6.Write a program to reverse a number using loop?(Get the input from user)

Sample Input:

Number: 14567

Sample Output:

Reverse Number: 76541

Test cases:

-45721

000

AD1947

!@#$%

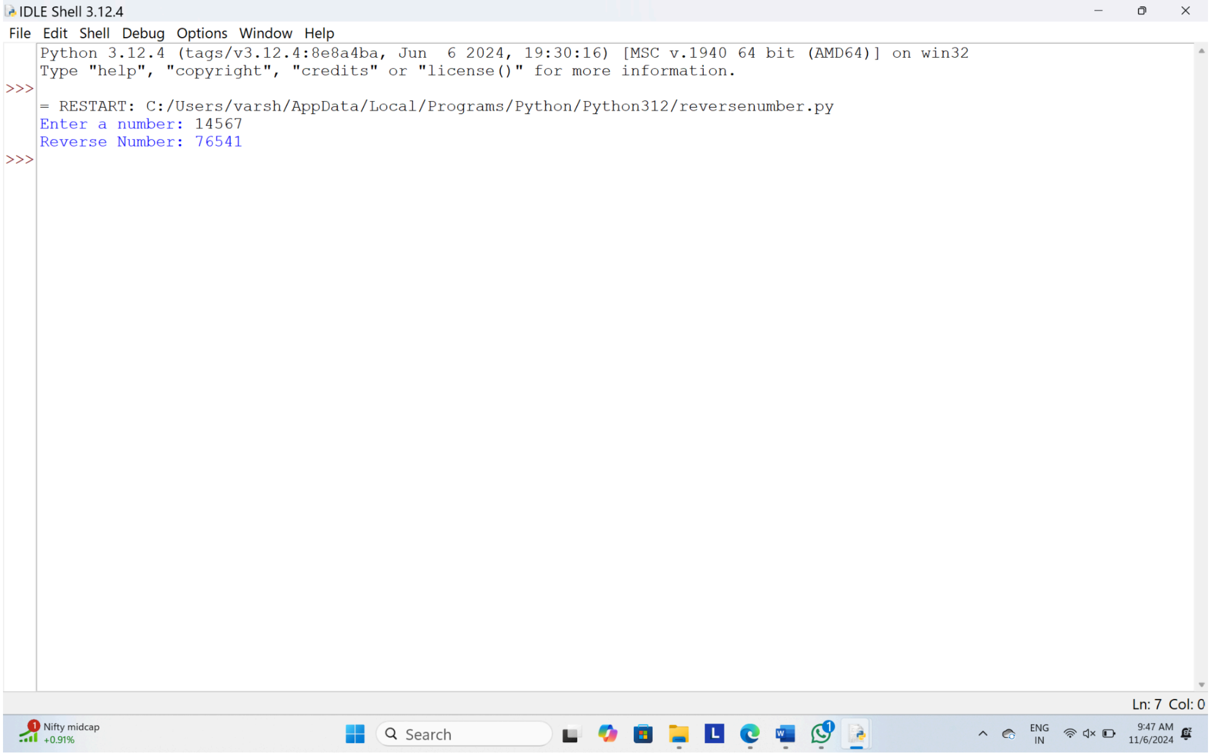
145*999=144855

PROGRAM:

```
# Function to reverse a number
def reverse_number(num_str):
        reversed_num_str = ""
        for i in range(len(num_str) - 1, -1, -1):  # Looping backward
    reversed_num_str += num_str[i]
        return reversed_num_str

# Main function to handle input and output
def main():
        num_str = input("Enter a number: ")

        # Check if the input is a valid number (allowing negative sign)
        if num_str.lstrip("-").isdigit():
    reversed_num_str = reverse_number(num_str)
    print(f"Reverse Number: {reversed_num_str}")
        else:
    print("Invalid input. Please enter a valid number.")

# Run the program
main()
```

OUTPUT:

```
Enter a number: 14567
Reverse Number: 76541
Enter a number: -45721
Reverse Number: 12754-
Enter a number: 000
Reverse Number: 000
Enter a number: AD1947
Invalid input. Please enter a valid number.
Enter a number: !@#$%
Invalid input. Please enter a valid number.
Enter a number: 145*999=144855
Invalid input. Please enter a valid number.
```



```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/varsh/AppData/Local/Programs/Python/Python312/reversenumber.py
Enter a number: 14567
Reverse Number: 76541
>>>
```

7.Write a program to find whether the person is eligible for vote or not. And if that particular person is not eligible, then print how many years are left to be eligible.

