

main.py

```
1 def login():
2     default_credentials = {'user123': 'password123'}
3
4     user_id = input("Enter user ID: ")
5     password = input("Enter password: ")
6
7     if user_id in default_credentials and default_credentials[user_id] == password:
8         print("Home-page")
9     else:
10        raise ValueError("Invalid-Credentials")
11
12 try:
13     login()
14 except ValueError as e:
15     print(e)
16
17
```



input

```
Enter user ID: 65
Enter password: 2003
Invalid-Credentials
```

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



main.py

```
1 class Ticket:
2     def __init__(self, **passenger_details):
3         self.passenger_details = passenger_details
4
5     def display_passenger_details(self):
6         for person, age in self.passenger_details.items():
7             print(f"{person}: Age {age}")
8
9     def calculate_total_fare(self):
10        total_fare = 0
11        for person, age in self.passenger_details.items():
12            fare = 100 # Assuming fare for each person is $100
13            if age >= 65: # Senior citizen concession
14                fare -= fare * 0.1 # 10% concession
15            total_fare += fare
16        return total_fare
17
18    def display_ticket_details(self):
19        print("Passenger Details:")
20        self.display_passenger_details()
21        total_fare = self.calculate_total_fare()
22        print(f"Total Fare: ${total_fare:.2f}")
23        grand_total = total_fare
24        print(f"Grand Total: ${grand_total:.2f}")
25
26
27 # Example usage
28 ticket = Ticket(person_A=21, person_B=24, person_C=65)
29 ticket.display_ticket_details()
```

input

```
Passenger Details:
person_A: Age 21
person_B: Age 24
person_C: Age 65
Total Fare: $290.00
Grand Total: $290.00
```

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



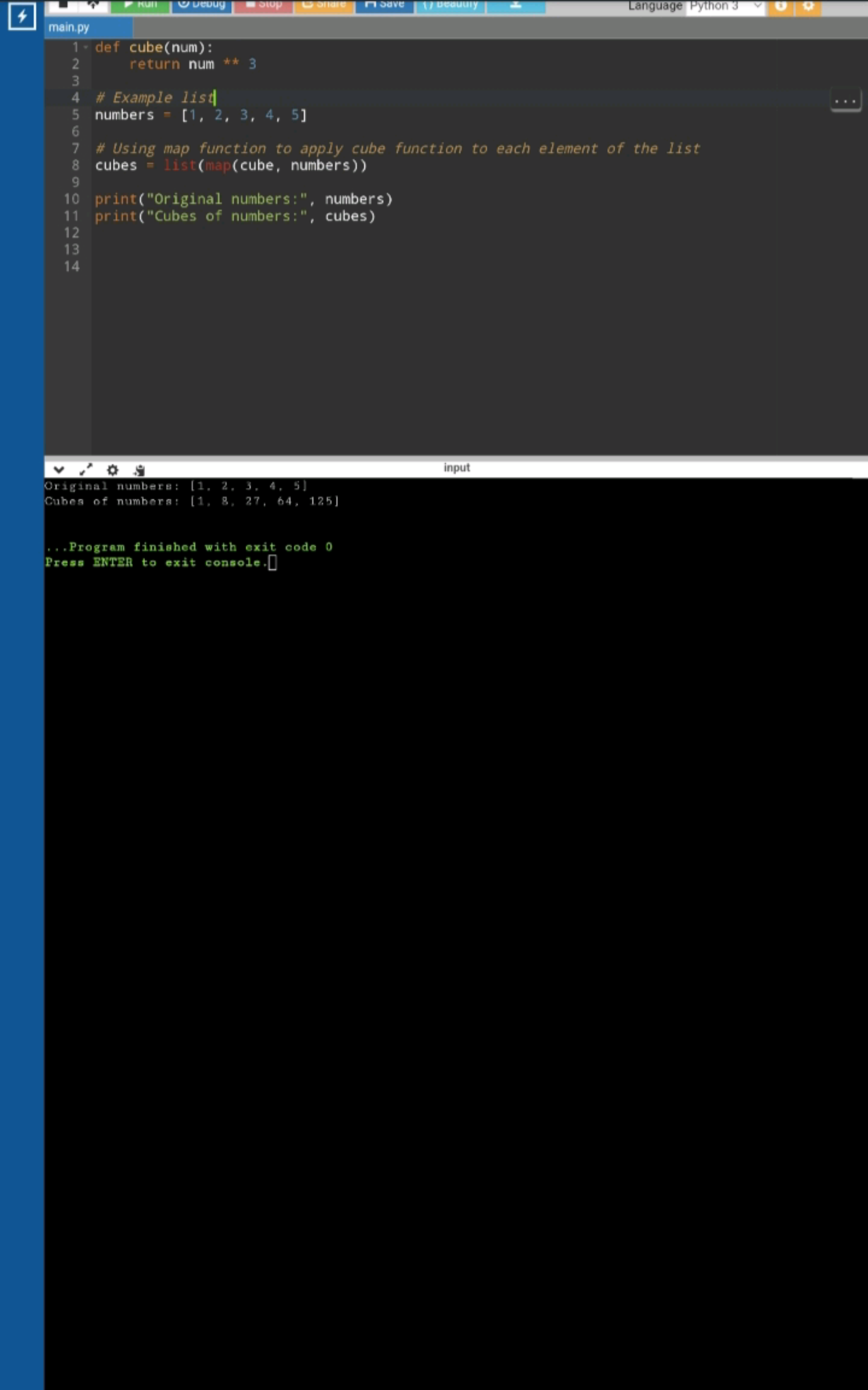
main.py

```
1 # Initialize an empty list to store elements
2 elements = []
3
4 # Take input from the user until 'stop' is entered
5 while True:
6     element = input("Enter an element to add to the list (or 'stop' to end): ")
7     if element.lower() == 'stop':
8         break
9     else:
10        elements.append(element)
11
12 # Print the final list
13 print("List of elements:", elements)
14
15
```



