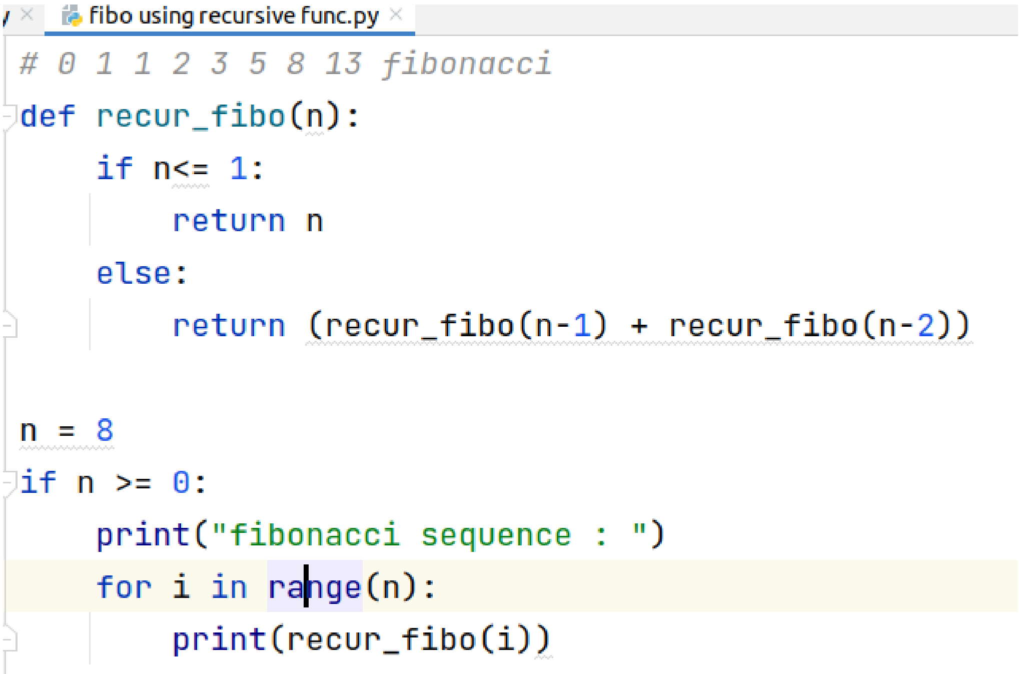
Discussion:

- At first define function name 'fibonacci' and parameter name ‘number’. Then using recursive ,recursive base case ‘number == 0’.
- If number == 0 then return 0 and close function . and not if then find a fibonacci number . then call the function.
- input range of fibonacci which we want (here range = 7) , and then find all fibonacci number sum. Then print fibonacci series and sum.

Problem 2:

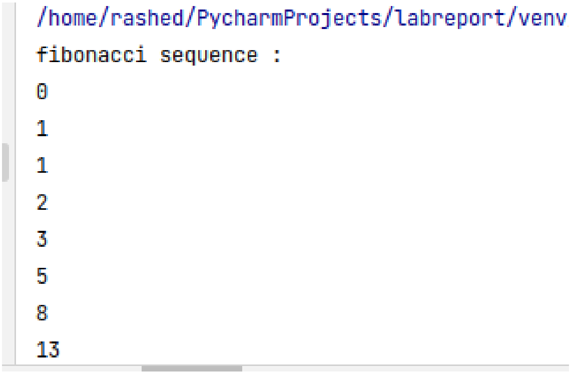
Problem name: Fibonacci using recursion.

Code:



```python
# 0 1 1 2 3 5 8 13 fibonacci
def recur_fibo(n):
    if n<= 1:
        return n
    else:
        return (recur_fibo(n-1) + recur_fibo(n-2))

n = 8
if n >= 0:
    print("fibonacci sequence : ")
    for i in range(n):
        print(recur_fibo(i))
```

Output:



```
/home/rashed/PycharmProjects/labreport/venv
fibonacci sequence :
0
1
1
2
3
5
8
13
```