Sample Input:

Enter your age:

7

Sample output:

You are allowed to vote after 11 years

Test cases:

25

Eighteen

12

-18

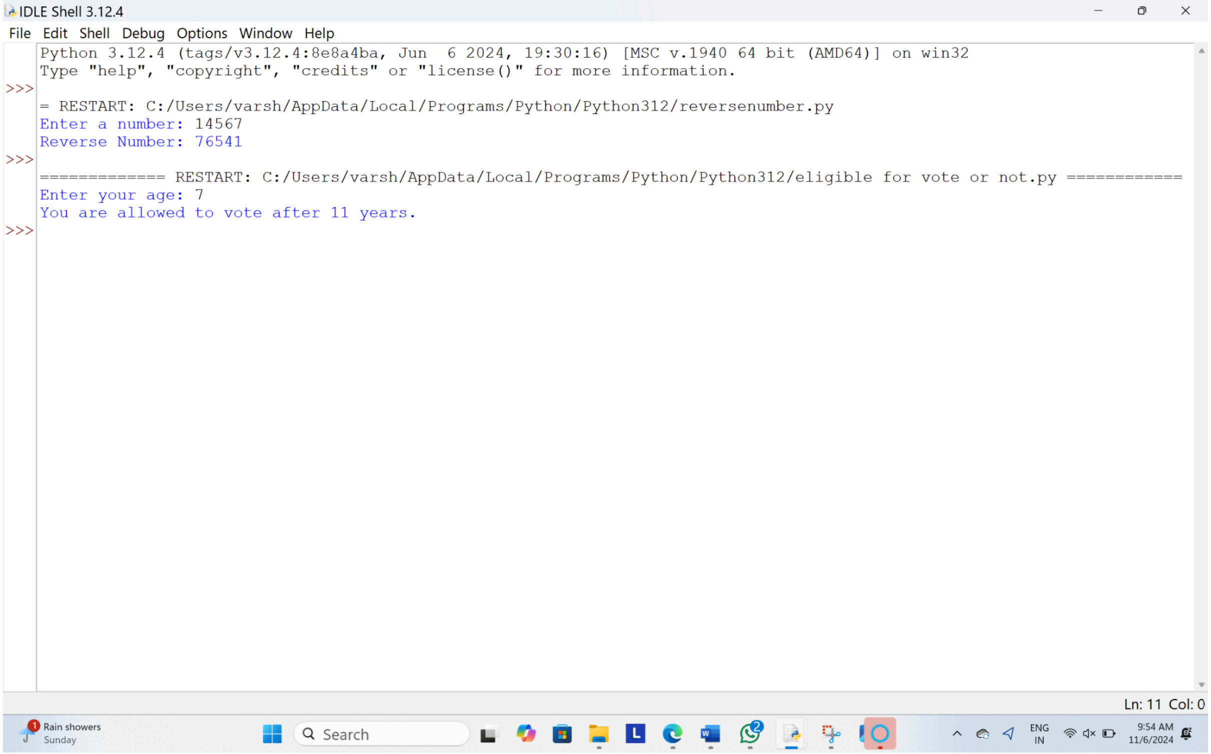
34.5

PROGRAM:

```
# Function to check voting eligibility
def check_voting_eligibility(age):
    if age < 0:
    return "Invalid input. Age cannot be negative."
    elif age >= 18:
    return "You are eligible to vote."
    else:
    years_left = 18 - age
    return f"You are allowed to vote after {years_left} years."

# Main function to handle input and output
def main():
    try:
    age = float(input("Enter your age: "))  # Accepts both integers and floats

    # Convert age to integer for further checks if it's a valid positive number
    if age.is_integer():
    age = int(age)
    result = check_voting_eligibility(age)
    else:
    result = "Invalid input. Age must be a whole number."
    except ValueError:

    result = "Invalid input. Please enter a valid number for age."


    print(result)


# Run the program

main()
```

