Problem Solving using Python Programming

Course code-24CSE0101

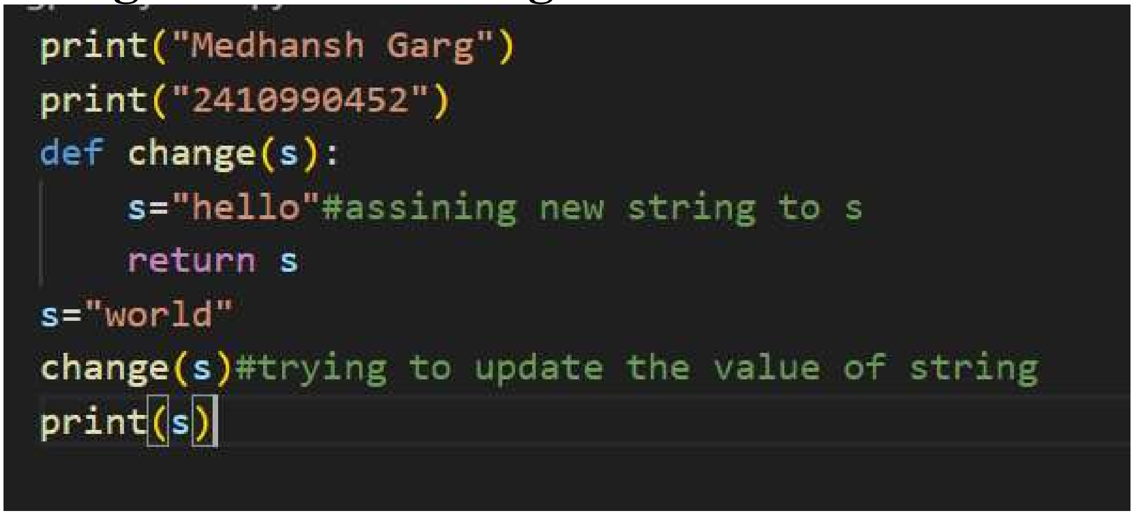
Submitted by:

Name:-Medhansh Garg

Roll no:2410990452

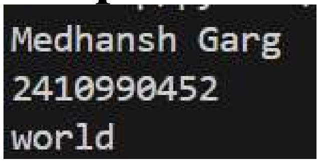
Group 6

Write a Python program that demonstrates passing a string by value to a function. The function should attempt to modify the string, and the program should show that the original string remains unchanged.



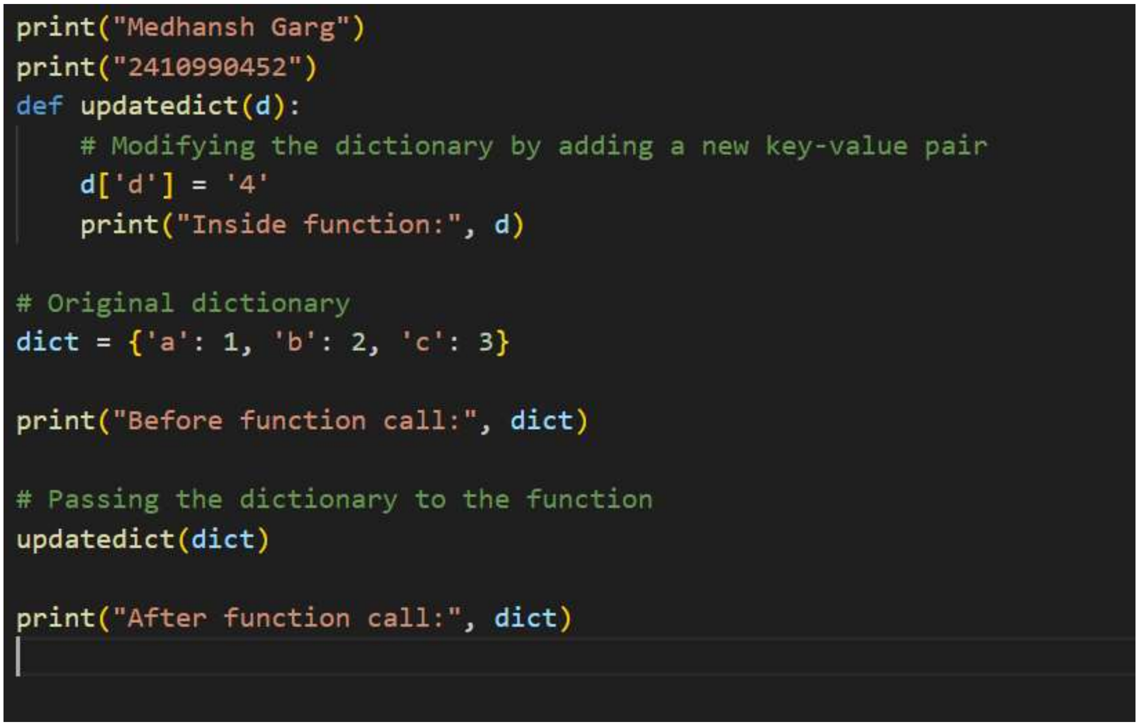
```
print("Medhansh Garg")
print("2410990452")
def change(s):
    s="hello"#assining new string to s
    return s
s="world"
change(s)#trying to update the value of string
print(s)
```

Output:



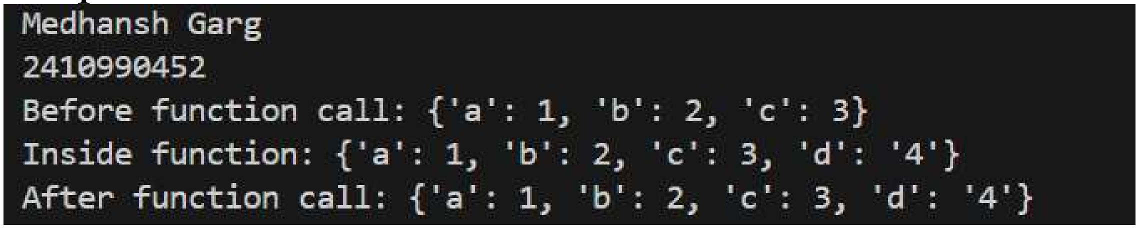
```
Medhansh Garg
2410990452
world
```

Write a Python program that demonstrates passing a dictionary by reference to a function. Modify the dictionary inside the function and show that the original dictionary is updated.



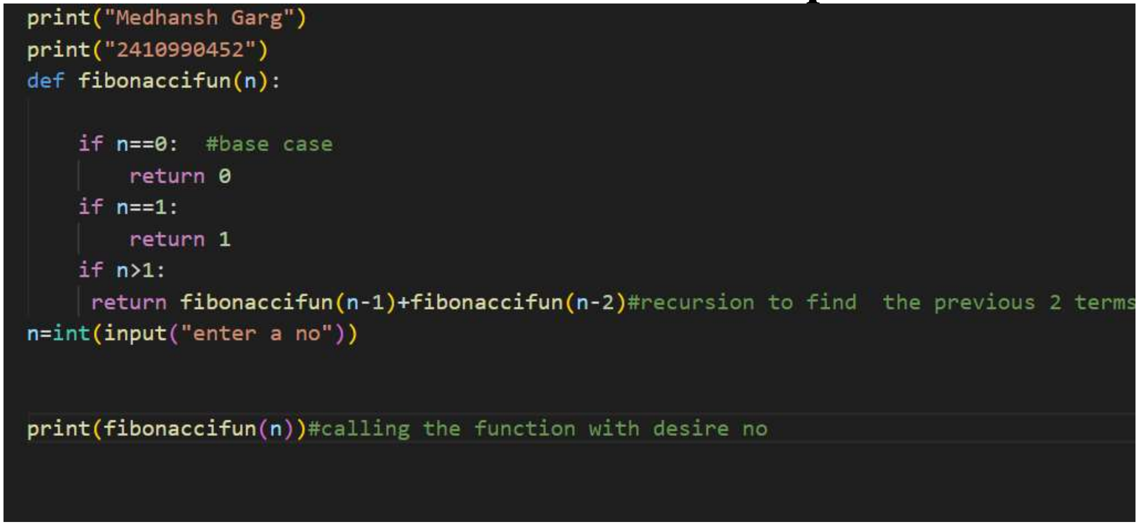
```python
print("Medhansh Garg")
print("2410990452")
def updatedict(d):
    # Modifying the dictionary by adding a new key-value pair
    d['d'] = '4'
    print("Inside function:", d)

# Original dictionary
dict = {'a': 1, 'b': 2, 'c': 3}

print("Before function call:", dict)

# Passing the dictionary to the function
updatedict(dict)

print("After function call:", dict)
```

Output:



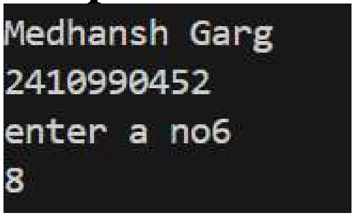
```
Medhansh Garg
2410990452
Before function call: {'a': 1, 'b': 2, 'c': 3}
Inside function: {'a': 1, 'b': 2, 'c': 3, 'd': '4'}
After function call: {'a': 1, 'b': 2, 'c': 3, 'd': '4'}
```

Write a Python program that defines a recursive function fibonacci(n) to compute the nth Fibonacci number. Test the function with user input.



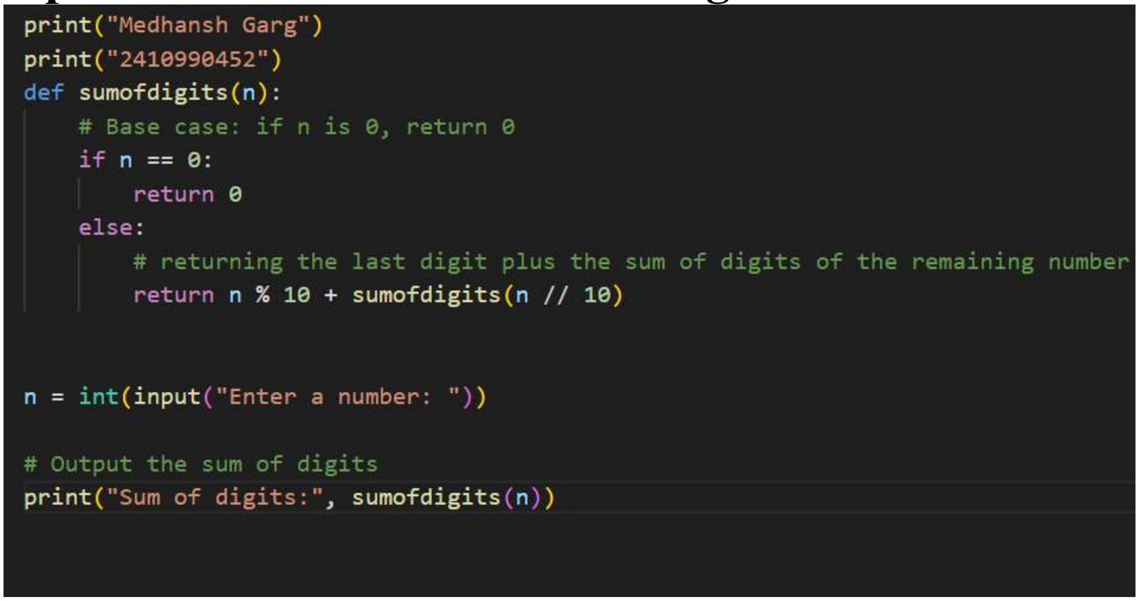
```
print("Medhansh Garg")
print("2410990452")
def fibonaccifun(n):

    if n==0:  #base case
        return 0
    if n==1:
        return 1
    if n>1:
     return fibonaccifun(n-1)+fibonaccifun(n-2)#recursion to find  the previous 2 terms
n=int(input("enter a no"))


print(fibonaccifun(n))#calling the function with desire no
```

Output:



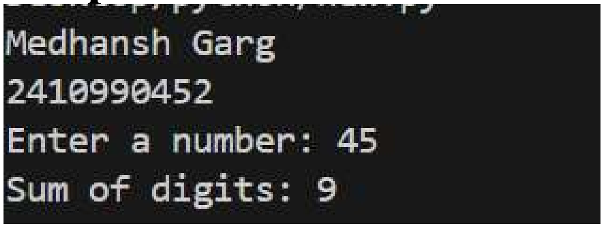
```
Medhansh Garg
2410990452
enter a no6
8
```

Write a Python program that defines a recursive function to calculate the sum of digits of a given number. The function should take an integer as input and return the sum of its digits.



