Lab5B01

Explanation:-

This is quite simple solution we reversed userInput String and compared with userInput String,if both are equal returns true

A screenshot of a computer

Description automatically generated

usrInput=input("Enter a string to see if it is a palindrome: ")  
reversedInput=usrInput[::-1]  
if(usrInput==reversedInput):  
 print(usrInput,"is one")  
else:  
 print(usrInput,"is not one")



A white background with text

Description automatically generated

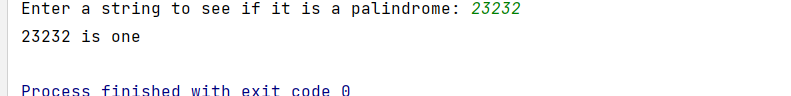
A screen shot of a computer

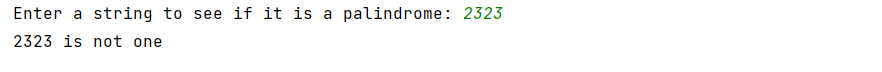
Description automatically generatedA close up of words

Description automatically generated

A white background with black text

Description automatically generated





Question: Lab5B02:-

This is bit tricky and funny,We have used multiple functions as mentioned below:

**Is\_Prime:-It will check generated number is prime or not and returns the same**

**Generate\_Random\_number():-In this function it will generate a randomNumber between the given digits**

**countCheck():-It will check count of the primeNumbers generated is mached with expectedNumber(UserInput)**

**number\_of\_primes\_in\_the\_Range():-returns unique primes by using set() and casted to list and printed list manipulations**

**main():-**

*#Name:Pravalika Rao Chitneni  
#CompletionDate:18/11/2023 10.40AM*import random  
  
*#is\_prime(parameter)  
#Accepts usrInput  
#Returns boolean true/false  
#This method checks passed parameter is primeNumber or Not*def is\_prime(generatedNumber):  
 if generatedNumber <= 1:  
 return False  
 for i in range(2, int(generatedNumber\*\*0.5) + 1):  
 if generatedNumber % i == 0:  
 return False  
 return True  
  
*#generate\_random\_number(parameter1,parameter2)  
#Accepts two userInputs  
#Returns primeNumber which is generated randomly*def generate\_random\_number(usrInput\_2, usrInput\_3):  
 while True:  
 generated\_RandomNumber = random.randint(usrInput\_2, usrInput\_3)  
 if is\_prime(generated\_RandomNumber):  
 return generated\_RandomNumber  
  
*#countCheck(parameter\_1,parameter\_2)  
#it will checks length of the primesnumbers count and checks count*def countCheck(usrInput\_2, usrInput\_3):  
 count = 0  
 for num in range(usrInput\_2, usrInput\_3 + 1):  
 if is\_prime(num):  
 count += 1  
 return count  
  
*#number\_of\_primes\_in\_the\_range(userInput1, userInput2, userInput3)  
#In this method we need to collect the primeNumbers which are generated randomly  
#Here we used set to collect the unique numbers   
#And the set was type casted to list  
#List was manipulated   
#printed list and reversed list and find the max and min of the list*def number\_of\_primes\_in\_the\_range(userInput1, userInput2, userInput3):  
 primes = set()  
 while len(primes) < userInput1:  
 random\_prime = generate\_random\_number(userInput2, userInput3)  
 primes.add(random\_prime)  
 resultList= list(primes)  
 print("The generated random number list:")  
 print(resultList)  
 print(resultList[::-1])  
 print("The minimum and maximum prime numbers are :",max(resultList),"and",min(resultList))  
  
  
*#main()  
#Is's more like a wrapper method  
#because we have called all the methods  
#and achieved the output*def main(input\_values):  
  
 while True:  
 try:  
 if len(input\_values) != 3:  
 raise ValueError()  
  
 usrInput\_1,usrInput\_2, usrInput\_3=input\_values  
 if usrInput\_1 <= 0:  
 raise ValueError()  
  
 if usrInput\_2 >= usrInput\_3:  
 raise ValueError()  
  
 if countCheck(usrInput\_2,usrInput\_3) < usrInput\_1:  
 raise ValueError()  
 break  
 except ValueError as e:  
 print()  
  
 number\_of\_primes\_in\_the\_range(usrInput\_1,usrInput\_2,usrInput\_3)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 print("Provide inputs of 3 integers from the keyboard")  
 print("The 1st is the number of unique prime numbers to be created;")  
 print("The 2nd and 3rd define the range of those prime numbers.")  
 input\_values = list(map(int, input("Enter those 3 integers seperated by space :").split()))  
 main(input\_values)

A white background with black text

Description automatically generated

A white background with text

Description automatically generated

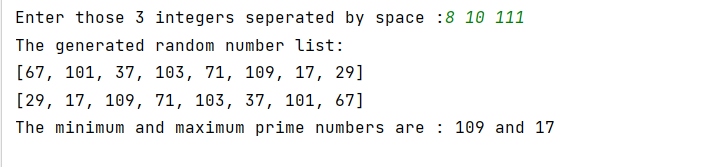
A white background with text

Description automatically generated

A group of colorful objects

Description automatically generated





A white background with black text

Description automatically generated

A white background with black text

Description automatically generated