Q1. Create a python program to sort the given list of tuples based on integer value using a lambda function. [('Sachin Tendulkar', 34357), ('Ricky Ponting', 27483), ('Jack Kallis', 25534), ('Virat Kohli', 24936)]

Answer:

To use the sorted() function along with a lambda function to sort the list of tuples based on the integer value. Here's a Python program that does that:

In this program:

The sorted() function is used to sort the list of tuples.

The key=lambda x: x[1] tells the sorted() function to use the second element (integer score) of each tuple for sorting.

The reverse=True sorts the list in descending order. If you want ascending order, just remove reverse=True.

```
# List of tuples
cricketers = [
    ('Sachin Tendulkar', 34357),
    ('Ricky Ponting', 27483),
    ('Jack Kallis', 25534),
    ('Virat Kohli', 24936)
]

# Sorting the list of tuples based on the integer value (second element in each tuple)
sorted_cricketers = sorted(cricketers, key=lambda x: x[1], reverse=True)

# Displaying the sorted list
for player in sorted_cricketers:
    print(player)
```

```
('Sachin Tendulkar', 34357)
('Ricky Ponting', 27483)
('Jack Kallis', 25534)
('Virat Kohli', 24936)
```

Q2. Write a Python Program to find the squares of all the numbers in the given list of integers using lambda and map functions. [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Answer:

using the map() function along with a lambda function to find the squares of all the numbers in the list. Here's a Python program that does that:

```
# List of integers
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# Using map and lambda to find the squares of all numbers
squares = list(map(lambda x: x ** 2, numbers))

# Displaying the result
print(squares)
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

Q3. Write a python program to convert the given list of integers into a tuple of strings. Use map and lambda functions

Given String: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Expected output: ('1', '2', '3', '4', '5', '6', '7', '8', '9', '10')

using the map() function along with a lambda function to convert the given list of integers into a tuple of strings. Here's the Python program to achieve this:

```
# List of integers
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# Using map and lambda to convert each integer to a string
string_tuple = tuple(map(lambda x: str(x), numbers))

# Displaying the result
print(string_tuple)
```

```
('1', '2', '3', '4', '5', '6', '7', '8', '9', '10')
```

Q4. Write a python program using reduce function to compute the product of a list containing numbers from 1 to 25.

Using the reduce() function from the functools module to compute the product of a list of numbers. Here's the Python program to compute the product of numbers from 1 to 25:

```
from functools import reduce

# List of numbers from 1 to 25
numbers = list(range(1, 26))

# Using reduce to compute the product of all numbers in the list
product = reduce(lambda x, y: x * y, numbers)

# Displaying the result
print(product)
```

```
15511210043330985984000000
```

Q5. Write a python program to filter the numbers in a given list that are divisible by 2 and 3 using the filter function. [2, 3, 6, 9, 27, 60, 90, 120, 55, 46]

Using the filter() function along with a lambda function to filter the numbers divisible by both 2 and 3. Here's the Python program for that:

```
# List of numbers
numbers = [2, 3, 6, 9, 27, 60, 90, 120, 55, 46]

# Using filter and lambda to get numbers divisible by both 2 and 3
divisible_by_2_and_3 = list(filter(lambda x: x % 2 == 0 and x % 3 == 0, numbers))

# Displaying the result
print(divisible_by_2_and_3)
```

```
[6, 60, 90, 120]
```

Q6. Write a python program to find palindromes in the given list of strings using lambda and filter function.

['python', 'php', 'aba', 'radar', 'level']

Using the filter() function along with a lambda function to find palindromes in the given list of strings. Here's the Python program for that:

```
# List of strings
words = ['python', 'php', 'aba', 'radar', 'level']

# Using filter and lambda to find palindromes
palindromes = list(filter(lambda x: x == x[::-1], words))

# Displaying the result
print(palindromes)
```

```
['php', 'aba', 'radar', 'level']
```

Thank you!