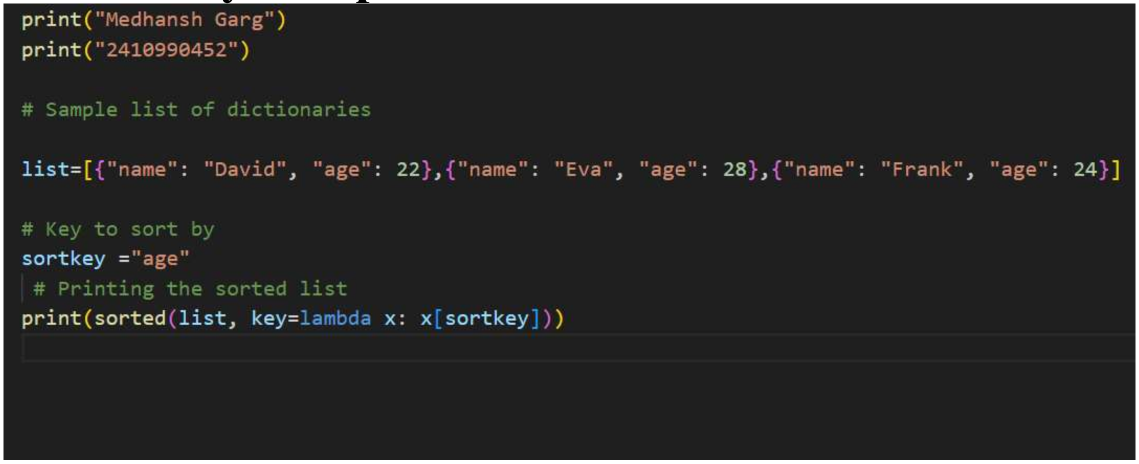
```python
print("Medhansh Garg")
print("2410990452")
def sumofdigits(n):
    # Base case: if n is 0, return 0
    if n == 0:
        return 0
    else:
        # returning the last digit plus the sum of digits of the remaining number
        return n % 10 + sumofdigits(n // 10)


n = int(input("Enter a number: "))

# Output the sum of digits
print("Sum of digits:", sumofdigits(n))
```

Output:



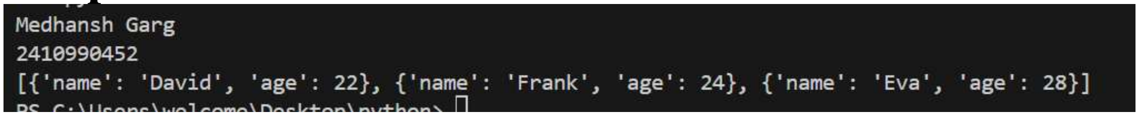
```
Medhansh Garg
2410990452
Enter a number: 45
Sum of digits: 9
```

Write a Python program that uses a lambda expression to sort a list of dictionaries based on the value of a specific key. The program should take the list and key as input.



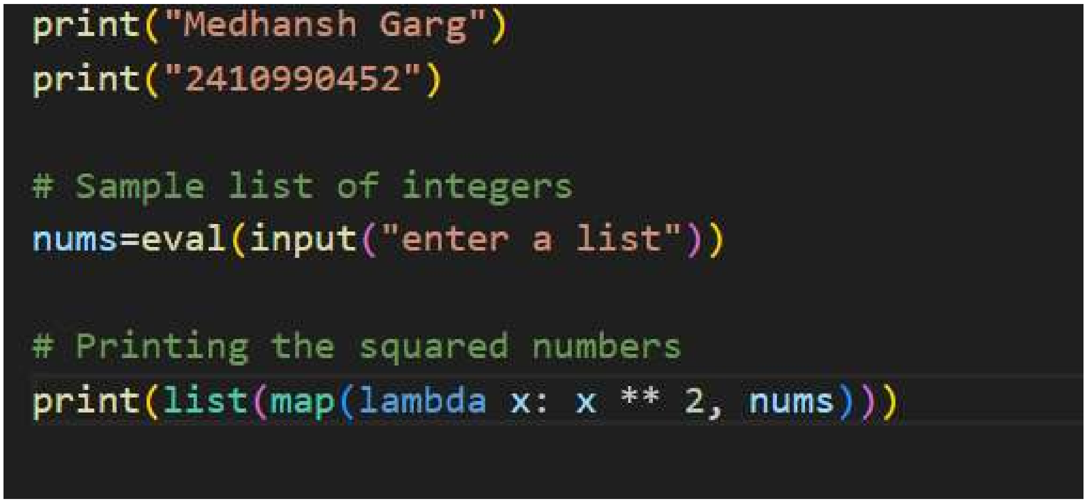
```python
print("Medhansh Garg")
print("2410990452")

# Sample list of dictionaries

list=[{"name": "David", "age": 22},{"name": "Eva", "age": 28},{"name": "Frank", "age": 24}]

# Key to sort by
sortkey ="age"
 # Printing the sorted list
print(sorted(list, key=lambda x: x[sortkey]))
```

Output:



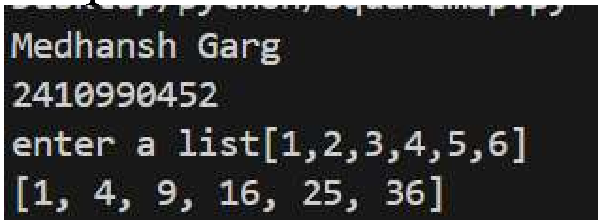
```
Medhansh Garg
2410990452
[{'name': 'David', 'age': 22}, {'name': 'Frank', 'age': 24}, {'name': 'Eva', 'age': 28}]
PS C:\Users\welcome\Desktop\python>
```

Write a Python program that uses a lambda expression and the map() function to compute the square of each number in a list of integers.



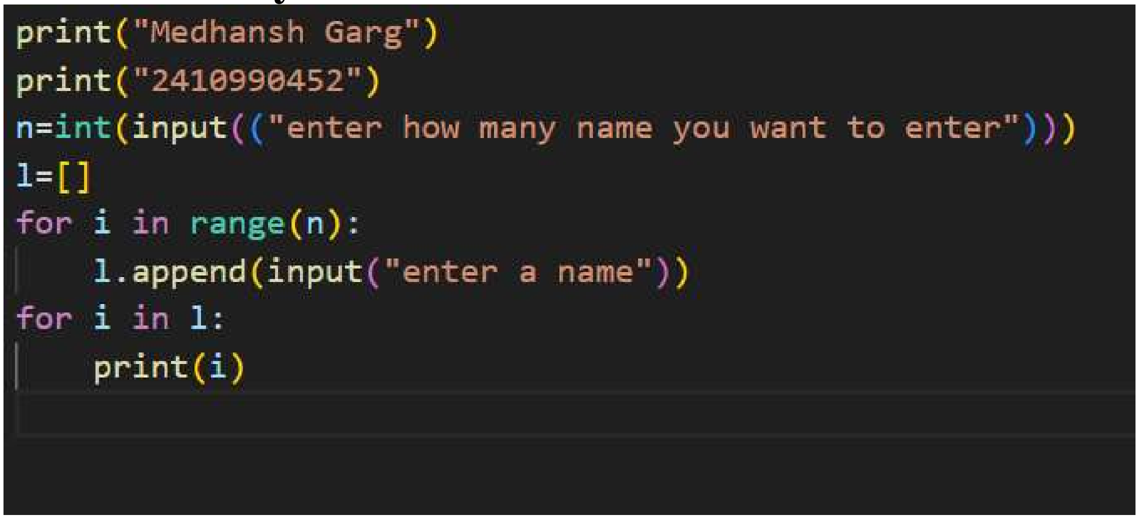
```python
print("Medhansh Garg")
print("2410990452")

# Sample list of integers
nums=eval(input("enter a list"))

# Printing the squared numbers
print(list(map(lambda x: x ** 2, nums)))
```

Output:



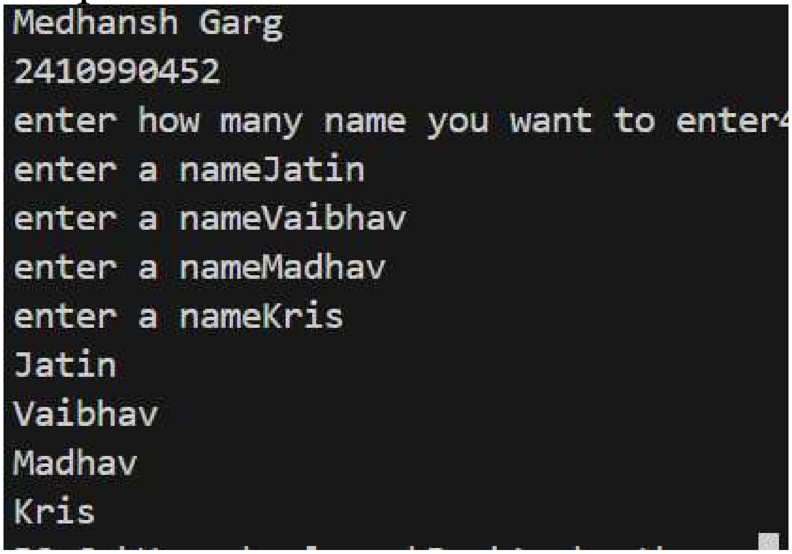
```
Medhansh Garg
2410990452
enter a list[1,2,3,4,5,6]
[1, 4, 9, 16, 25, 36]
```

Write a Python program that initializes a list of names entered by the user and prints each name using a for loop. Allow the user to enter as many names as they like.



