



main.py

```
1 program to rotate the list to given index.Eg: list['a', 'b', 'c', 'd', 'e']Index = 2rotated
2 = ['a', 'b', 'c', 'd', 'e']
3 _index = 2
4
5 ion_index >= len(my_list):
6 t("Index out of range")
7
8 ted_list = my_list[rotation_index:] + my_list[:rotation_index]
9 t(rotated_list)
10
```

input

```
['c', 'd', 'e', 'a', 'b']

...Program finished with exit code 0
Press ENTER to exit console.
```



main.py

```
1 # A list contains both positive and negative numbers. sort thelist by ignoring sign.Actual l
2 my_list = [9, -4, 2, 0, -2, 7, -3, 5, -5]
3 sorted_list = sorted(my_list, key=lambda x: abs(x))
4 print(sorted_list)
5
```

input

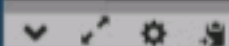
```
[0, 2, -2, -3, -4, 5, -5, 7, 9]

...Program finished with exit code 0
Press ENTER to exit console.
```



main.py

```
1 # Write a function that take a sentence and returns the sentence by shuffling the words.
2 import random
3
4 def shuffle_sentence(sentence):
5     words = sentence.split()
6     random.shuffle(words)
7     shuffled_sentence = ' '.join(words)
8     return shuffled_sentence
9 sentence = "This is a sample sentence to demonstrate shuffling words."
10 shuffled_sentence = shuffle_sentence(sentence)
11 print(shuffled_sentence)
12
```



input

a to shuffling is This demonstrate sample words. sentence

...Program finished with exit code 0
Press ENTER to exit console.



main.py

```
1 # Write a function to return LCM of two numbers
2 def gcd(a, b):
3     while b:
4         a, b = b, a % b
5     return a
6
7 def lcm(a, b):
8     return (a * b) // gcd(a, b)
9
10 num1 = 1245
11 num2 = 1807
12 result = lcm(num1, num2)
13 print("LCM of", num1, "and", num2, "is:", result)
14
```

⌵ ↶ ⚙ 📄

input

LCM of 1245 and 1807 is: 2249715

...Program finished with exit code 0
Press ENTER to exit console.



main.py

```
1 # A list contains words. Write a program to delete the words form list if it contains more
2 def count_vowels(word):
3     vowels = "aeiouAEIOU"
4     count = 0
5     for char in word:
6         if char in vowels:
7             count += 1
8     return count
9 def filter_words(words):
10     filtered_words = [word for word in words if count_vowels(word) <= 2]
11     return filtered_words
12 word_list = ["apple", "banana", "orange", "watermelon", "kiwi"]
13 filtered_list = filter_words(word_list)
14 print(filtered_list)
15
```



input

```
['apple', 'kiwi']
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```



main.py

```
1 #Write a function to take a n-digits number and check whether it is dominated by even numbe
2 def digit_dominance(number):
3     even_count = 0
4     odd_count = 0
5
6     while number > 0:
7         digit = number % 10
8         if digit % 2 == 0:
9             even_count += 1
10        else:
11            odd_count += 1
12        number //= 10
13
14    if even_count > odd_count:
15        return "Dominated by even numbers"
16    elif odd_count > even_count:
17        return "Dominated by odd numbers"
18    else:
19        return "Balanced"
20
21
22 number = 1234
23 result = digit_dominance(number)
24 print(f"{number} is {result}")
25
26 number = 2234
27 result = digit_dominance(number)
28 print(f"{number} is {result}")
29
30 number = 1111
31 result = digit_dominance(number)
32 print(f"{number} is {result}")
33
```

input

```
1234 is Balanced
2234 is Dominated by even numbers
1111 is Dominated by odd numbers
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```



Run

Debug

Stop

Share

Save

{ } Beautify



Language Python 3



main.py

```
1 #write a program to convert a column matrix to row matrixcol_matrix: [1, 2, 3, 4]row_matrix:
2 def column_to_row(column_matrix):
3     row_matrix = [[element] for element in column_matrix]
4     return row_matrix
5 column_matrix = [1, 2, 3, 4]
6 row_matrix = column_to_row(column_matrix)
7 print(row_matrix)
8
```



input

```
[[1], [2], [3], [4]]
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```



main.py

```
1 # write a program to calculate sum of prime numbers within agiven range.
2 def is_prime(num):
3     if num <= 1:
4         return False
5     for i in range(2, int(num**0.5) + 1):
6         if num % i == 0:
7             return False
8     return True
9
10 def sum_of_primes_in_range(start, end):
11     prime_sum = 0
12     for num in range(start, end + 1):
13         if is_prime(num):
14             prime_sum += num
15     return prime_sum
16
17 start_range = 10
18 end_range = 50
19 result = sum_of_primes_in_range(start_range, end_range)
20 print("Sum of prime numbers between", start_range, "and", end_range, "is:", result)
21
```

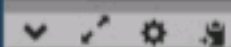
Sum of prime numbers between 10 and 50 is: 311

...Program finished with exit code 0
Press ENTER to exit console.



main.py

```
1 # write a program to add tuple to list
2 my_list = [(1, 2), (3, 4), (5, 6)]
3 new_tuple = (7, 8)
4
5 my_list.append(new_tuple)
6
7 print(my_list)
8
```



input

```
[(1, 2), (3, 4), (5, 6), (7, 8)]
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```



main.py

```
1 # write a program to import math module and use factorial function gcd function calculate area of circle
2 def factorial(n):
3     result = 1
4     for i in range(1, n + 1):
5         result *= i
6     return result
7
8 def gcd(a, b):
9     while b:
10         a, b = b, a % b
11     return a
12
13 def area_of_circle(radius):
14     return 3.14 * radius**2
15
16 num = 5
17 factorial_result = factorial(num)
18 print("Factorial of", num, "is:", factorial_result)
19
20 num1 = 36
21 num2 = 48
22 gcd_result = gcd(num1, num2)
23 print("GCD of", num1, "and", num2, "is:", gcd_result)
24
25 radius = 5
26 area = area_of_circle(radius)
27 print("Area of circle with radius", radius, "is:", area)
28
```

Factorial of 5 is: 120
GCD of 36 and 48 is: 12
Area of circle with radius 5 is: 78.5

...Program finished with exit code 0
Press ENTER to exit console.[]