OUTPUT:

```
Enter your age: 7

You are allowed to vote after 11 years

Enter your age: 25

You are eligible to vote.

Enter your age: Eighteen

Invalid input. Please enter a valid number for age.

Enter your age: 12

You are allowed to vote after 6 years

Enter your age: -18

Invalid input. Age cannot be negative.

Enter your age: 34.5

Invalid input. Age must be a whole number.
```



```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:/Users/varsh/AppData/Local/Programs/Python/Python312/reversenumber.py
Enter a number: 14567
Reverse Number: 76541
>>> 
============ RESTART: C:/Users/varsh/AppData/Local/Programs/Python/Python312/eligible for vote or not.py ============
Enter your age: 7
You are allowed to vote after 11 years.
>>> 
```

8.Write a program to print the total amount available in the ATM machine with the conditions applied.

Total denominations are 2000, 500, 200, 100, get the denomination priority from the user and the total number of notes from the user to display the total available balance to the user

Sample Input:

Enter the 1st Denomination: 500

Enter the 1st Denomination number of notes: 4

Enter the 2nd Denomination: 100

Enter the 2nd Denomination number of notes: 20

Enter the 3rd Denomination: 200

Enter the 3rd Denomination number of notes: 32

Enter the 4th Denomination: 2000

Enter the 4th Denomination number of notes: 1

Sample Output:

Total Available Balance in ATM: 12400

Test Cases:

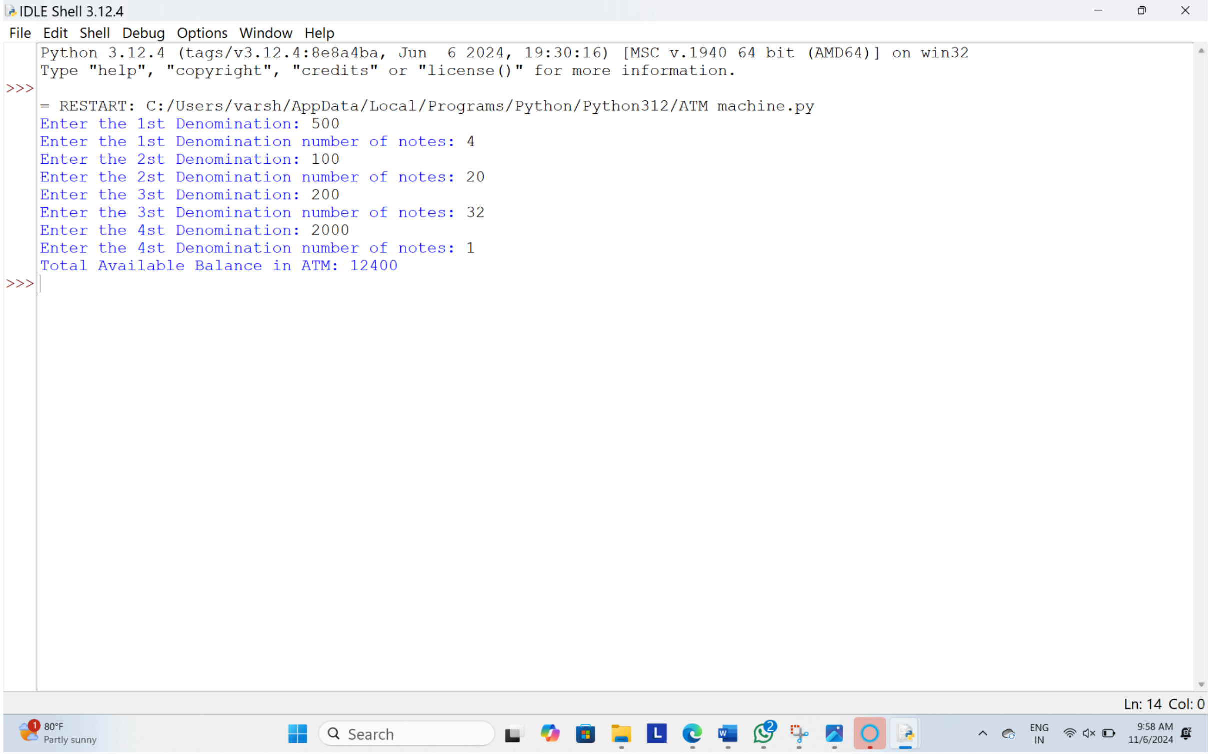
3 Hidden Test cases (Think Accordingly based on Denominations)