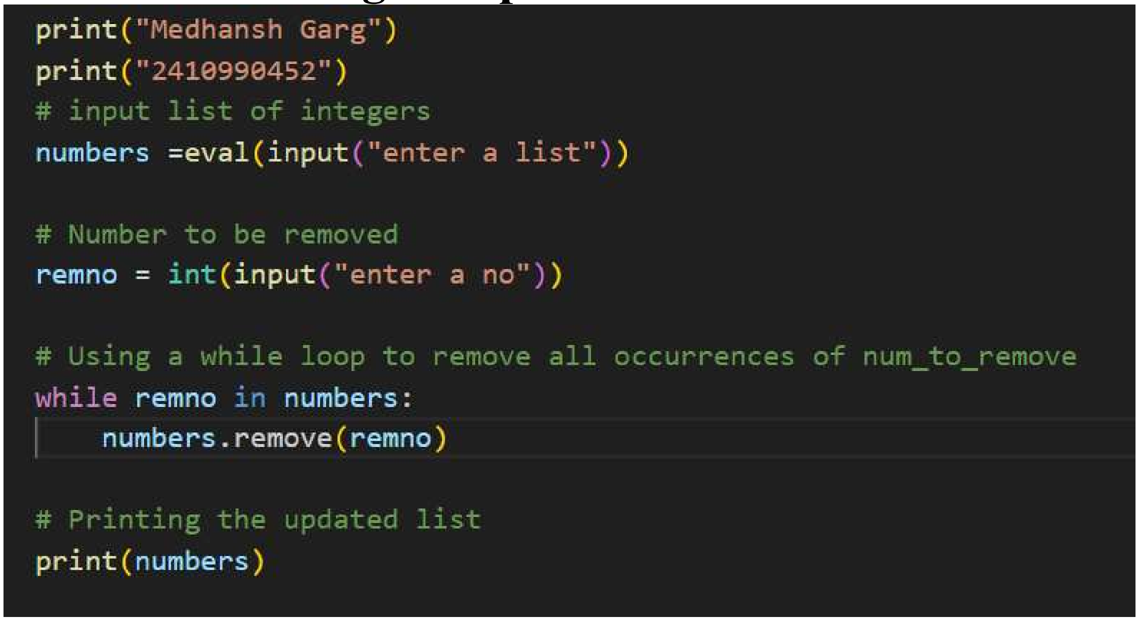
```
print("Medhansh Garg")
print("2410990452")
n=int(input(("enter how many name you want to enter")))
l=[]
for i in range(n):
    l.append(input("enter a name"))
for i in l:
    print(i)
```

Output:



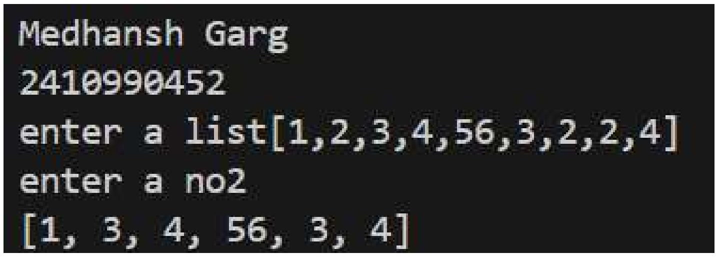
```
Medhansh Garg
2410990452
enter how many name you want to enter4
enter a nameJatin
enter a nameVaibhav
enter a nameMadhav
enter a nameKris
Jatin
Vaibhav
Madhav
Kris
```

Write a Python program that takes a list of integers and a number n as input. Remove all occurrences of n from the list using a loop.



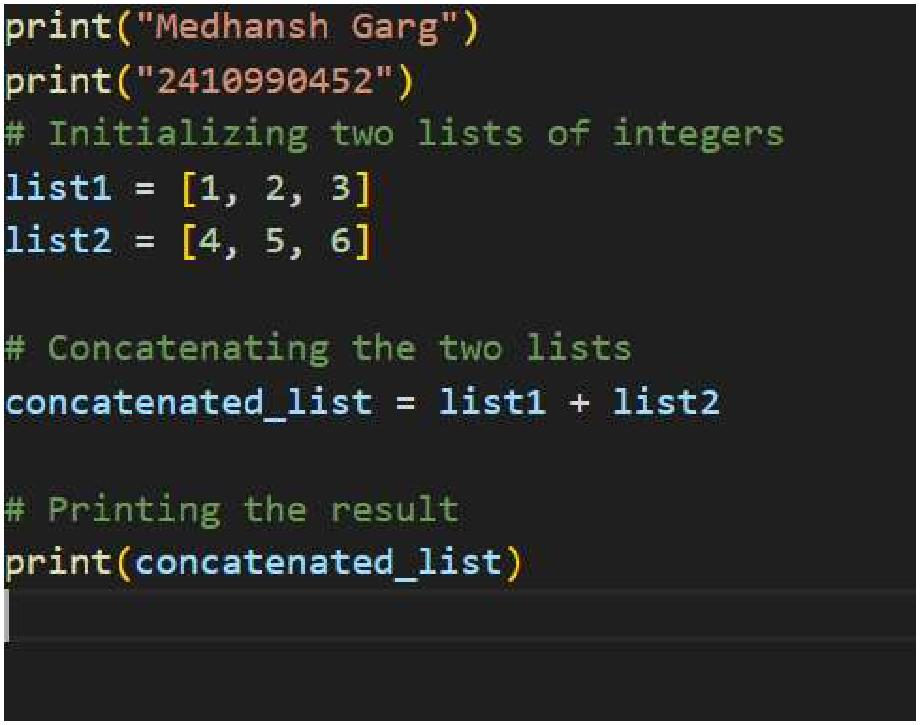
```python
print("Medhansh Garg")
print("2410990452")
# input list of integers
numbers =eval(input("enter a list"))

# Number to be removed
remno = int(input("enter a no"))

# Using a while loop to remove all occurrences of num_to_remove
while remno in numbers:
    numbers.remove(remno)

# Printing the updated list
print(numbers)
```

Output:



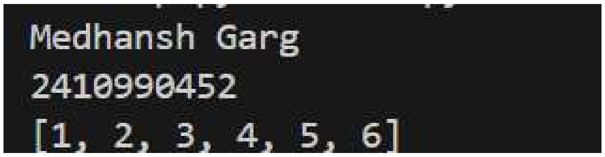
```
Medhansh Garg
2410990452
enter a list[1,2,3,4,56,3,2,2,4]
enter a no2
[1, 3, 4, 56, 3, 4]
```

Write a Python program that initializes two lists of integers and concatenates them into a single list. Print the result.



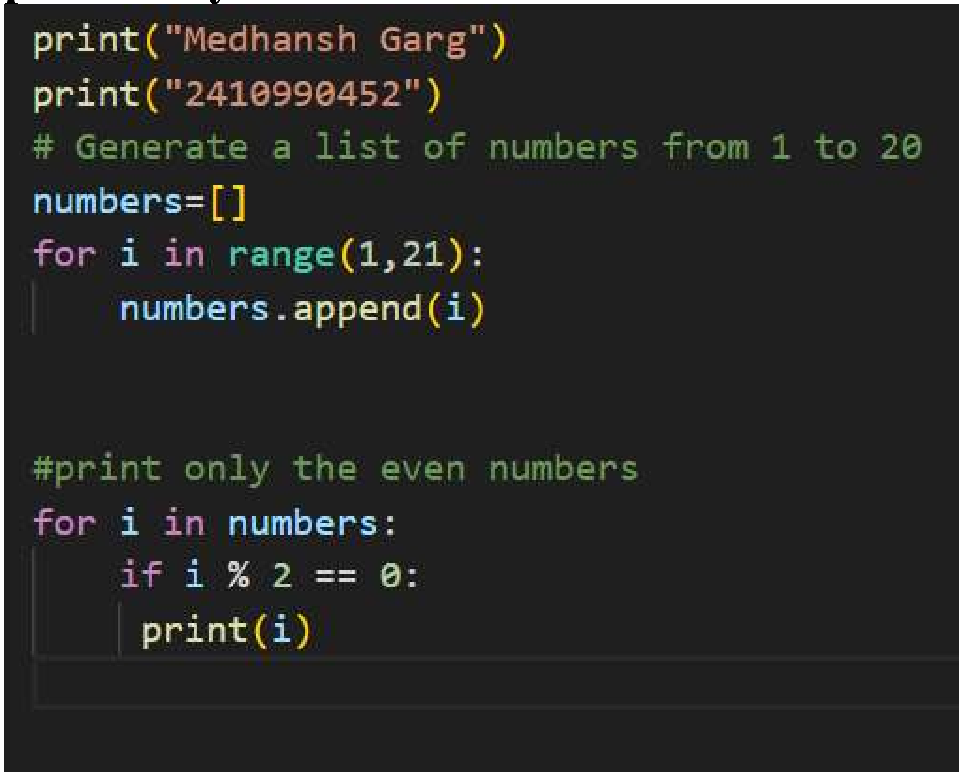
```python
print("Medhansh Garg")
print("2410990452")
# Initializing two lists of integers
list1 = [1, 2, 3]
list2 = [4, 5, 6]

# Concatenating the two lists
concatenated_list = list1 + list2

# Printing the result
print(concatenated_list)
```

Output:



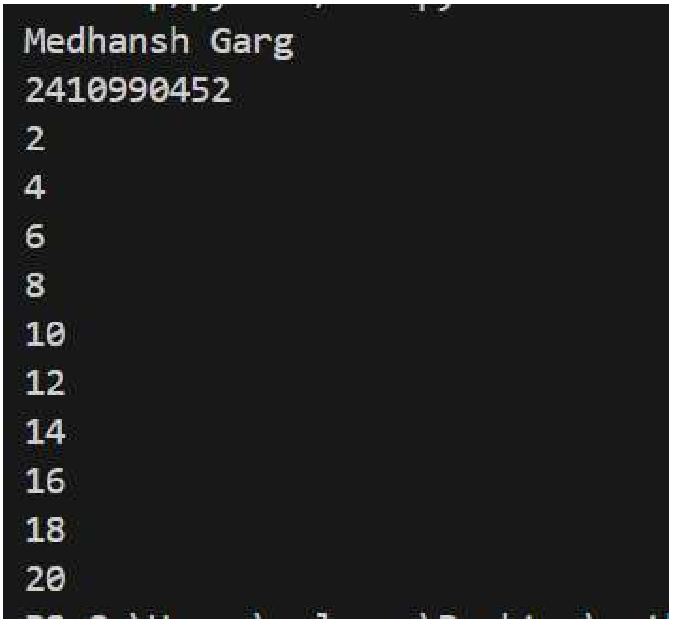
```
Medhansh Garg
2410990452
[1, 2, 3, 4, 5, 6]
```

Write a Python program that generates a list of numbers from 1 to 20 using the range() function and prints only the even numbers from the list.