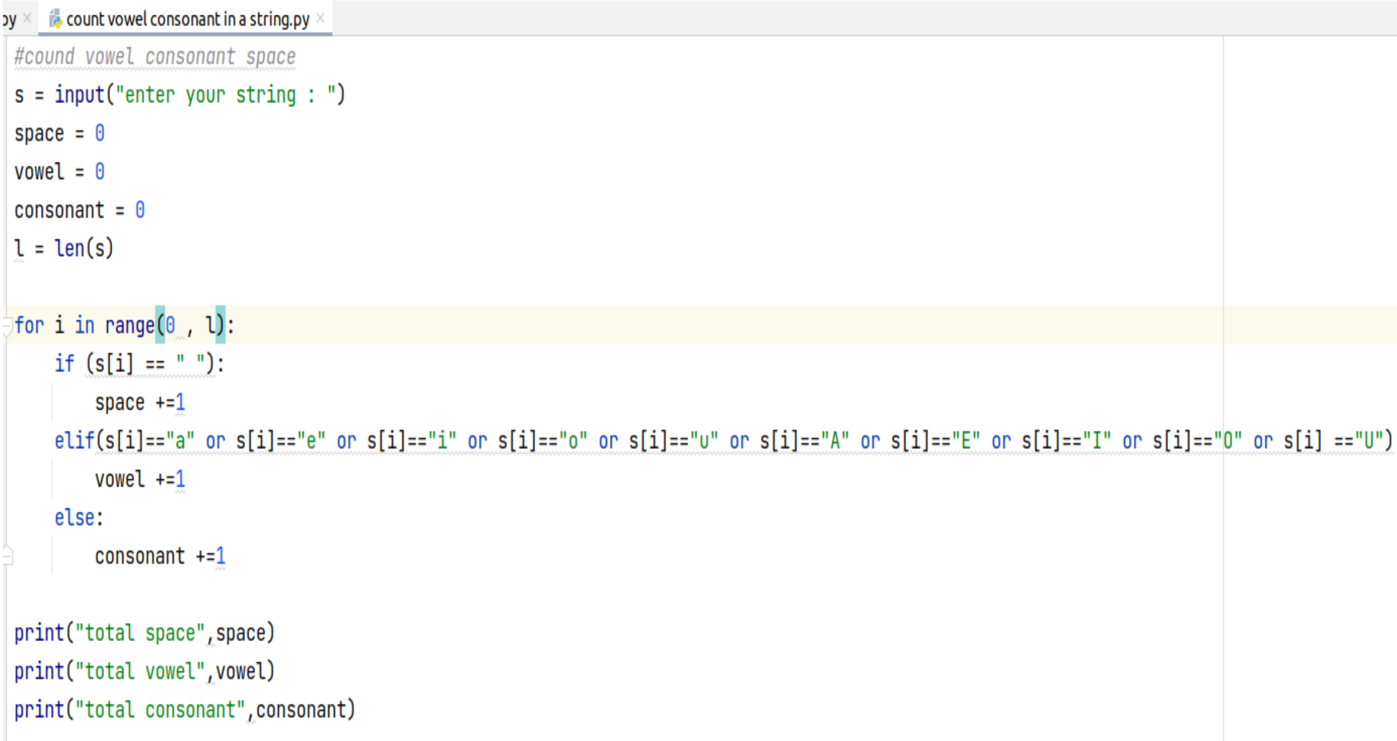
Discussion:

- At first get recursive function base case is n<=1. If n<1 so close function and return n value. Not if find recursive.
- Then call function for print sequence .I use if condition , n value must be n>=0
- then use for loop start i and end n value , here n = 8.
- And print fibonacci sequence.

Problem 3:

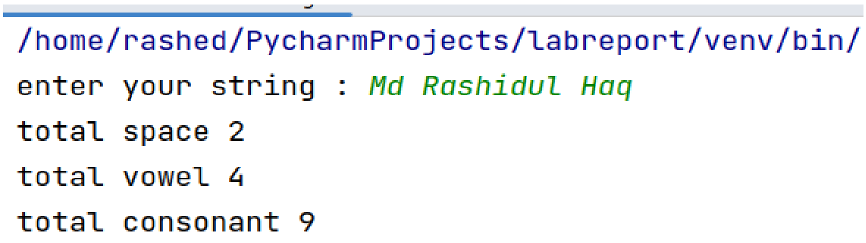
Problem name: Count vowel,consonant, space.

