NAME = PRACHI DNYANDEV JADHAV.

P.R.N.:24030332905018

ASSIGNMENT:-01

1] Check the given number is odd or even.

In [3]:
```
def check_odd_even(number): if
    number%2==0:
        return
    "even" else:
        return "odd"
num=int(input("Enter the number:"))
print(f"The number {num} is {check_odd_even(num)}.")
```

```
Enter the number: 3
The number 3 is odd.
```

2] Count the total number of digits in a number.

In [4]:
```
def count_digit(number):
    return len(str(abs(number)))
num=int(input("Enter the number: "))
print(f"The number {num} is having {count_digit(num)} digits in it.")
```

```
Enter the number:  123465
The number 123465 is having 6 digits in it.
```

3] Write a Python program to print the reverse number pattern using a for loop 5 4 3 2 1 4 3 2 1 3 2 1 2 1 1

In [7]:
```
def reverse_number_pattern(n):
    for i in range (n, 0, -1):
        for j in range (i, 0, -1):
            print(j, end=" ")
        print()

num=int(input("enter the number:"))
reverse_number_pattern(num)
```

```
enter the number: 6
6 5 4 3 2 1
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

4]Print all prime numbers within a range.

In [6]:
```
def print_primes(start, end):
    for num in range (start, end+1):
        if num>1:

            for i in range (2, int(num**0.5)+1):
                if (num%i)==0:
                    break
                else:
                    print(num,end="")
    print()

start_range = int(input("enter the start of the range")) end_range
= int(input("enter the end of the range")) print(f"Prime numbers
between {start_range} and{end_range}") print_primes(start_range,
end_range)
```

```
enter the start of the range 2 enter
the end of the range 7

Prime numbers between 2 and7 57
```

5] Find the factorial of a given number.

In [10]:
```
def factorial(n):
    if n==0 or n==1:
        return 1
    else:
        return n * factorial(n-1)
num= int(input("enter the number:"))
print(f"The factorial of {num} is {factorial(num)}.")
```

```
enter the number: 0

The factorial of 0 is 1.
```

6] Program to check if number is palindrome.

In [17]:
```
def is_palindrome(number):
    return str(number) == str(number)[::-1]

num= int(input("enter the number:"))
if is_palindrome(num):
    print(f"the number {num} is palindrome")
else:
    print(f"the number {num} is not palindrome")
```

```
enter the number: 121

the number 121 is palindrome
```

7] Program to Check Armstrong Number.

In [18]:
```
def  is_armstrong(number):
    num_str = str(number)
    power = len(num_str)
    total = sum(int(digit) ** power for digit in num_str)
```

```
    return total == number


num = int(input("Enter a number: "))
if is_armstrong(num):
    print(f"The number {num} is an Armstrong number.")
else:
    print(f"The number {num} is not an Armstrong number.")
```

```
Enter a number:  153

The number 153 is an Armstrong number.
```

8] Find Maximum of three numbers.

In [21]:
```
def find_maximum(a, b, c): return
    max(a, b, c)


num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: ")) num3
= int(input("Enter third number: "))
print(f"The maximum number among {num1}, {num2}, and {num3} is {find_maximu
```

```
Enter first number:  1
Enter second number:  6
Enter third number: 4

The maximum number among 1, 6, and 4 is 6.
```

9] Find the Sum of digits.

In [22]:
```
def sum_of_digits(number):
    return sum(int(digit) for digit in str(abs(number)))


num = int(input("Enter a number: "))
print(f"The sum of digits in {num} is {sum_of_digits(num)}.")
```

```
Enter a number:  12312

The sum of digits in 12312 is 9.
```

10] Python Program to Print the Natural Numbers Summation Pattern Given a natural number n, the task is to write a Python program to first find the sum of first n natural numbers and then print each step as a pattern. Input: 5 Output: 1 = 1 1 + 2 = 3 1 + 2 + 3 = 6 1 + 2 + 3 + 4 = 10 1 + 2 + 3 + 4 + 5 = 15

In [23]:
```
def summation_pattern(n):
    total = 0
    for i in range(1, n + 1):
        total += i
        print(" + ".join(map(str, range(1, i + 1))), "=", total)


num = int(input("Enter a natural number: "))
summation_pattern(num)
```

Enter a natural number: 5

```
1 = 1
1 + 2 = 3
1 + 2 + 3 = 6
1 + 2 + 3 + 4 = 10
1 + 2 + 3 + 4 + 5 = 15
```

In [ ]