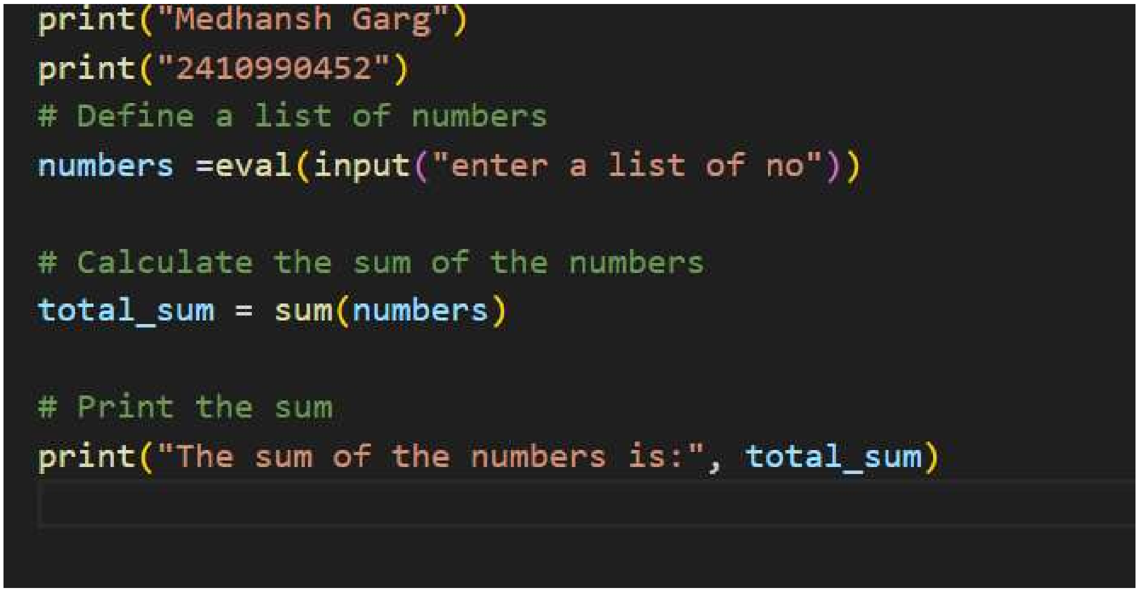
```
print("Medhansh Garg")
print("2410990452")
# Generate a list of numbers from 1 to 20
numbers=[]
for i in range(1,21):
    numbers.append(i)


#print only the even numbers
for i in numbers:
    if i % 2 == 0:
     print(i)
```

Output



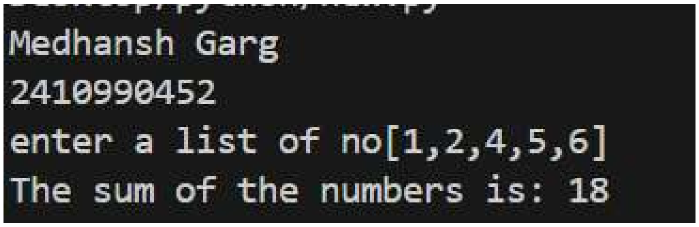
```
Medhansh Garg
2410990452
2
4
6
8
10
12
14
16
18
20
```

Write a Python program that takes a list of numbers and prints the sum of the numbers using the sum() function.



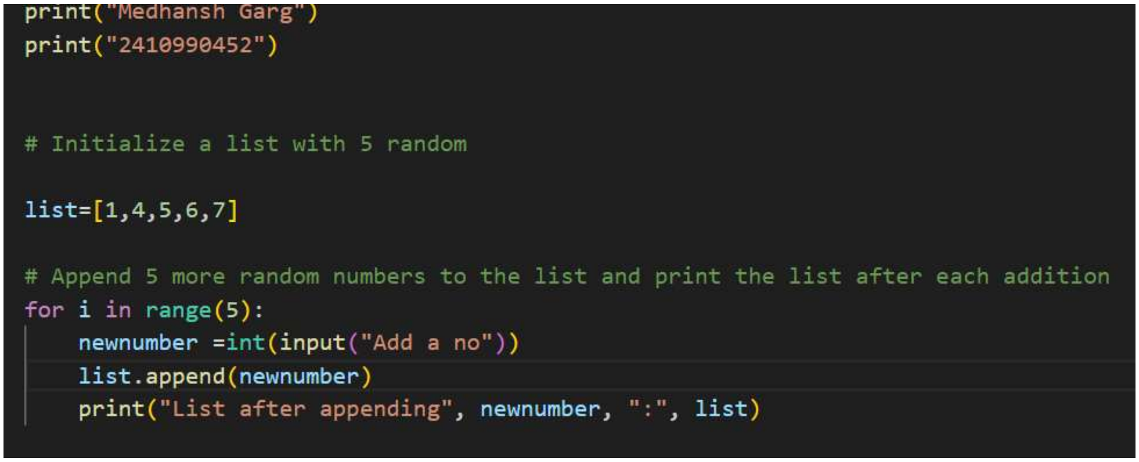
```
print("Medhansh Garg")
print("2410990452")
# Define a list of numbers
numbers =eval(input("enter a list of no"))

# Calculate the sum of the numbers
total_sum = sum(numbers)

# Print the sum
print("The sum of the numbers is:", total_sum)
```

Output:



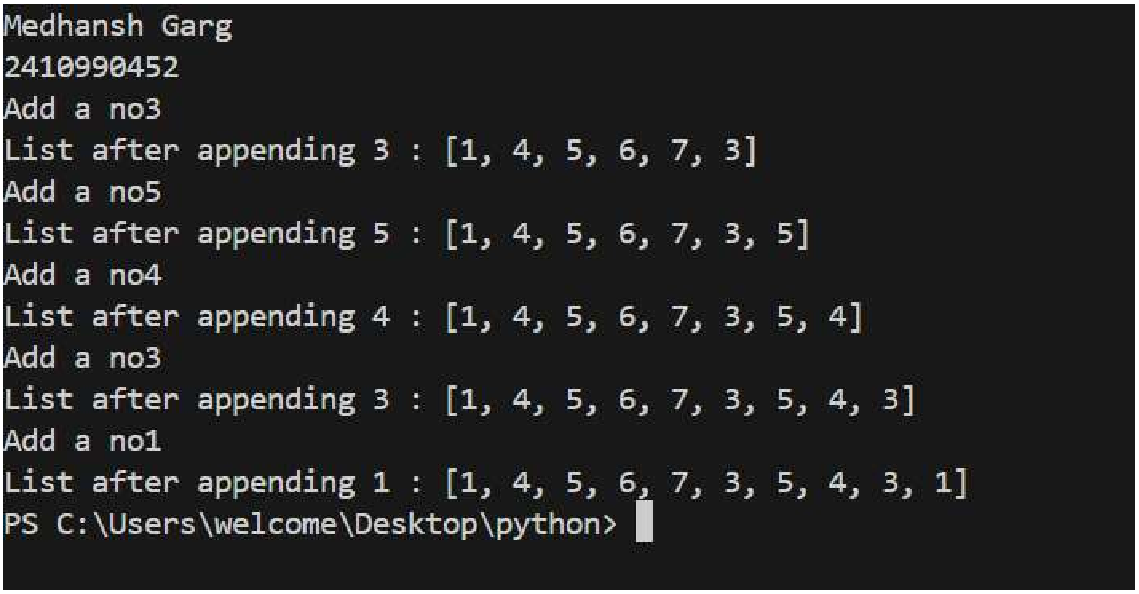
```
Medhansh Garg
2410990452
enter a list of no[1,2,4,5,6]
The sum of the numbers is: 18
```

Write a Python program to initialize a list with random numbers and demonstrate the use of append() to add more numbers to the list. Print the list after each addition.



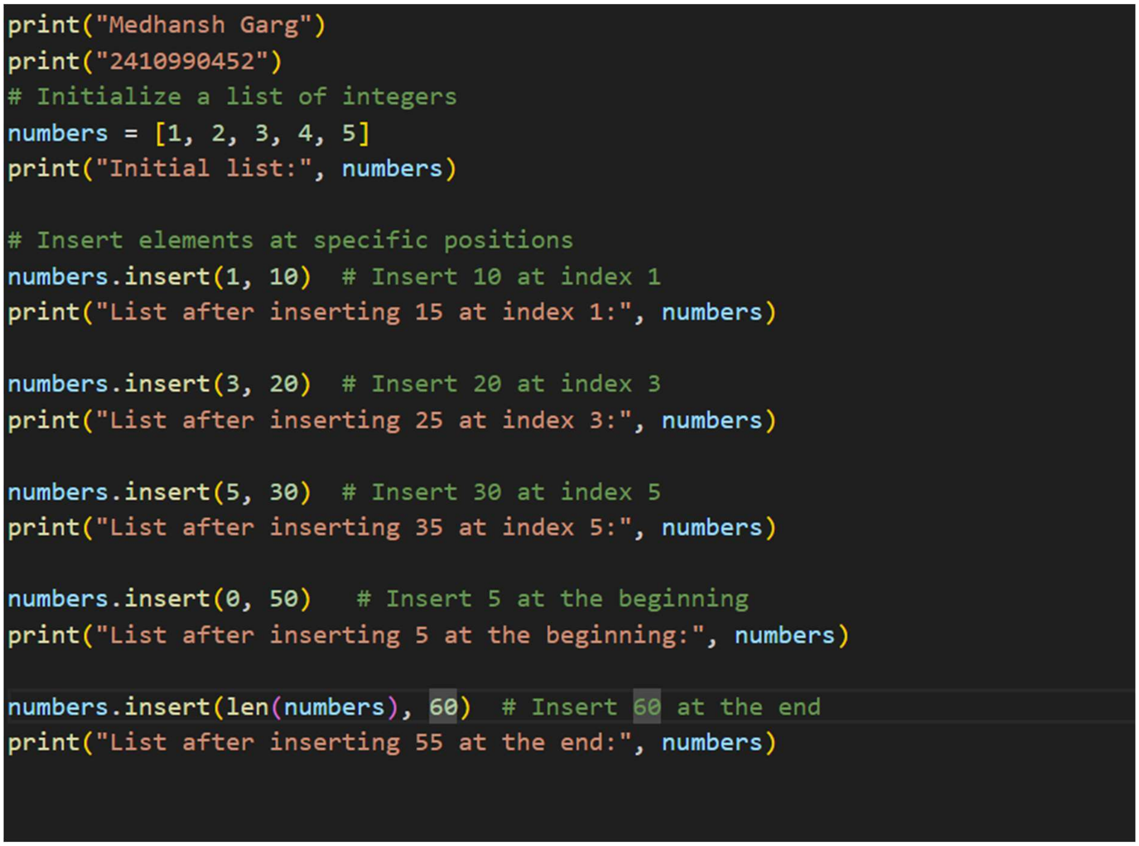
```
print("Medhansh Garg")
print("2410990452")


# Initialize a list with 5 random

list=[1,4,5,6,7]

# Append 5 more random numbers to the list and print the list after each addition
for i in range(5):
    newnumber =int(input("Add a no"))
    list.append(newnumber)
    print("List after appending", newnumber, ":", list)
```

Output:



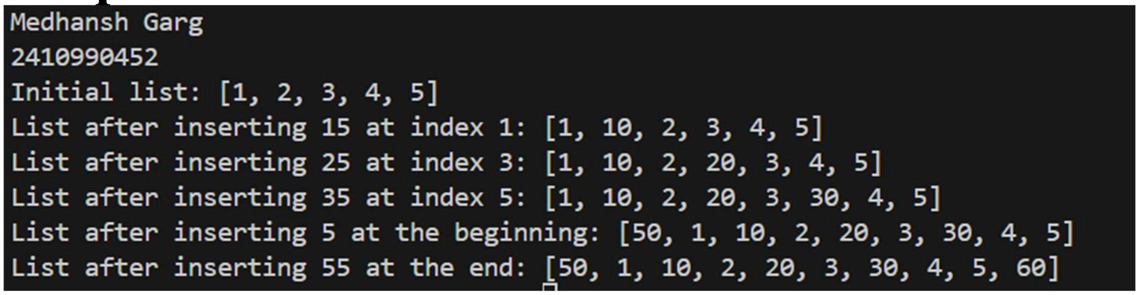
```
Medhansh Garg
2410990452
Add a no3
List after appending 3 : [1, 4, 5, 6, 7, 3]
Add a no5
List after appending 5 : [1, 4, 5, 6, 7, 3, 5]
Add a no4
List after appending 4 : [1, 4, 5, 6, 7, 3, 5, 4]
Add a no3
List after appending 3 : [1, 4, 5, 6, 7, 3, 5, 4, 3]
Add a no1
List after appending 1 : [1, 4, 5, 6, 7, 3, 5, 4, 3, 1]
PS C:\Users\welcome\Desktop\python>
```

Write a Python program that initializes a list of integers and uses the insert() method to add elements at specific positions. Print the list after each insertion.