Code:



```
#cound vowel consonant space
s = input("enter your string : ")
space = 0
vowel = 0
consonant = 0
l = len(s)

for i in range(0 , l):
    if (s[i] == " "):
        space +=1
    elif(s[i]=="a" or s[i]=="e" or s[i]=="i" or s[i]=="o" or s[i]=="u" or s[i]=="A" or s[i]=="E" or s[i]=="I" or s[i]=="O" or s[i] =="U")
        vowel +=1
    else:
        consonant +=1

print("total space",space)
print("total vowel",vowel)
print("total consonant",consonant)
```

Output:



```
/home/rashed/PycharmProjects/labreport/venv/bin/
enter your string : Md Rashidul Haq
total space 2
total vowel 4
total consonant 9
```

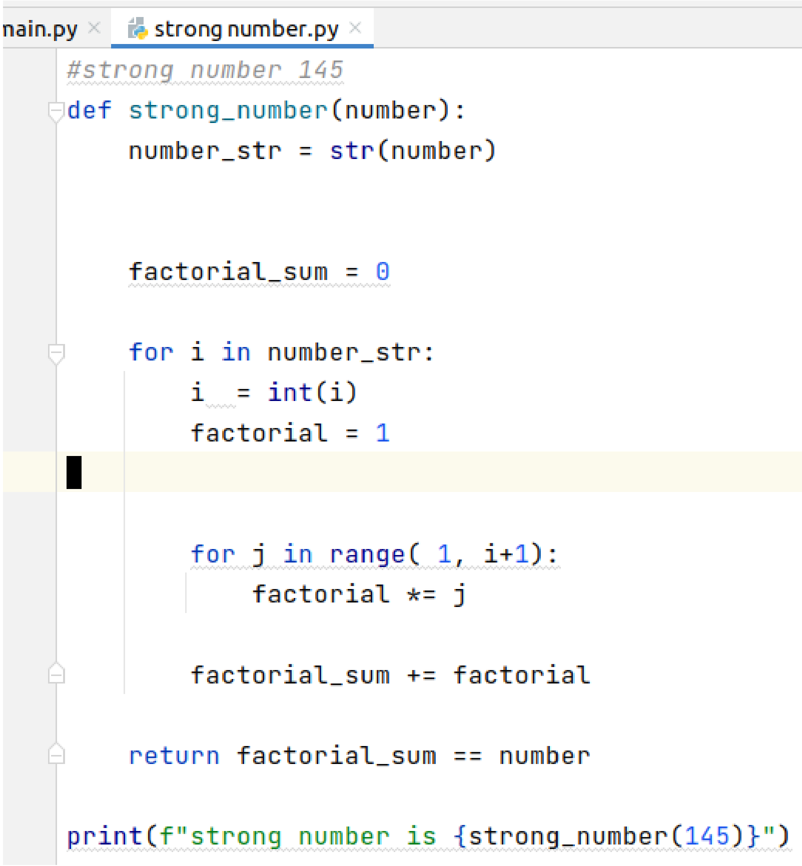
Discussion:

- First input number.
- Second all initial value is 0. And also length value for count length.
- Then I use for loop this start i and end length here i=0.
- Then use if condition if get “space” and check length up to last index then space+1.
- And similarly check vowel and consonant if gate then vowel+1 and consonant+1.
- At last print space , vowel, consonant.

Problem 4:

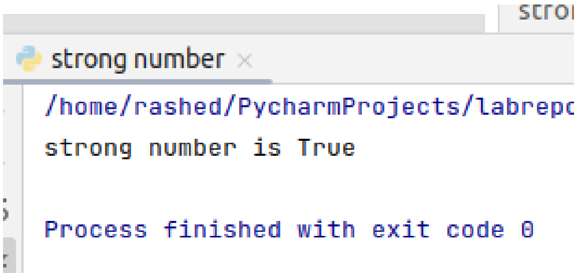
Problem name: Strong Number.

