



main.py



Save

Run

Output

Clear

```
1  #Write a program to check a string is palidrome or not
2  from collections import Counter
3
4  def count_occurrences(lst):
5      # Using Counter to count occurrences of elements in the list
6      occurrences = Counter(lst)
7
8      # Printing the occurrences
9      for element, count in occurrences.items():
10         print(f"{element}: {count}")
11
12 # Example list
13 my_list = [1, 2, 3, 4, 1, 2, 3, 1, 2, 1]
14
15 # Counting occurrences
16 count_occurrences(my_list)
17
```

```
1: 4
2: 3
3: 2
4: 1
> |
```



JS





main.py



Save

Run

Output

Clear

```
1 #Write a program to counr occurance of each element in the list
2 from collections import Counter
3
4 def count_occurrences(lst):
5     # Using Counter to count occurrences of elements in the list
6     occurrences = Counter(lst)
7
8     # Printing the occurrences
9     for element, count in occurrences.items():
10         print(f"{element}: {count}")
11
12 # Example list
13 my_list = [1, 2, 3, 4, 1, 2, 3, 1, 2, 1]
14
15 # Counting occurrences
16 count_occurrences(my_list)
17
```

```
1: 4
2: 3
3: 2
4: 1
>
```



JS





main.py



Save

Run

Output

Clear

```
1  #Write a program to find mean,median,variance and deviation of the list using numpy
2  import numpy as np
3
4  def calculate_statistics(lst):
5      # Convert the list to a numpy array
6      arr = np.array(lst)
7
8      # Calculate mean
9      mean = np.mean(arr)
10
11     # Calculate median
12     median = np.median(arr)
13
14     # Calculate variance
15     variance = np.var(arr)
16
17     # Calculate standard deviation
18     std_dev = np.std(arr)
19
20     return mean, median, variance, std_dev
21
22 # Example list
23 my_list = [1, 2, 3, 4, 5]
24
25 # Calculate statistics
26 mean, median, variance, std_dev = calculate_statistics(my_list)
27
28 # Print the results
29 print("Mean:", mean)
30 print("Median:", median)
31 print("Variance:", variance)
32 print("Standard Deviation:", std_dev)
33
```

```
Mean: 3.0
Median: 3.0
Variance: 2.0
Standard Deviation: 1.4142135623730951
> |
```



main.py



Save

Run

Output

Clear

```
1  #Write a program to read data from CSV file and display first five rows and last five rows
2  def fibonacci(n):
3      # Initialize variables to store the first two terms of the sequence
4      a, b = 0, 1
5
6      # Initialize a list to store the Fibonacci sequence
7      fibonacci_sequence = []
8
9      # Check if n is less than or equal to 0
10     if n <= 0:
11         print("Please enter a positive integer.")
12     else:
13         count = 0
14         # Generate the Fibonacci sequence using a while loop
15         while count < n:
16             fibonacci_sequence.append(a)
17             nth_term = a + b
18             # Update the values of a and b for the next iteration
19             a = b
20             b = nth_term
21             count += 1
22     return fibonacci_sequence
23
24 # Example: Generating Fibonacci sequence up to 10 terms
25 n = 78
26 fib_sequence = fibonacci(n)
27 print(f"Fibonacci sequence up to {n} terms:", fib_sequence)
28
```

Fibonacci sequence up to 78 terms: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229, 832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352, 24157817, 39088169, 63245986, 102334155, 165580141, 267914296, 433494437, 701408733, 1134903170, 1836311903, 2971215073, 4807526976, 7778742049, 12586269025, 20365011074, 32951280099, 53316291173, 86267571272, 139583862445, 225851433717, 365435296162, 591286729879, 956722026041, 1548008755920, 2504730781961, 4052739537881, 6557470319842, 10610209857723, 17167680177565, 27777890035288, 44945570212853, 72723460248141, 117669030460994, 190392490709135, 308061521170129, 498454011879264, 806515533049393, 1304969544928657, 2111485077978050, 3416454622906707, 5527939700884757]

> |



main.py



Save

Run

Output

Clear

```
1  # Python3 program to display Prime numbers till N
2
3  #function to check if a given number is prime
4  def isPrime(n):
5      #since 0 and 1 is not prime return false.
6      if(n==1 or n==0): return False
7
8      #Run a loop from 2 to n-1
9      for i in range(2,n):
10         #if the number is divisible by i, then n is not a prime number.
11         if(n%i==0):
12             return False
13
14     #otherwise, n is prime number.
15     return True
16
17
18
19 # Driver code
20 N = 500
21 #check for every number from 1 to N
22 for i in range(1,N+1):
23     #check if current number is prime
24     if(isPrime(i)):
25         print(i,end=" ")
26
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139
149 151 157 163 167 173 179 181 191 193 197 199 211 223 227 229 233 239 241 251 257 263 269 271 277 281
283 293 307 311 313 317 331 337 347 349 353 359 367 373 379 383 389 397 401 409 419 421 431 433 439 443
449 457 461 463 467 479 487 491 499 > |
```