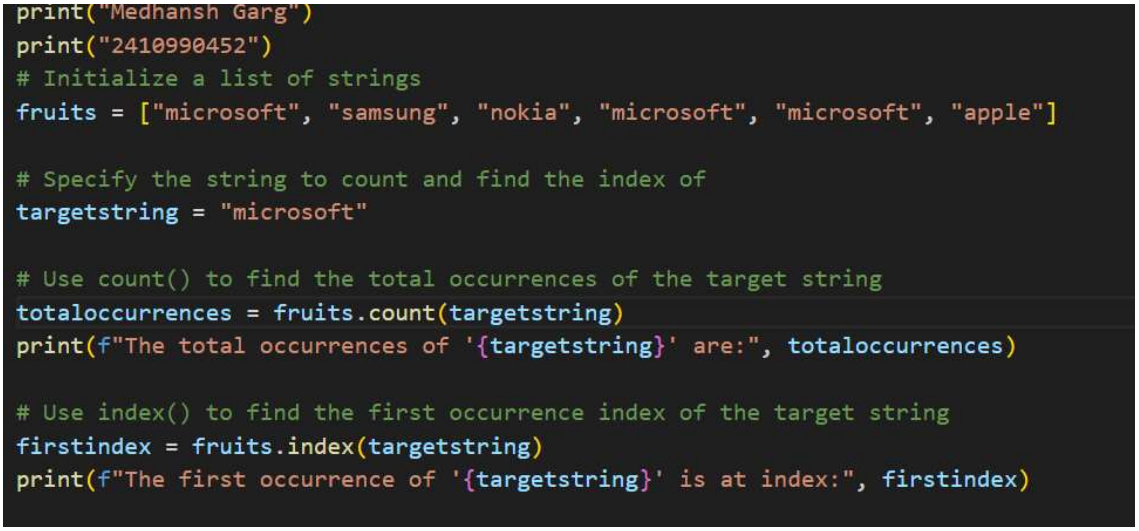
```python
print("Medhansh Garg")
print("2410990452")
# Initialize a list of integers
numbers = [1, 2, 3, 4, 5]
print("Initial list:", numbers)

# Insert elements at specific positions
numbers.insert(1, 10)  # Insert 10 at index 1
print("List after inserting 15 at index 1:", numbers)

numbers.insert(3, 20)  # Insert 20 at index 3
print("List after inserting 25 at index 3:", numbers)

numbers.insert(5, 30)  # Insert 30 at index 5
print("List after inserting 35 at index 5:", numbers)

numbers.insert(0, 50)   # Insert 5 at the beginning
print("List after inserting 5 at the beginning:", numbers)

numbers.insert(len(numbers), 60)  # Insert 60 at the end
print("List after inserting 55 at the end:", numbers)
```

Output:



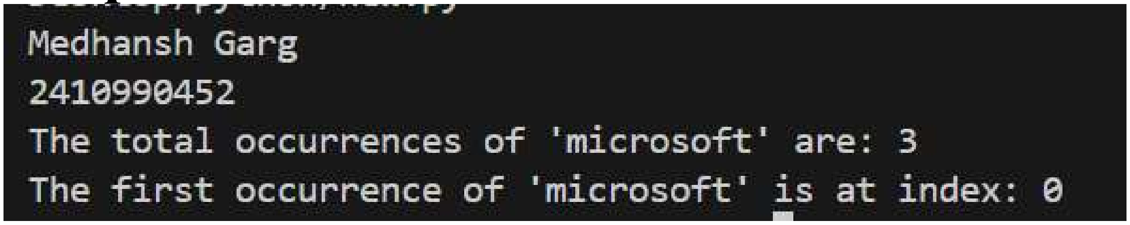
```
Medhansh Garg
2410990452
Initial list: [1, 2, 3, 4, 5]
List after inserting 15 at index 1: [1, 10, 2, 3, 4, 5]
List after inserting 25 at index 3: [1, 10, 2, 20, 3, 4, 5]
List after inserting 35 at index 5: [1, 10, 2, 20, 3, 30, 4, 5]
List after inserting 5 at the beginning: [50, 1, 10, 2, 20, 3, 30, 4, 5]
List after inserting 55 at the end: [50, 1, 10, 2, 20, 3, 30, 4, 5, 60]
```

Write a Python program that initializes a list of strings and demonstrates the use of count() and index() methods. Print the total occurrences of a specific string and its first occurrence index.



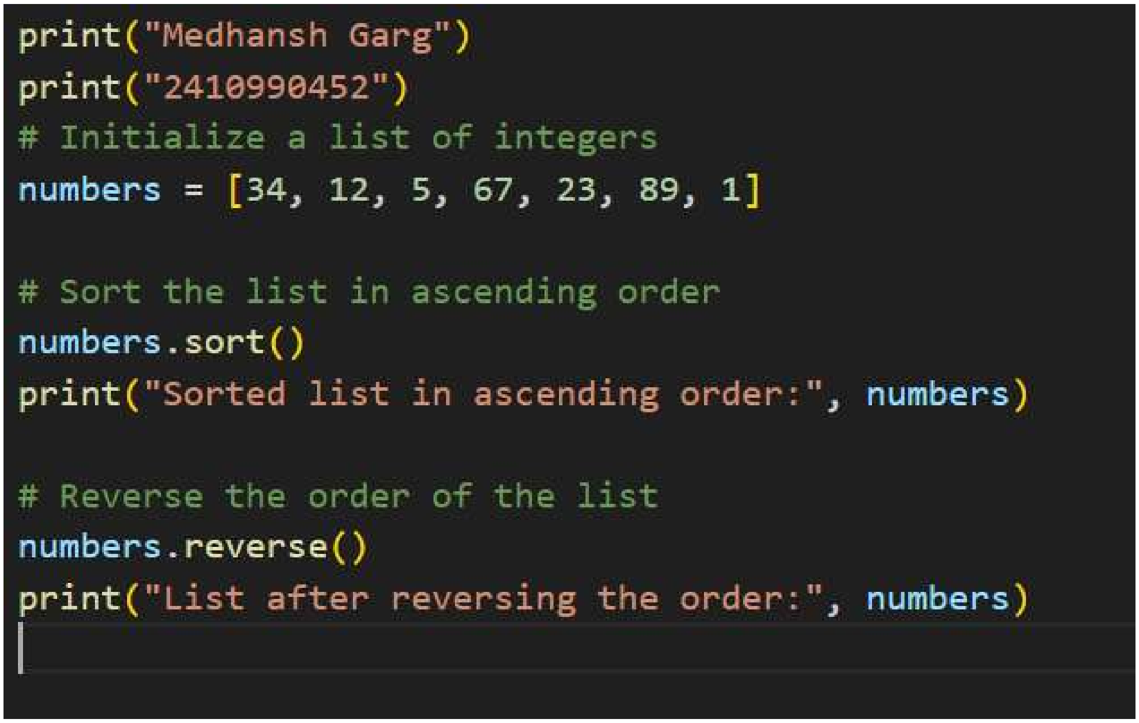
```
print("Medhansh Garg")
print("2410990452")
# Initialize a list of strings
fruits = ["microsoft", "samsung", "nokia", "microsoft", "microsoft", "apple"]

# Specify the string to count and find the index of
targetstring = "microsoft"

# Use count() to find the total occurrences of the target string
totaloccurrences = fruits.count(targetstring)
print(f"The total occurrences of '{targetstring}' are:", totaloccurrences)

# Use index() to find the first occurrence index of the target string
firstindex = fruits.index(targetstring)
print(f"The first occurrence of '{targetstring}' is at index:", firstindex)
```

Output:



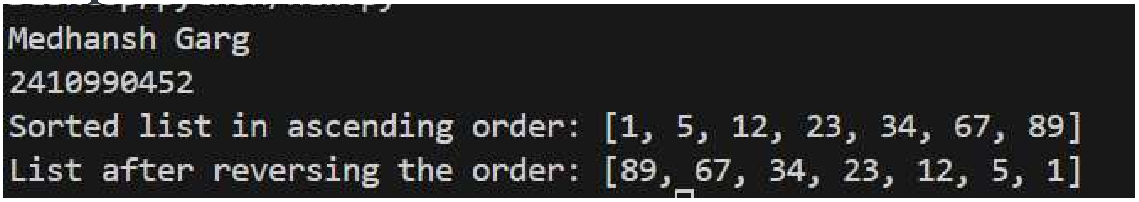
```
Medhansh Garg
2410990452
The total occurrences of 'microsoft' are: 3
The first occurrence of 'microsoft' is at index: 0
```

Write a Python program that sorts a list of integers in ascending order using the sort() method and then reverses the order using the reverse() method. Print the sorted and reversed lists.



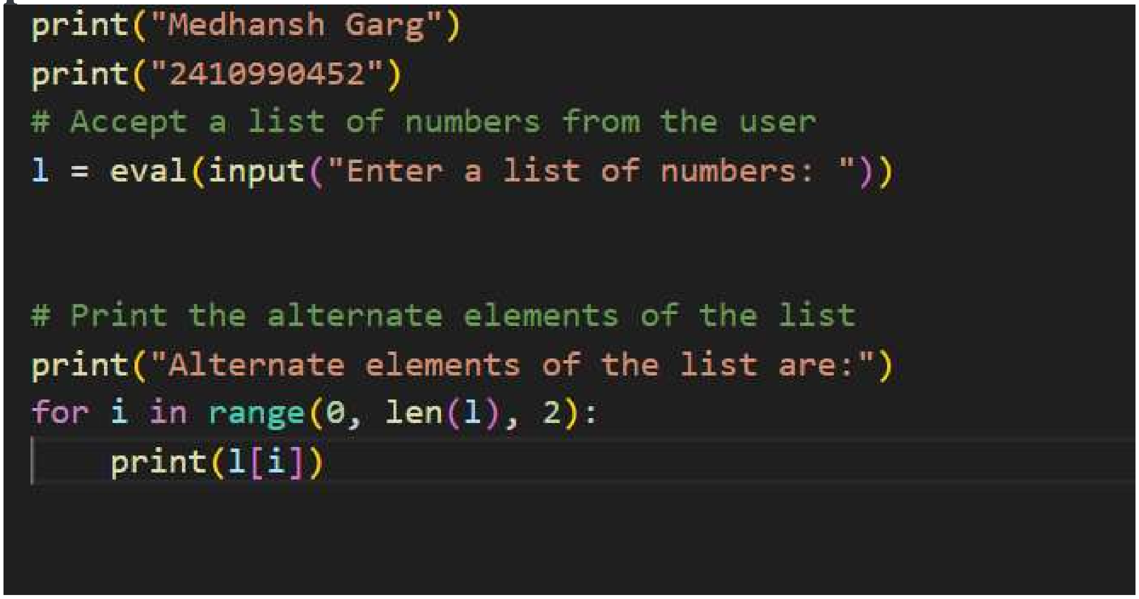
```python
print("Medhansh Garg")
print("2410990452")
# Initialize a list of integers
numbers = [34, 12, 5, 67, 23, 89, 1]

# Sort the list in ascending order
numbers.sort()
print("Sorted list in ascending order:", numbers)

# Reverse the order of the list
numbers.reverse()
print("List after reversing the order:", numbers)
```

Output:



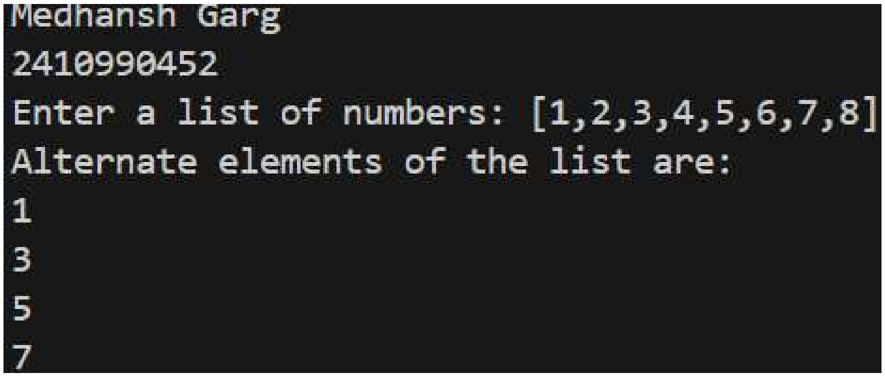
```
Medhansh Garg
2410990452
Sorted list in ascending order: [1, 5, 12, 23, 34, 67, 89]
List after reversing the order: [89, 67, 34, 23, 12, 5, 1]
```

Write a program that accepts a list from user and print the alternate element of list.



