



main.py

```
1 # Write a program to calculate the sum of list by index_range taken from console.
2 numbers = [int(num) for num in input("Enter a list of numbers separated by spaces: ").split()]
3 start = int(input("Enter the start index of the range: "))
4 end = int(input("Enter the end index of the range: "))
5
6 if start < 0 or end >= len(numbers):
7     print("Index out of range")
8 else:
9     total = sum(numbers[start:end + 1])
10    print("Sum within the specified range:", total)
11
```

input

```
Enter a list of numbers separated by spaces: 46 87 40 356 346 3565 346
Enter the start index of the range: 0
Enter the end index of the range: 6
Sum within the specified range: 4786
```

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



main.py

```
1 # Write a function to return whether an 8-bit binary string has equal no of zeros and ones
2 def check_balance(binary_string):
3     count_ones = binary_string.count('1')
4     count_zeros = binary_string.count('0')
5     return count_ones == count_zeros
6 binary_string = "0111100"
7 result = check_balance(binary_string)
8 if result:
9     print("The binary string is balanced.")
10 else:
11     print("The binary string is unbalanced.")
12
```

input

The binary string is unbalanced.

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





main.py

```
1 # Write a function to take two values as parameters and return ratio of two values.
2 def calculate_ratio(value1, value2):
3     if value2 == 0:
4         return "Cannot divide by zero"
5     return value1 / value2
6 value1 = 10
7 value2 = 5
8 ratio = calculate_ratio(value1, value2)
9 print("The ratio of", value1, "to", value2, "is:", ratio)
10
```

input

```
The ratio of 10 to 5 is: 2.0

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



main.py

```
1 # A dictionary of a product contains brands as keys () and list of MRP and discount percent
2 def calculate_final_price(product_dict):
3     final_price_dict = {}
4     for brand, details in product_dict.items():
5         mrp, discount_percentage = details
6         final_price = mrp * (1 - discount_percentage / 100)
7         final_price_dict[brand] = final_price
8     return final_price_dict
9 product = {'brand1': [100, 10], 'brand2': [200, 20], 'brand3': [150, 15]}
10 final_price_dict = calculate_final_price(product)
11 print("Final Price Dictionary:", final_price_dict)
12
```



main.py

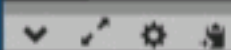
```
1 # Write a program to make class user:That takes user-id, mobile-no and Gmail-id to construc
2 class User:
3     def __init__(self, user_id, mobile_no, gmail_id):
4         self.user_id = user_id
5         self.mobile_no = mobile_no
6         self.gmail_id = gmail_id
7
8     def display_details(self):
9         masked_mobile = "*****" + str(self.mobile_no)[-2:]
10        index_at = self.gmail_id.index('@')
11        masked_gmail = "*****" + self.gmail_id[:index_at]
12
13        print("User-id:", self.user_id)
14        print("Mobile no:", masked_mobile)
15        print("Gmail id:", masked_gmail + "@gmail.com")
16
17 user1 = User("admin123", "123456789", "example123@gmail.com")
18 user1.display_details()
19
```

User-id: admin123  
Mobile no: \*\*\*\*\*89  
Gmail id: \*\*\*\*\*example123@gmail.com  
  
...Program finished with exit code 0  
Press ENTER to exit console.



main.py

```
1 # Write a function to return the list of capital letters in a string.Hint:by checking ASCII
2 def find_capital_letters(string):
3     return [char for char in string if char.isupper()]
4 input_string = "Hello World!"
5 result = find_capital_letters(input_string)
6 print("Capital letters in the string:", result)
7
```



Input

Capital letters in the string: ['H', 'W']

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

OnlineGDB

beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom 

new

Learn Programming

Programming Questions

Jobs 

new

Sign Up

Login

Learn Python with KodeKloud

Run

Debug

Stop

Share

Save

{ } Beautify

main.py

1

#write a program to display k column of 3\*3 matrix.

2

matrix = [[1, 2, 3],

3

[4, 5, 6],

4

[7, 8, 9]]

5

k = 1

6

for i in range(3):

7

print(matrix[i][k])

8

input

2

5

8

...Program finished with exit code 0

Press ENTER to exit console.



main.py

```
1 # write a program to take list of integers and convert their signs i.e. positive to negative
2 integers = input("Enter a list of integers separated by spaces: ").split()
3 integers = [int(x) for x in integers]
4 converted_integers = [-x for x in integers]
5 print("Converted list with signs reversed:", converted_integers)
6
```

input

Enter a list of integers separated by spaces: 24 80 -85 -234 8642  
Converted list with signs reversed: [-24, -80, 85, 234, -8642]

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





main.py

```
1 # Write a function to take shape of two matrix and check whether a matrix multiplication is possible
2 def check_matrix_multiplication(shape1, shape2):
3     columns_shape1 = shape1[1] if len(shape1) == 2 else shape1[0]
4     rows_shape2 = shape2[0] if len(shape2) == 2 else 1
5     if columns_shape1 == rows_shape2:
6         return True
7     else:
8         return False
9 shape1 = (3, 4)
10 shape2 = (4, 2, 5)
11
12 if check_matrix_multiplication(shape1, shape2):
13     print("Matrix multiplication is possible.")
14 else:
15     print("Matrix multiplication is not possible.")
16
```

Matrix multiplication is not possible.

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