Code:



```
#strong number 145
def strong_number(number):
    number_str = str(number)


    factorial_sum = 0

    for i in number_str:
        i  = int(i)
        factorial = 1


        for j in range( 1, i+1):
            factorial *= j

        factorial_sum += factorial

    return factorial_sum == number

print(f"strong number is {strong_number(145)}")
```

Output:



```
/home/rashed/PycharmProjects/labrepc
strong number is True

Process finished with exit code 0
```

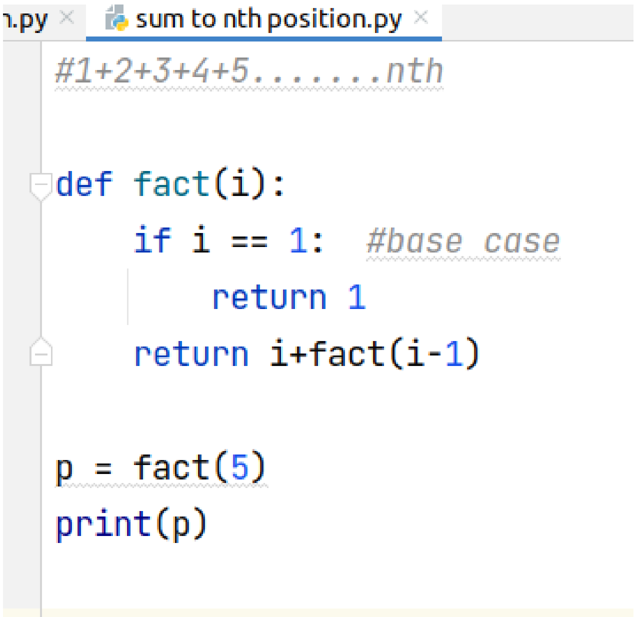
Discussion:

- Define function and initially factorial_sum=0.
- Then i use for loop. This loop is start i and end number_str and store value factorial=1.
- Then use another for loop name j .
- This range 1 to last value.and factorial=factorial*j. Then calculate factorial_sum + factorial. And return factorial_sum=number.
- After print boolean if number is strong then print strong number is true otherwise false.

Problem 5:

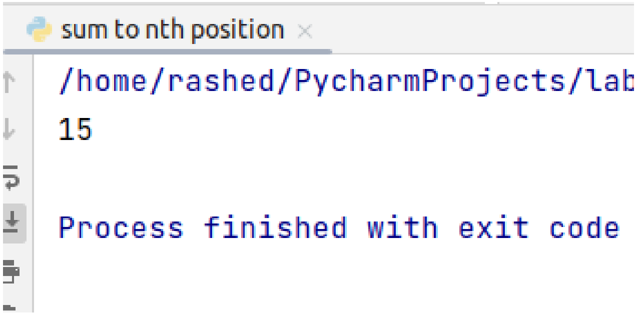
Problem name: Sum nth position using function and recursive.

Code:



```
#1+2+3+4+5.......nth

def fact(i):
    if i == 1:   #base case
        return 1
    return i+fact(i-1)

p = fact(5)
print(p)
```

Output:



```
/home/rashed/PycharmProjects/lab
15

Process finished with exit code
```