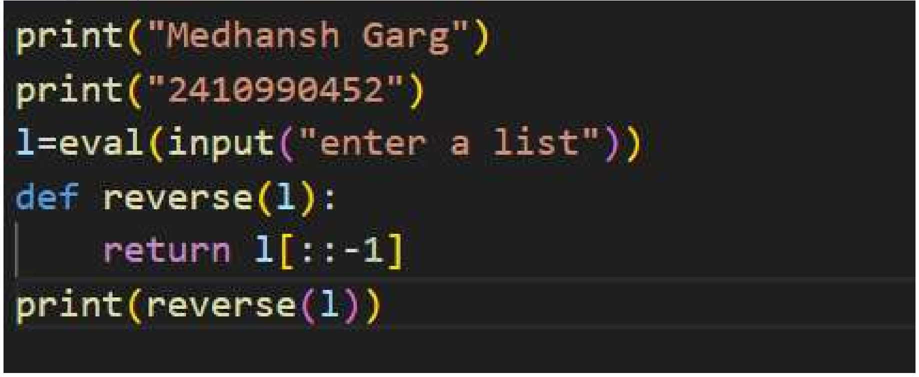
```python
print("Medhansh Garg")
print("2410990452")
# Accept a list of numbers from the user
l = eval(input("Enter a list of numbers: "))


# Print the alternate elements of the list
print("Alternate elements of the list are:")
for i in range(0, len(l), 2):
    print(l[i])
```

Output:



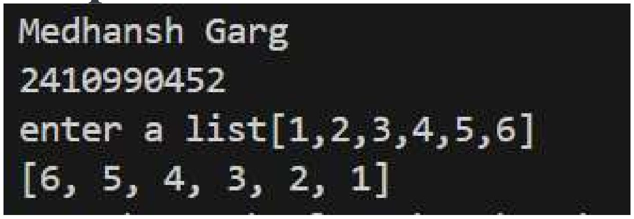
```
Medhansh Garg
2410990452
Enter a list of numbers: [1,2,3,4,5,6,7,8]
Alternate elements of the list are:
1
3
5
7
```

Write a program that accepts a list from user. Your program should reverse the content of list and display it. Do not use reverse () method.



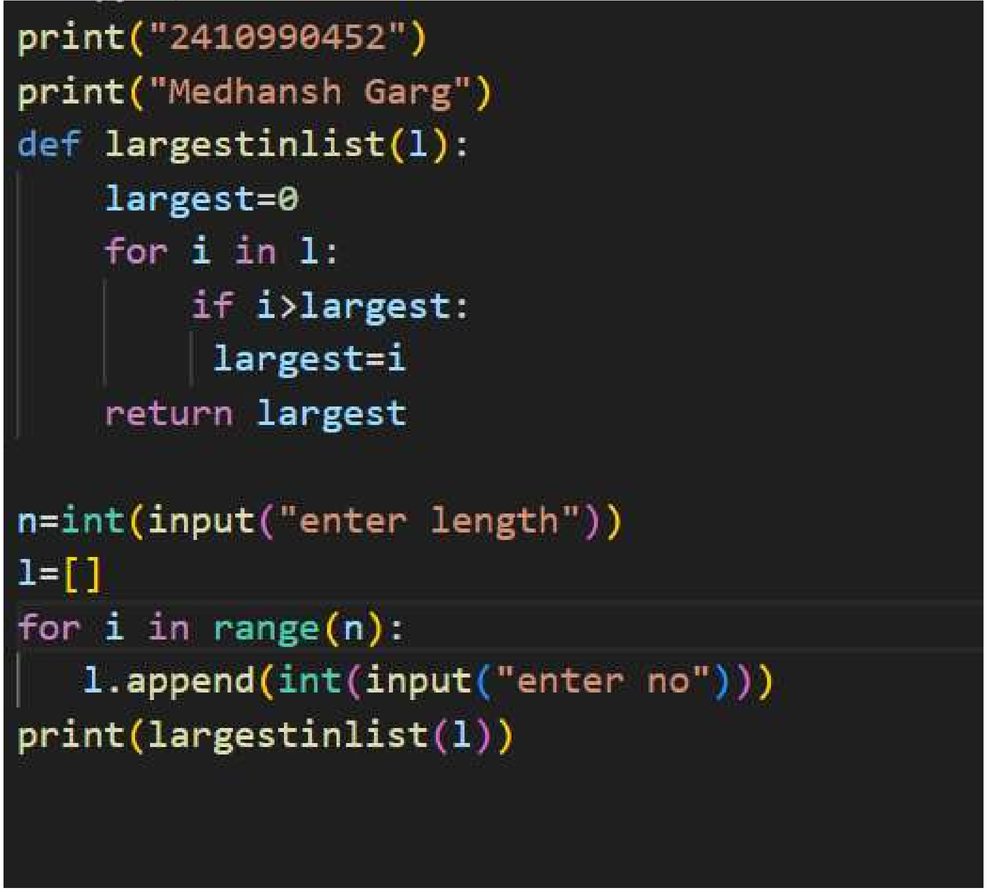
```
print("Medhansh Garg")
print("2410990452")
l=eval(input("enter a list"))
def reverse(l):
    return l[::-1]
print(reverse(l))
```

Output:



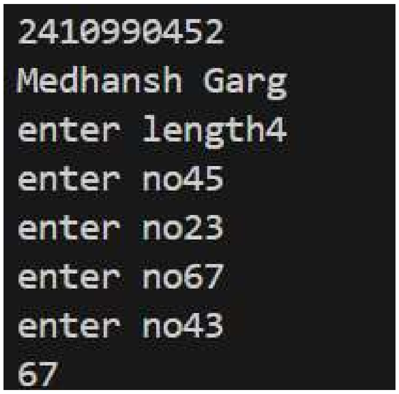
```
Medhansh Garg
2410990452
enter a list[1,2,3,4,5,6]
[6, 5, 4, 3, 2, 1]
```

Find and display the largest number of a list without using built-in function max(). Your program should ask the user to input values in list from keyboard.



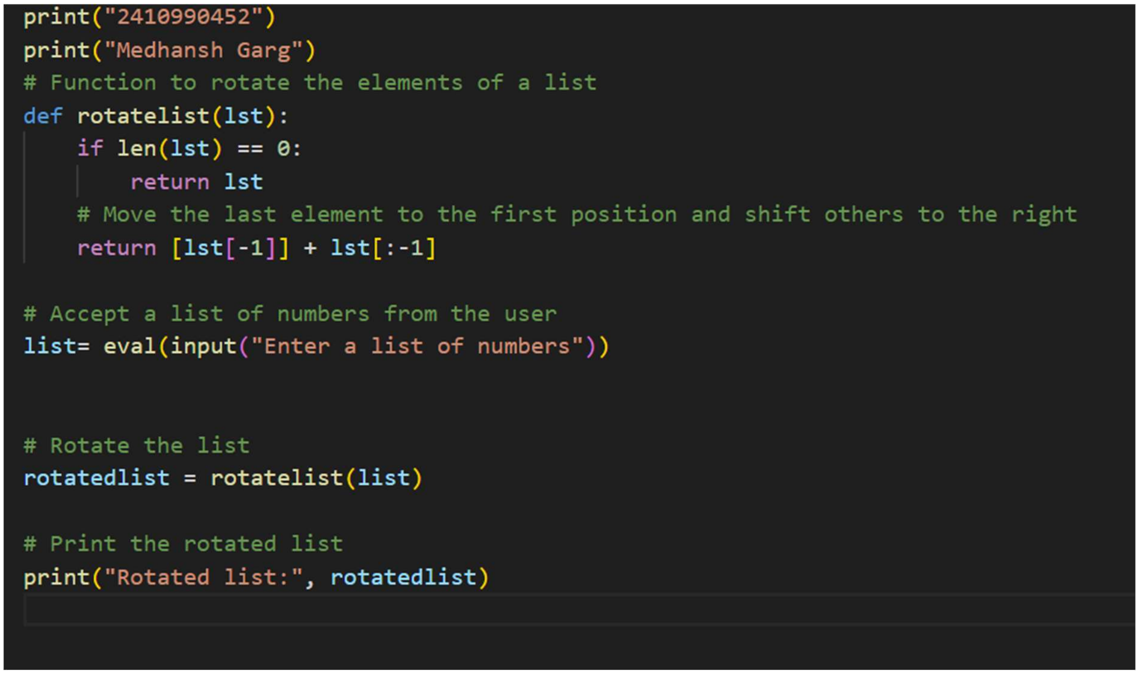
```
print("2410990452")
print("Medhansh Garg")
def largestinlist(l):
    largest=0
    for i in l:
        if i>largest:
         largest=i
    return largest

n=int(input("enter length"))
l=[]
for i in range(n):
   l.append(int(input("enter no")))
print(largestinlist(l))
```

Output:



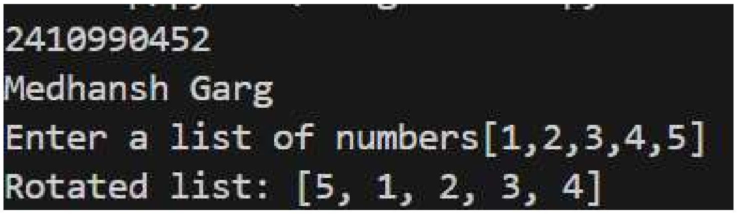
```
2410990452
Medhansh Garg
enter length4
enter no45
enter no23
enter no67
enter no43
67
```

Write a program that rotates the element of a list so that the element at the first index moves to the second index, the element in the second index moves to the third index, etc., and the element in the last index moves to the first index.



