

-
-
1. Write a Python program to find the Nth Fibonacci number. The program should take the value of N as input

n=8

Output 0 1 1 2 3 5 8 13

```
N = int(input("Enter the value of N to find the Nth Fibonacci number: "))
```

```
if N <= 0:
```

```
    print("Invalid input. Please enter a positive integer.")
```

```
else:
```

```
    fib1, fib2 = 0, 1
```

```
    for i in range(2, N):
```

```
        fib_sum = fib1 + fib2
```

```
        fib1, fib2 = fib2, fib_sum
```

```
    print(f"The {N}th Fibonacci number is:", fib2 if N > 1 else fib1)
```

-
-
2. Write a python program for sum of squares of first N natural numbers. Given a positive integer N, the task is to find $1^2 + 2^2 + 3^2 + \dots + N^2$

```
def sum_of_squares(n):
```

```
    return (n * (n + 1) * (2 * n + 1)) // 6
```

```
n = int(input("Enter a positive integer: "))
```

```
print("Sum of squares of the first", n, "natural numbers:", sum_of_squares(n))
```

-
-
3. Write a Python program that accepts a string and calculates the number of digits and letters.

Sample Data : Python 3.2

Expected Output :

Letters 6

Digits 2

```
def count_letters_digits(string):
```

```
    num_letters = 0
```

```
    num_digits = 0
```

```
    for char in string:
```

```
        if char.isalpha():
```

```
            num_letters += 1
```

```

        elif char.isdigit():
            num_digits += 1
    return num_letters, num_digits

n = input("Enter a string: ")
letters, digits = count_letters_digits(n)
print("Letters:", letters)
print("Digits:", digits)

```

-
4. Write a Python program to reverse the order of the items in the array.

Sample Output

Original array: array('i', [1, 3, 5, 3, 7, 1, 9, 3])

Reverse the order of the items:

array('i', [3, 9, 1, 7, 3, 5, 3, 1])

```

import array as arr
arr = arr.array('i', [1, 3, 5, 3, 7, 1, 9, 3])
print("Original array:", arr)
arr.reverse()
print("Reversed array:", arr)

```

-
5. Write a python program to merge 2 lists and also print the list in the Ascending order using built in function

input:

a=[1,2,7,4,5]

b=[11,10,13]

Output:

c=[1,2,7,4,5,11,10,13]

Z= [1,2,4,5,7,10,11,13]

```

# Initialize the lists

```

```

a = [1, 2, 7, 4, 5]

```

```

b = [11, 10, 13]

```

```

# Merge the lists

```

```

c = a + b

```

```

# Sort the merged list in ascending order

```

```
Z = sorted(c)
```

```
# Print the merged and sorted list
```

```
print("Merged list:", c)
```

```
print("Sorted list:", Z)
```

-
6. Write a Python program to count the number of even and odd numbers in a series of numbers.

Sample numbers : numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9)

Expected Output :

Number of even numbers : 5

Number of odd numbers : 4

```
def count_even_odd(numbers):
    even_count = sum(num % 2 == 0 for num in numbers)
    odd_count = len(numbers) - even_count
    return even_count, odd_count
numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9)
even_numbers, odd_numbers = count_even_odd(numbers)
print("Number of even numbers:", even_numbers)
print("Number of odd numbers:", odd_numbers)
```

-
7. Write a Python program that checks if a specific substring is present in a given string. The program should print "Substring found" if the substring is present, otherwise print "Substring not found"
- Sample Input: python is Integrated Language
- Substring : Language
- Output: Substring found

```
string = "python is Integrated Language"
substring = "language"
string = string.lower()
substring = substring.lower()
if substring in string:
    print("Substring found")
else:
    print("Substring not found")
```

8. Write a program to find the Factorial of a number using recursion.

```
def factorial(n):
    if n == 0 :
        return 1
    else:
        return n * factorial(n-1)
num = int(input("Enter a number: "))
print("factorial=",factorial(num))
```

9. Write a program to check if a number is a prime number or not using recursion.

```
def is_prime_recursive(number, divisor=2):
    if number < 2:
        return False
    if divisor * divisor > number:
        return True
    if number % divisor == 0:
        return False
    return is_prime_recursive(number, divisor + 1)
num = int(input("Enter a number: "))
if is_prime_recursive(num):
    print(f'{num} is a prime number.')
else:
    print(f'{num} is not a prime number.')
```

(OR)

```
def prime(n,i):
    if i==1:
        return 1
    if n%i==0:
        return 0
    return prime(n,i-1)
n=int(input("enter a num "))
x=prime(n,n-1)

if x==1:
```

```
    print("prime number ")
if x==0:
    print("not prime number ")
```

prime number

=====