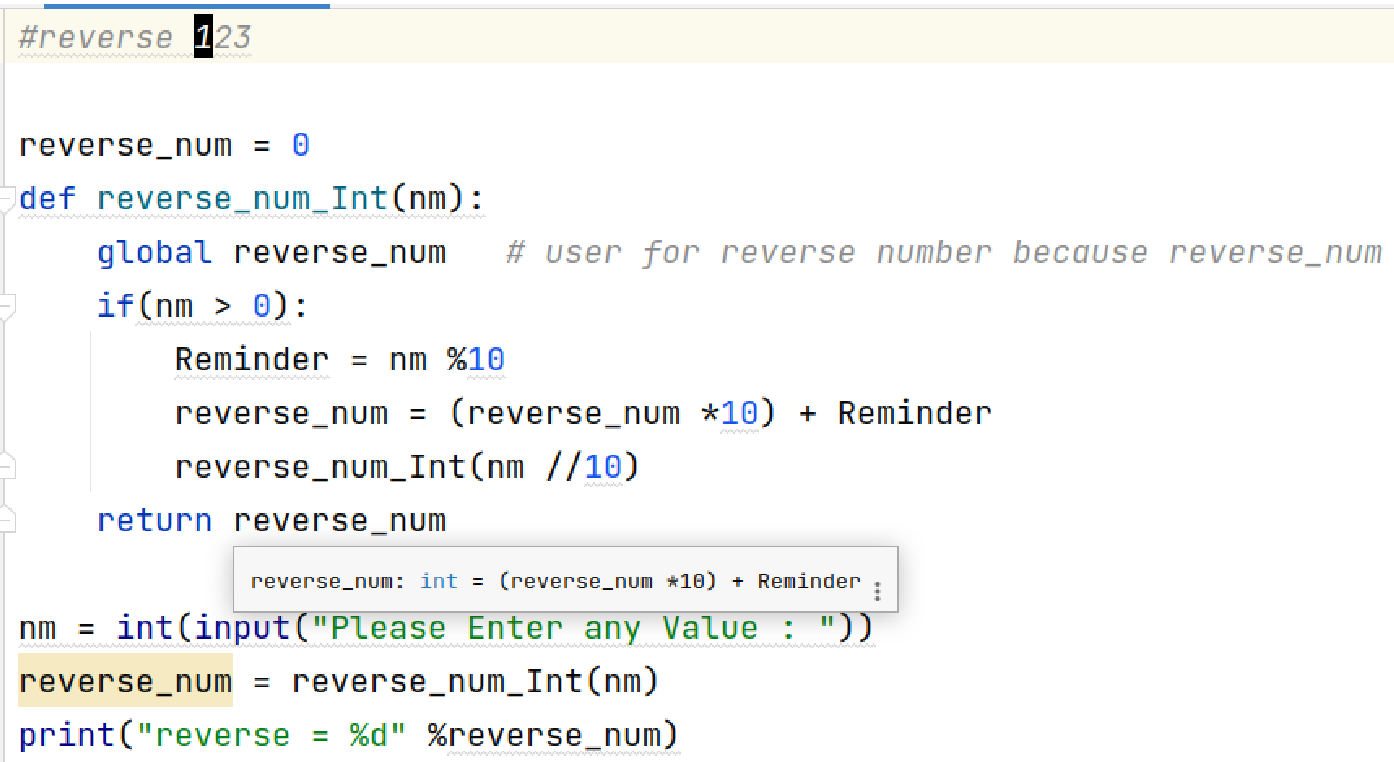
Discussion:

- I use recursive function for sum of given position.
- At first define function name fact and this parameter i. Base case is i=1, if i value 1 then return 1 and end this function.
- And return i+fact(i-1).
- Then call function and also include n value and print sum .

Problem 6:

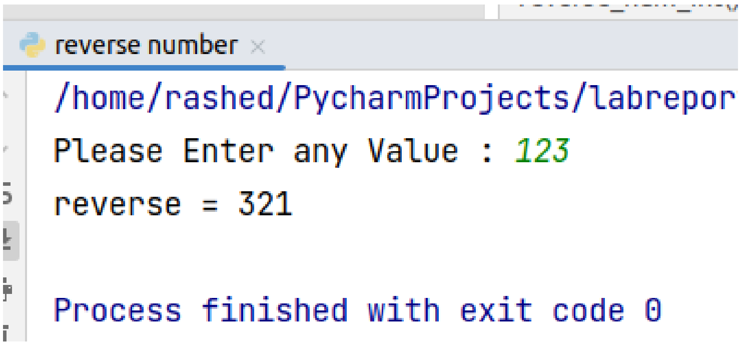
Problem name: Reverse number.