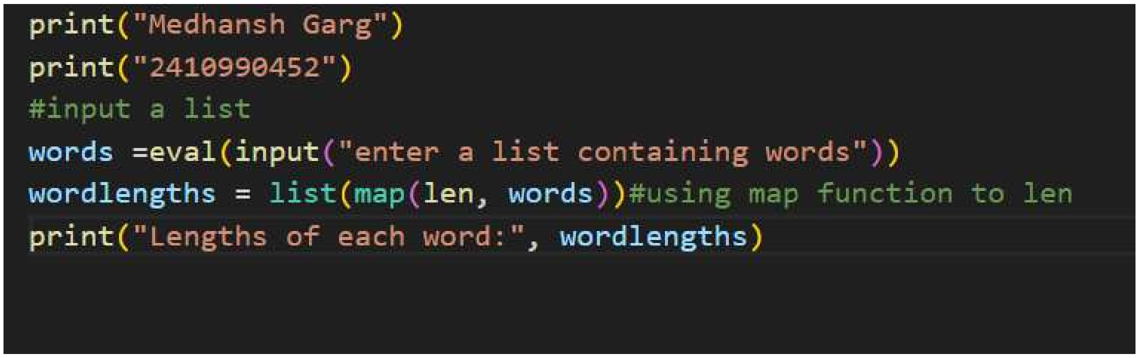
```python
print("2410990452")
print("Medhansh Garg")
# Function to rotate the elements of a list
def rotatelist(lst):
    if len(lst) == 0:
        return lst
    # Move the last element to the first position and shift others to the right
    return [lst[-1]] + lst[:-1]

# Accept a list of numbers from the user
list= eval(input("Enter a list of numbers"))


# Rotate the list
rotatedlist = rotatelist(list)

# Print the rotated list
print("Rotated list:", rotatedlist)
```

Output:



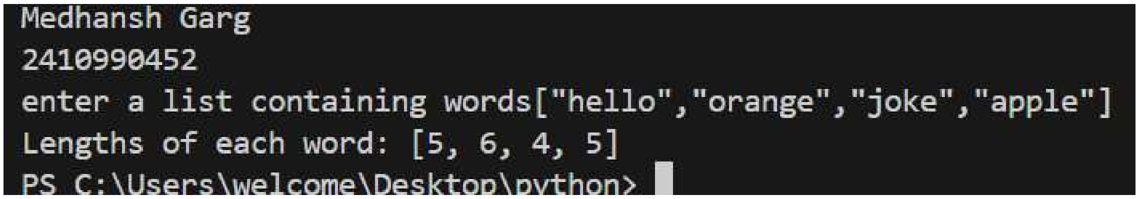
```
2410990452
Medhansh Garg
Enter a list of numbers[1,2,3,4,5]
Rotated list: [5, 1, 2, 3, 4]
```

Use the map function to find the length of each word in a list.



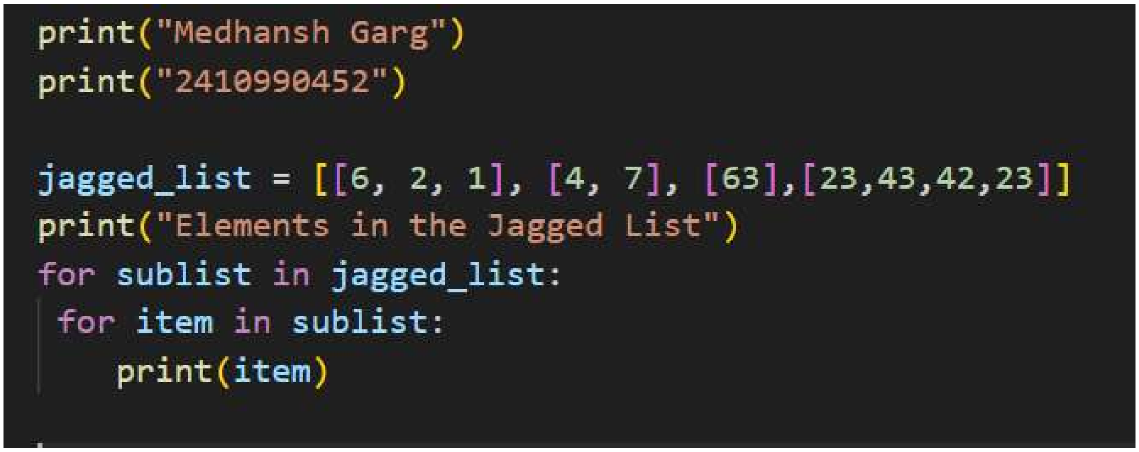
```python
print("Medhansh Garg")
print("2410990452")
#input a list
words =eval(input("enter a list containing words"))
wordlengths = list(map(len, words))#using map function to len
print("Lengths of each word:", wordlengths)
```

Output:



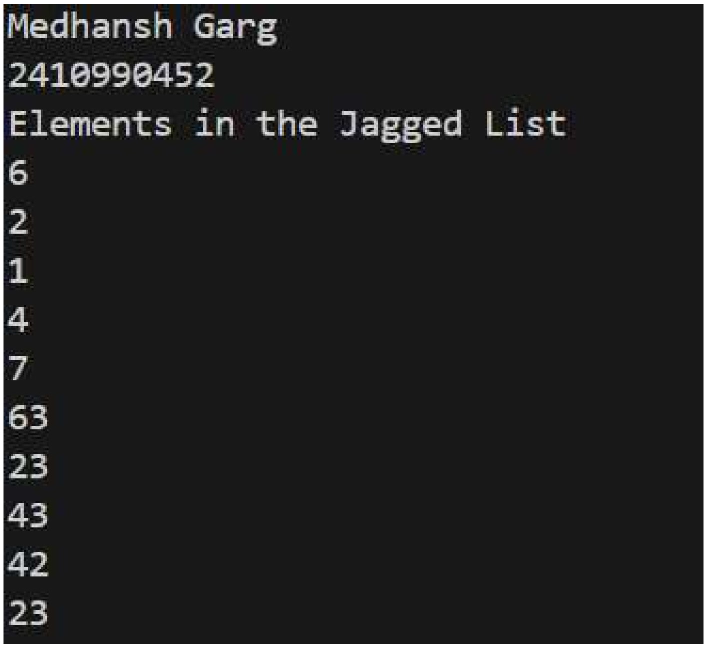
```
Medhansh Garg
2410990452
enter a list containing words["hello","orange","joke","apple"]
Lengths of each word: [5, 6, 4, 5]
PS C:\Users\welcome\Desktop\python>
```

Create a jagged list containing three sub lists of different lengths and print the elements.



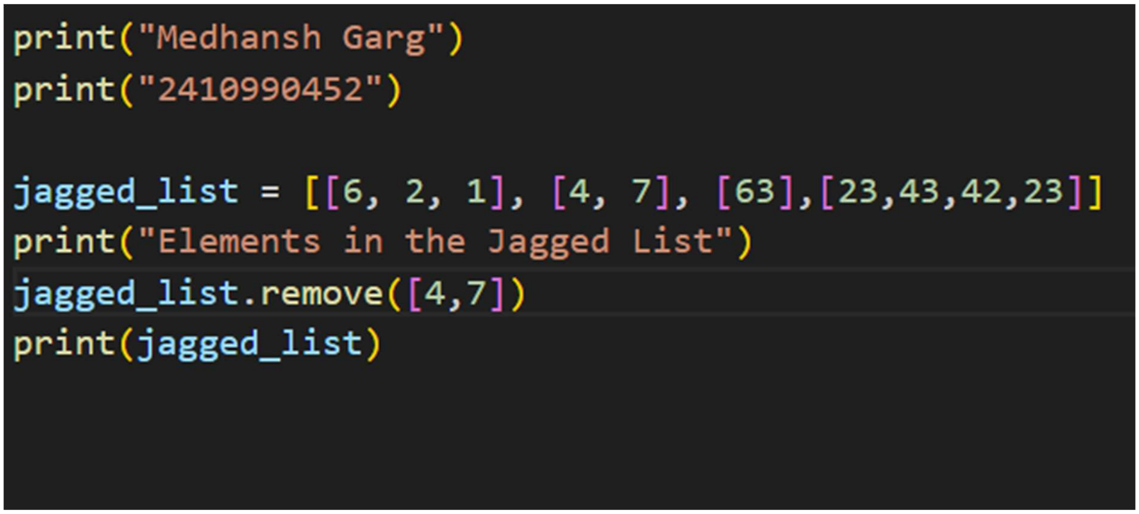
```
print("Medhansh Garg")
print("2410990452")

jagged_list = [[6, 2, 1], [4, 7], [63],[23,43,42,23]]
print("Elements in the Jagged List")
for sublist in jagged_list:
 for item in sublist:
    print(item)
```

Output:



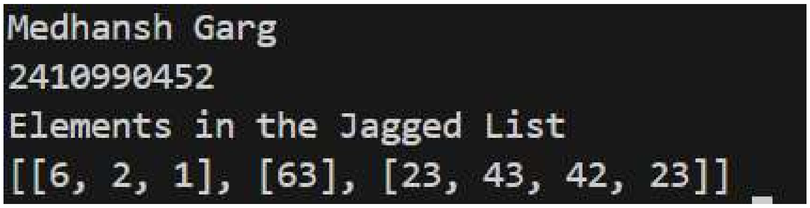
```
Medhansh Garg
2410990452
Elements in the Jagged List
6
2
1
4
7
63
23
43
42
23
```

Removing a sub list from jagged list.



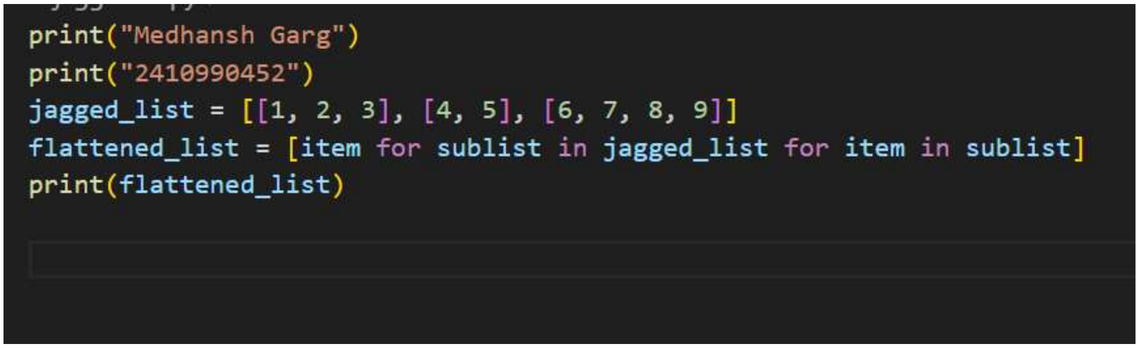
```
print("Medhansh Garg")
print("2410990452")

jagged_list = [[6, 2, 1], [4, 7], [63],[23,43,42,23]]
print("Elements in the Jagged List")
jagged_list.remove([4,7])
print(jagged_list)
```

Output:



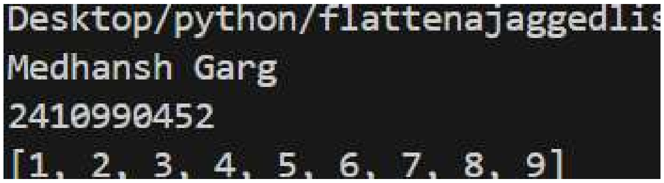
```
Medhansh Garg
2410990452
Elements in the Jagged List
[[6, 2, 1], [63], [23, 43, 42, 23]]
```

Program to flatten a jagged list.