input

```
Enter an element to add to the list (or 'stop' to end): 65
Enter an element to add to the list (or 'stop' to end): 56
Enter an element to add to the list (or 'stop' to end): 45
Enter an element to add to the list (or 'stop' to end): 23
Enter an element to add to the list (or
'stop' to end): 45
```



main.py

```
1 def cube(num):
2     return num ** 3
3
4 # Example list
5 numbers = [1, 2, 3, 4, 5]
6
7 # Using map function to apply cube function to each element of the list
8 cubes = list(map(cube, numbers))
9
10 print("Original numbers:", numbers)
11 print("Cubes of numbers:", cubes)
12
13
14
```

input

Original numbers: [1, 2, 3, 4, 5]
Cubes of numbers: [1, 8, 27, 64, 125]

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

```
1 def remove_vowel_starting_strings(input_list):
2     vowels = {'a', 'e', 'i', 'o', 'u'}
3     result = [string for string in input_list if string[0].lower() not in vowels]
4     return result
5
6 # Example list containing strings starting with vowels and consonants
7 input_list = ['apple', 'banana', 'pear', 'carrot', 'orange', 'grape']
8
9 # Removing strings starting with vowels
10 result_list = remove_vowel_starting_strings(input_list)
11
12 # Printing the result
13 print("Original list:", input_list)
14 print("List after removing strings starting with vowels:", result_list)
```



Input

```
Original list: ['apple', 'banana', 'pear', 'carrot', 'orange', 'grape']
List after removing strings starting with vowels: ['banana', 'pear', 'carrot', 'grape']
```

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

```
1 def transpose(matrix):
2     return [[matrix[j][i] for j in range(len(matrix))] for i in range(len(matrix))]
3
4 # Example matrix
5 matrix = [[1, 2, 3],
6           [4, 5, 6],
7           [7, 8, 9]]
8
9 # Transpose the matrix
10 transposed_matrix = transpose(matrix)
11
12 # Print the transposed matrix
13 for row in transposed_matrix:
14     print(row)
```



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