Code:



```
#reverse 123

reverse_num = 0
def reverse_num_Int(nm):
    global reverse_num   # user for reverse number because reverse_num
    if(nm > 0):
        Reminder = nm %10
        reverse_num = (reverse_num *10) + Reminder
        reverse_num_Int(nm //10)
    return reverse_num

nm = int(input("Please Enter any Value : "))
reverse_num = reverse_num_Int(nm)
print("reverse = %d" %reverse_num)
```

Output:



```
/home/rashed/PycharmProjects/labrepor
Please Enter any Value : 123
reverse = 321

Process finished with exit code 0
```

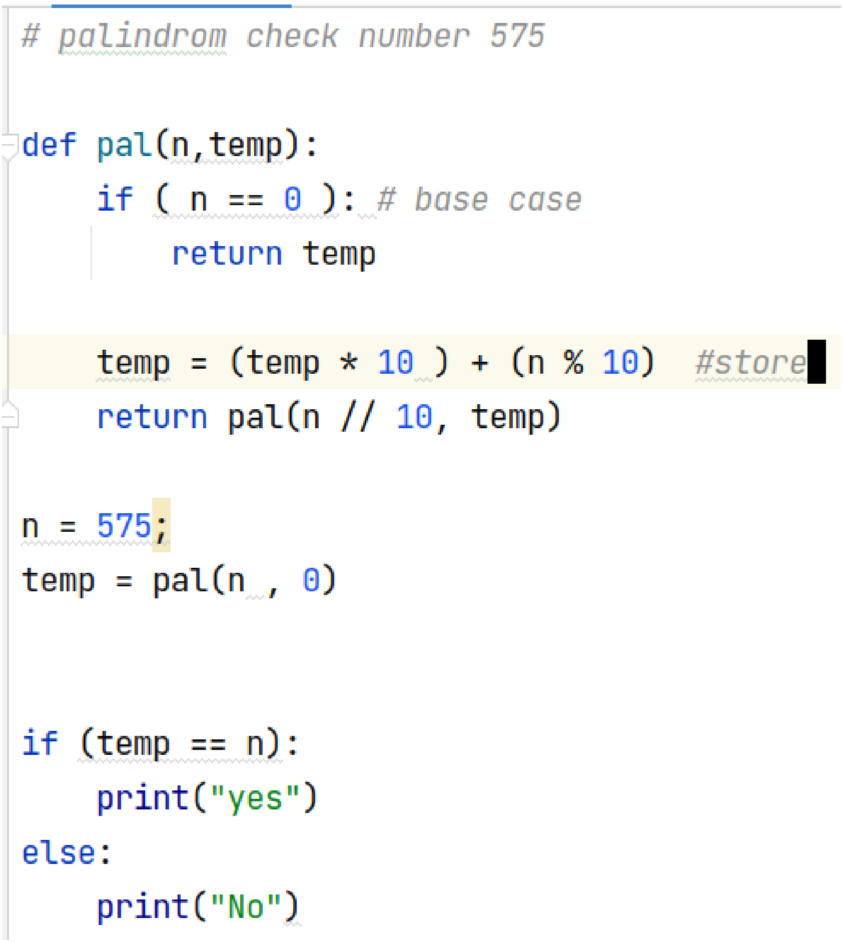
Discussion:

- First reverse_num initial value is 0.
- After define function and reverse_num global because of we can’t use reverse_num in function without global so i use global reverse_num.
- And used if condition number must be grater then 0. reminder = number modulus 10 and reverse_num*10+reminder.
- Then call function input a number .
- Print this reverse number.

Problem 7:

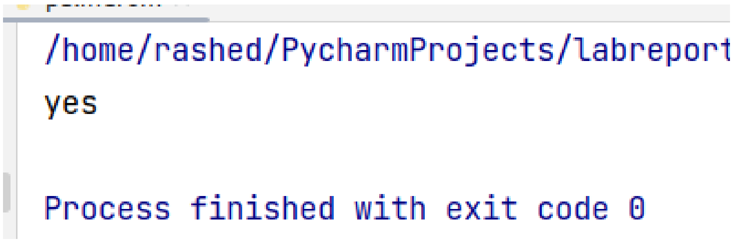
Problem name: Palindrome check number

Code:



```
# palindrom check number 575

def pal(n,temp):
    if ( n == 0 ): # base case
        return temp

    temp = (temp * 10 ) + (n % 10)  #store
    return pal(n // 10, temp)

n = 575;
temp = pal(n , 0)


if (temp == n):
    print("yes")
else:
    print("No")
```

Output:



```
/home/rashed/PycharmProjects/labreport
yes

Process finished with exit code 0
```