PROGRAM:

```
# Function to calculate total balance in ATM
def calculate_total_balance(denominations, note_counts):
        total_balance = 0
        for denomination, count in zip(denominations, note_counts):
        total_balance += denomination * count
        return total_balance

# Main function to get user input and display total balance
def main():
        denominations = []
        note_counts = []

        # Loop to get input for 4 denominations
        for i in range(4):
        denomination = int(input(f"Enter the {i + 1}st Denomination: "))
        count = int(input(f"Enter the {i + 1}st Denomination number of notes: "))

     denominations.append(denomination)
     note_counts.append(count)

        # Calculate total balance
        total_balance = calculate_total_balance(denominations, note_counts)
        # Display total balance

        print(f"Total Available Balance in ATM: {total_balance}")


# Run the program

main()
```

OUTPUT:

```
Enter the 1st Denomination: 500

Enter the 1st Denomination number of notes: 4

Enter the 2nd Denomination: 100

Enter the 2nd Denomination number of notes: 20

Enter the 3rd Denomination: 200

Enter the 3rd Denomination number of notes: 32

Enter the 4th Denomination: 2000

Enter the 4th Denomination number of notes: 1

Total Available Balance in ATM: 12400
```



```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/varsh/AppData/Local/Programs/Python/Python312/ATM machine.py
Enter the 1st Denomination: 500
Enter the 1st Denomination number of notes: 4
Enter the 2st Denomination: 100
Enter the 2st Denomination number of notes: 20
Enter the 3st Denomination: 200
Enter the 3st Denomination number of notes: 32
Enter the 4st Denomination: 2000
Enter the 4st Denomination number of notes: 1
Total Available Balance in ATM: 12400
>>>
```

9.Write a program using choice to check

Case 1: Given string is palindrome or not

Case 2: Given number is palindrome or not

Sample Input:

Case = 1

String = MADAM

Sample Output:

Palindrome

Test cases:

