Medha Nikalesh Bapu

24030331245030

1. CHECK THE GIVEN NUMBER IS ODD OR EVEN

In [12]:
```
def check_odd_or_even(number):
      if number%2==0:
            return "even"
         elif number%2!=0:
           return "odd"


num=int(input("Enter the number:"))
print(f"the number {num} is {check_odd_or_even(num)}.")
```

Enter the number: 3

the number 3 is odd.

2. Count the total number of digits in a number

In [3]:
```
def count_digits(number):
      number = abs(number)
      return len(str(number))

num = int(input("Enter a number: "))
digit_count = count_digits(num)
print(f"Total number of digits: {digit_count}")
```

Enter a number: 2345

Total number of digits: 4

3. Write a Python program to print the reverse number pattern using a for loop

In [4]:
```
def reverse_number_pattern(rows):
      for  i  in  range(rows, 0, -1):
           for  j  in range(1, i + 1):
                print(j, end=" ")
           print()


#Getthenumber of rows from the user
rows  =  int(input("Enter the number of rows: "))

# Callthefunction to print the pattern
reverse_number_pattern(rows)
```

Enter the number of rows: 5

1 2 3 4 5
1 2 3 4
1 2 3

```
1 2
1
```

4. Print all prime numbers within a range

In [5]:

```
def is_prime(num):
    if num <= 1:
        return False
    if num <= 3:
        return True
    if num % 2 == 0 or num % 3 == 0:
        return False
    i = 5
    while i * i <= num:
        if num % i == 0 or num % (i + 2) == 0:
            return False
        i += 6
    return True

def print_primes_in_range(start, end):
    for num in range(start, end + 1):
        if is_prime(num):
            print(num)

start_range = int(input("Enter the starting number of the range: "))
end_range = int(input("Enter the ending number of the range: "))

print("Prime numbers in the range", start_range, "to", end_range, "are:")
print_primes_in_range(start_range, end_range)
```

```
Enter the starting number of the range: 1 Enter
the ending number of the range: 30
Prime numbers in the range 1 to 30 are:
2
3
5
7
11
13
17
19
23
29
```

5. Find the factorial of a given number

In [6]:

```
def factorial(n):
    if n < 0:
        return "Factorial is not defined for negative numbers"
    elif n == 0:
        return 1
    else:
        fact = 1
        for i in range(1, n + 1):
            fact = fact * i
        return fact

number = int(input("Enter a number: "))
```

```
result = factorial(number)
print(f"The factorial of {factorial(number)}")
```

Enter a number: 1

The factorial of 1

6. Program to check if number is palindrome

In [7]:
```
def is_palindrome(n):
    return str(n) == ''.join(reversed(str(n)))

# Get the number from the user
n = int(input("Enter number: "))

# Check if the number is a palindrome
if is_palindrome(n):
   print("The number is a palindrome!")
else:
   print("The number is not a palindrome.")
```

Enter number: 5

The number is a palindrome!

7. Program to Check Armstrong Number

153 is an Armstrong number because:

1^3+5^3+3^3=1+125+27=153

In [8]:
```
# Ask the user to enter a number
num = int(input("Enter a number: "))

# Convert the number to a string to count the digits
num_str = str(num)
num_digits = len(num_str)

# Initialize sum to 0
sum_of_powers = 0

# Loop through each digit in the number
for digit in num_str:
    sum_of_powers += int(digit) ** num_digits          # Raise to the power and add

# Check if the sum of powers equals the original number
if sum_of_powers == num:
    print(f"{num} is an Armstrong number.")
else:
    print(f"{num} is not an Armstrong number.")
```

Enter a number: 153

153 is an Armstrong number.

8. Find Maximum of three numbers

In [10]:

```
# Ask the user to enter the first number
num1 = float(input("Enter the first number: "))

# Ask the user to enter the second number
num2 = float(input("Enter the second number: "))

# Ask the user to enter the third number
num3 = float(input("Enter the third number: "))

# Compare the three numbers using if-elif-else
if (num1 >= num2) and (num1 >= num3):
    maximum = num1
elif (num2 >= num1) and (num2 >= num3):
    maximum = num2
else:
    maximum = num3

# Print the maximum number
print("The maximum number is:", maximum)
```

Enter the first number: 7 Enter the second number: 45 Enter the third number: 33

The maximum number is: 45.0

9. Find the Sum of digits

In [11]:

```
# Ask the user to enter a number
num = int(input("Enter a number: "))

# Make sure the number is positive
num = abs(num)

# Initialize a variable to store the sum of digits
sum_of_digits = 0

# Loop until the number becomes 0
while num > 0:
    digit = num % 10            # Get the last digit
    sum_of_digits += digit      # Add the digit to the sum
    num = num // 10             # Remove the last digit

# Print the result
print("Sum of digits is:", sum_of_digits)
```

Enter a number: 4444

Sum of digits is: 16

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.

In [12]:

```
# Ask the user to enter a natural number
n = int(input("Enter a natural number: "))
```

```python
# Initialize the sum variable
total_sum = 0

# Loop from 1 to n
for i in range(1, n + 1):
    total_sum += i      # Add current number to total sum

    # Create the pattern string "1 + 2 + ... + i"
    pattern = " + ".join(str(x) for x in range(1, i + 1))

    # Print the pattern along with the current sum
    print(f"{pattern} = {total_sum}")
```

Enter a natural number: 10

1 = 1
1 + 2 = 3
1 + 2 + 3 = 6
1 + 2 + 3 + 4 = 10
1 + 2 + 3 + 4 + 5 = 15
1 + 2 + 3 + 4 + 5 + 6 = 21
1 + 2 + 3 + 4 + 5 + 6 + 7 = 28
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 = 45
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55