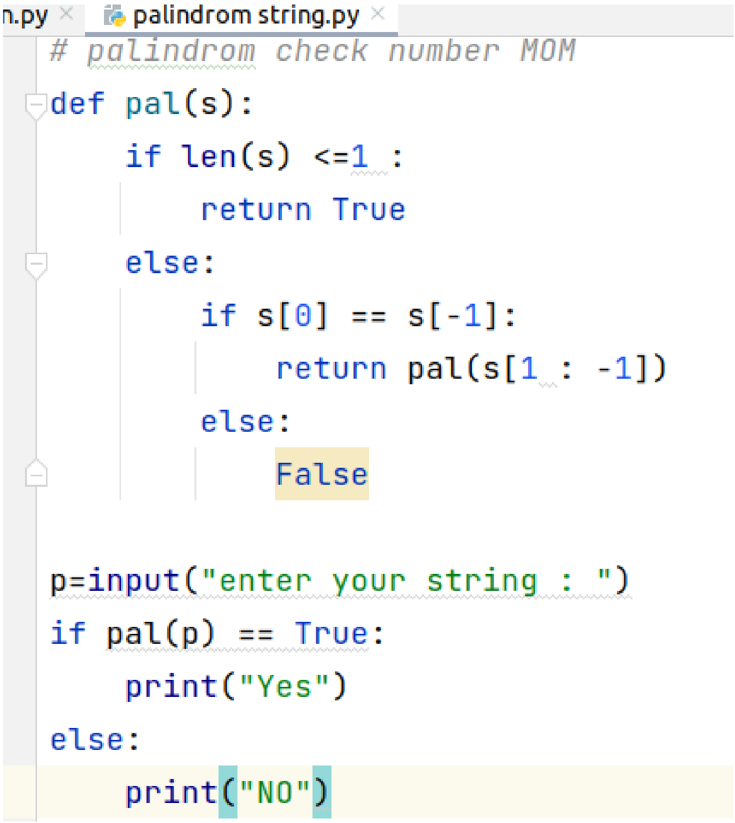
Discussion:

- Use recursive function. Base case is n=0. If n=0 then return temp value. Then temp value is (temp*10)+(n%10) mainly we store value in temp.
- Then call temp value and set a value to check this number is palindrome or not.
- If temp value equal given value so we can say palindrome yes or not match print no.

Problem 8:

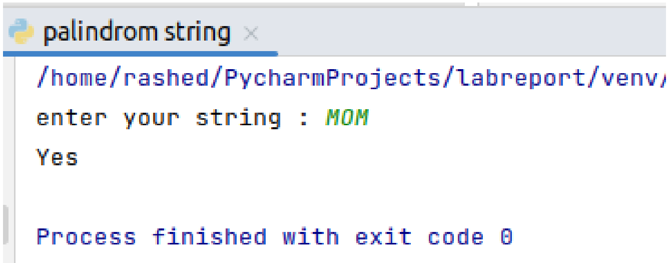
Problem name: Palindrome check string using recursive.

Code:



```
# palindrom check number MOM
def pal(s):
    if len(s) <=1 :
        return True
    else:
        if s[0] == s[-1]:
            return pal(s[1 : -1])
        else:
            False

p=input("enter your string : ")
if pal(p) == True:
    print("Yes")
else:
    print("NO")
```

Output:



```
/home/rashed/PycharmProjects/labreport/venv/
enter your string : MOM
Yes

Process finished with exit code 0
```