MONEY

5678765

MALAY12321ALAM

MALAYALAM

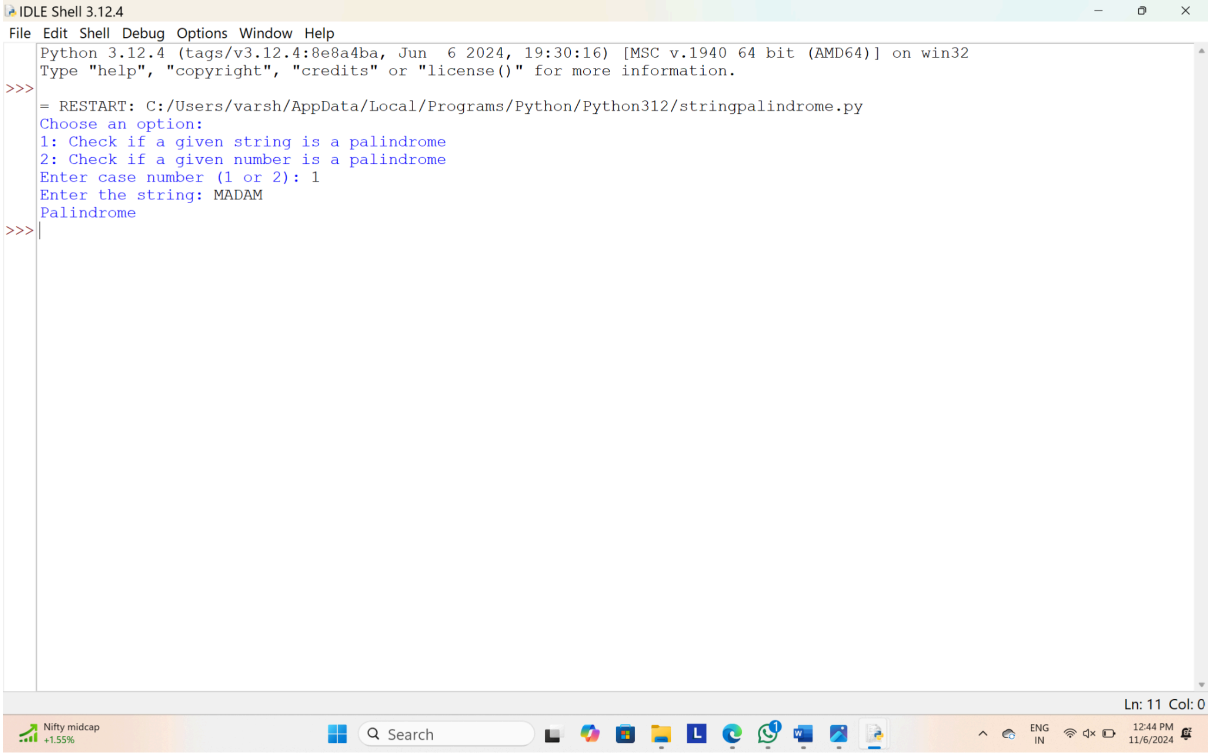
1234.4321

PROGRAM:

```
# Function to check if a string is a palindrome
def is_palindrome_string(s):
    return s == s[::-1]  # Check if string is equal to its reverse

# Function to check if a number is a palindrome
def is_palindrome_number(n):
    return str(n) == str(n)[::-1]  # Check if number is equal to its reverse string

# Main function to handle user input and choice
def main():
    print("Choose an option:")
    print("1: Check if a given string is a palindrome")
    print("2: Check if a given number is a palindrome")

    case = int(input("Enter case number (1 or 2): "))

    if case == 1:
    string_input = input("Enter the string: ")

    if is_palindrome_string(string_input):

   print("Palindrome")

    else:

   print("Not a Palindrome")


    elif case == 2:

    number_input = input("Enter the number: ")

    if is_palindrome_number(number_input):

   print("Palindrome")

    else:

   print("Not a Palindrome")


    else:

    print("Invalid case number. Please enter 1 or 2.")


# Run the program

main()
```

OUTPUT:

```
Choose an option:

1: Check if a given string is a palindrome

2: Check if a given number is a palindrome

Enter case number (1 or 2): 1

Enter the string: MADAM

Palindrome
```



```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/varsh/AppData/Local/Programs/Python/Python312/stringpalindrome.py
Choose an option:
1: Check if a given string is a palindrome
2: Check if a given number is a palindrome
Enter case number (1 or 2): 1
Enter the string: MADAM
Palindrome
>>>
```

10.Find the LCM and GCD of n numbers?

Sample Input:

N value = 2

Number 1 = 16

Number 2 = 20

Sample Output:

LCM = 80

GCD = 4

Test cases:

N = 3, {12, 25, 30}

N = 2, {52, 25, 63}

N = 3, {17, 19, 11}

N = -2, {52, 60}

N = 2, {30, 45}

PROGRAM:

```
import math
from functools import reduce

# Function to calculate GCD of two numbers
def gcd(x, y):
    return math.gcd(x, y)

# Function to calculate LCM of two numbers
def lcm(x, y):
    return abs(x * y) // gcd(x, y)

# Function to calculate GCD of a list of numbers
def gcd_multiple(numbers):
    return reduce(gcd, numbers)

# Function to calculate LCM of a list of numbers
def lcm_multiple(numbers):
    return reduce(lcm, numbers)

# Main function to handle user input and calculations
```

```
def main():
        n = int(input("Enter the number of values (N): "))


        if n <= 0:
        print("N must be a positive integer.")
        return


        numbers = []


        # Input the numbers
        for i in range(n):
        num = int(input(f"Enter Number {i + 1}: "))
      numbers.append(num)


        # Calculate GCD and LCM
        gcd_result = gcd_multiple(numbers)
        lcm_result = lcm_multiple(numbers)


        # Display results
        print(f"GCD = {gcd_result}")
        print(f"LCM = {lcm_result}")


# Run the program
main()
```

OUTPUT:

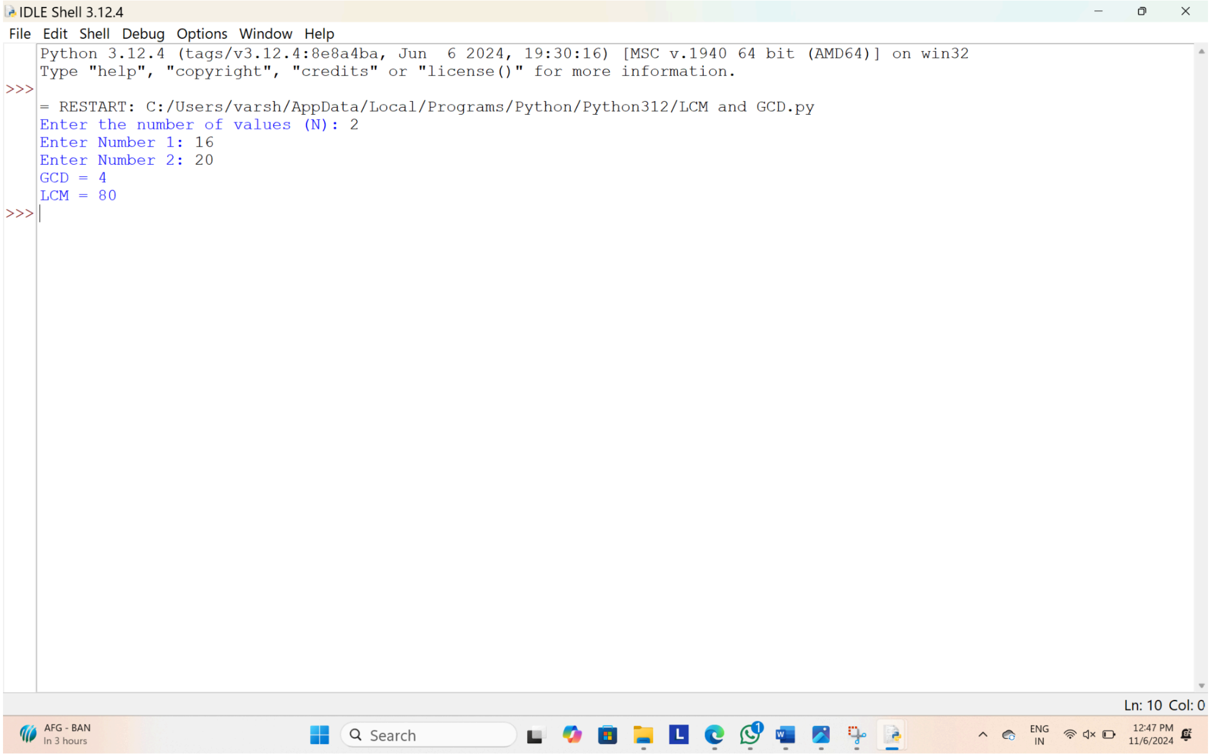
Enter the number of values (N): 2

Enter Number 1: 16

Enter Number 2: 20

GCD = 4

LCM = 80



```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:/Users/varsh/AppData/Local/Programs/Python/Python312/LCM and GCD.py
Enter the number of values (N): 2
Enter Number 1: 16
Enter Number 2: 20
GCD = 4
LCM = 80
>>> 
```