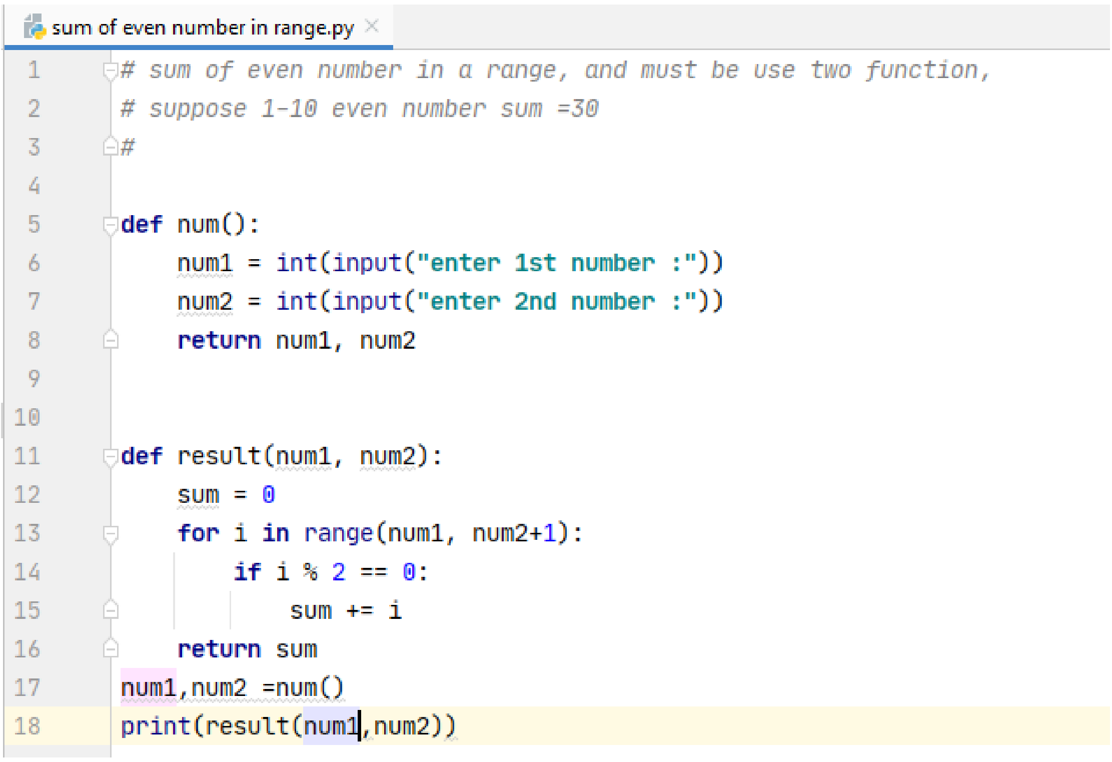
Discussion:

- Define recursive function . and base case this length <=1.
- Here one character if user input so this character is palindrome string .
- Else one more character user input that check operation.check first character and last character. If first character and last character will same then check second character and before last character,if will same then check after it will be check middle point of character then return true of false .
- Input a string then check condition will true it return true so that character is palindrome character YES.
- else get false then print NO.

Problem 8:

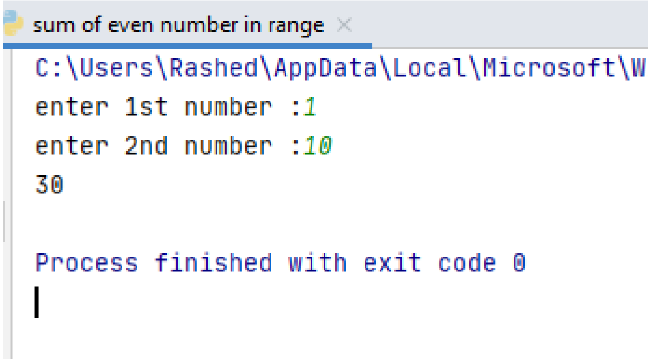
Problem name: Sum of even number in a range.

Code:



```
# sum of even number in a range, and must be use two function,
# suppose 1-10 even number sum =30
#

def num():
    num1 = int(input("enter 1st number :"))
    num2 = int(input("enter 2nd number :"))
    return num1, num2


def result(num1, num2):
    sum = 0
    for i in range(num1, num2+1):
        if i % 2 == 0:
            sum += i
    return sum
num1,num2 =num()
print(result(num1,num2))
```

Output:



```
C:\Users\Rashed\AppData\Local\Microsoft\W
enter 1st number :1
enter 2nd number :10
30

Process finished with exit code 0
```

Discussion:

- First function use for input 1st number and last number.
- Then second function for calculate in a range sum of even number.
- Then use for loop start i in range 1st number and end last+1.
- And then check even number and store sum and also return sum.
- After parameter call and print sum of all even number in a range.