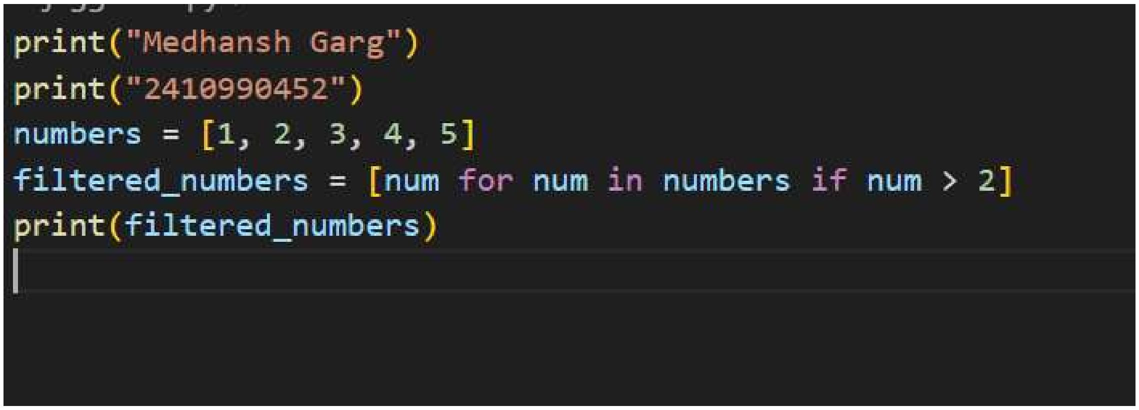
```python
print("Medhansh Garg")
print("2410990452")
jagged_list = [[1, 2, 3], [4, 5], [6, 7, 8, 9]]
flattened_list = [item for sublist in jagged_list for item in sublist]
print(flattened_list)
```

Output:



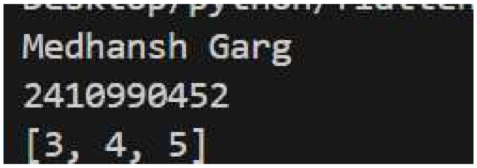
```
Desktop/python/flattenajaggedlis
Medhansh Garg
2410990452
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Filter elements greater than two elements from a list.



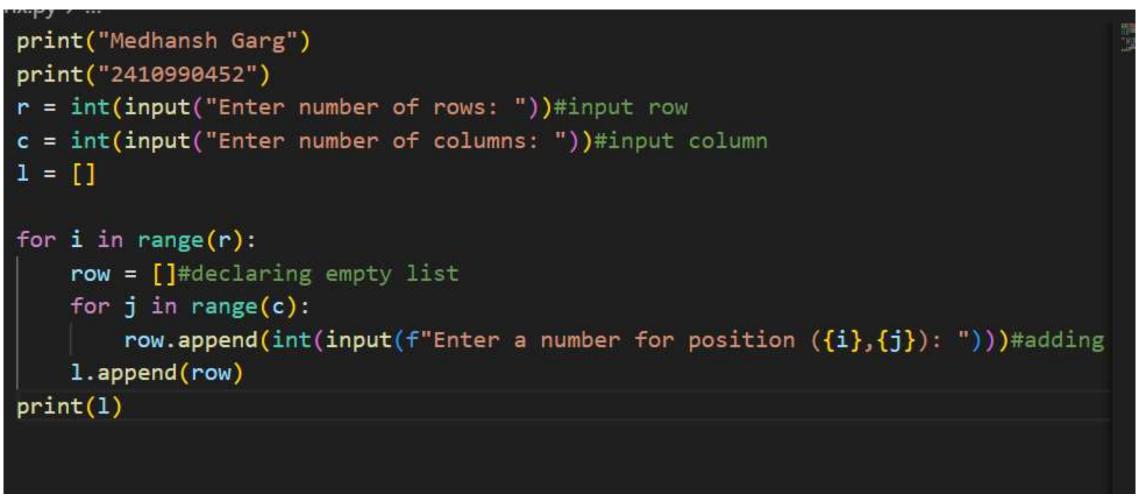
```
print("Medhansh Garg")
print("2410990452")
numbers = [1, 2, 3, 4, 5]
filtered_numbers = [num for num in numbers if num > 2]
print(filtered_numbers)
```

Output:



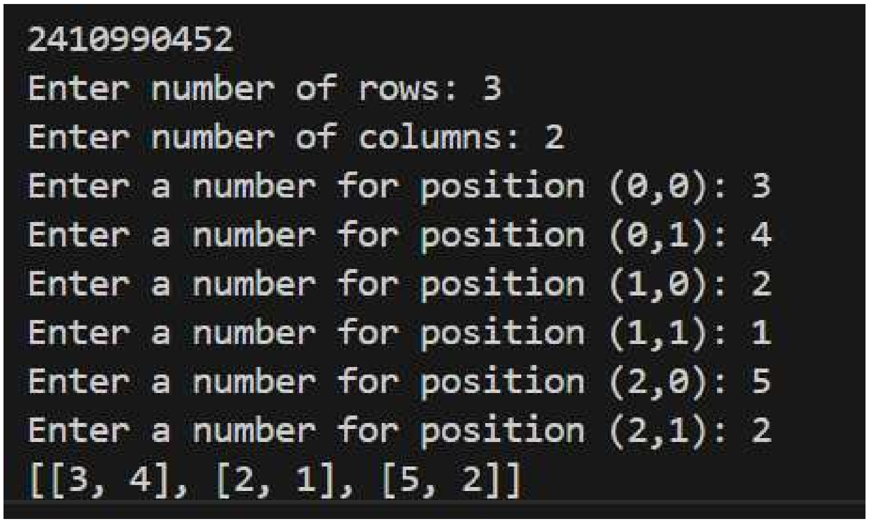
```
Medhansh Garg
2410990452
[3, 4, 5]
```

Programme to create a 2 dimension list



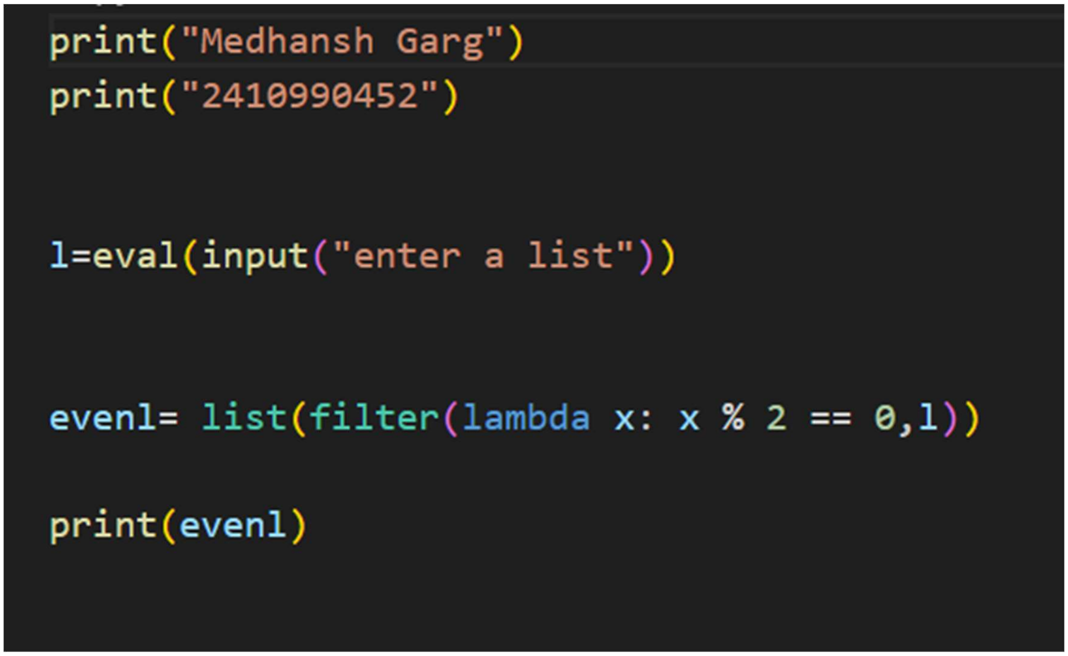
```
print("Medhansh Garg")
print("2410990452")
r = int(input("Enter number of rows: "))#input row
c = int(input("Enter number of columns: "))#input column
l = []

for i in range(r):
    row = []#declaring empty list
    for j in range(c):
        row.append(int(input(f"Enter a number for position ({i},{j}): ")))#adding
    l.append(row)
print(l)
```

Output:



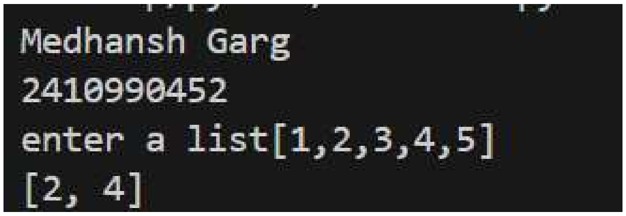
```
2410990452
Enter number of rows: 3
Enter number of columns: 2
Enter a number for position (0,0): 3
Enter a number for position (0,1): 4
Enter a number for position (1,0): 2
Enter a number for position (1,1): 1
Enter a number for position (2,0): 5
Enter a number for position (2,1): 2
[[3, 4], [2, 1], [5, 2]]
```

Programme to filter even no from list.



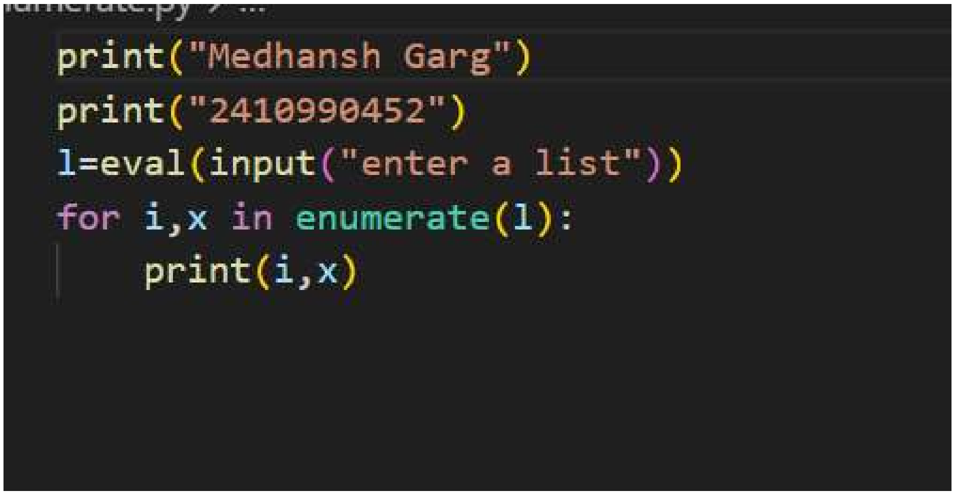
```
print("Medhansh Garg")
print("2410990452")


l=eval(input("enter a list"))


evenl= list(filter(lambda x: x % 2 == 0,l))

print(evenl)
```

Output:



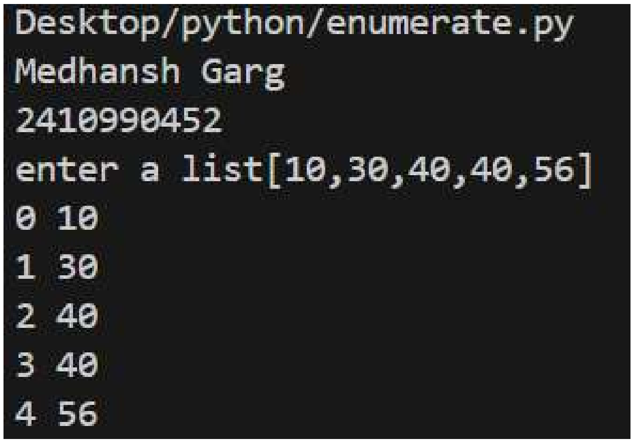
```
Medhansh Garg
2410990452
enter a list[1,2,3,4,5]
[2, 4]
```

Programme to print index and elements using enumerate in list.



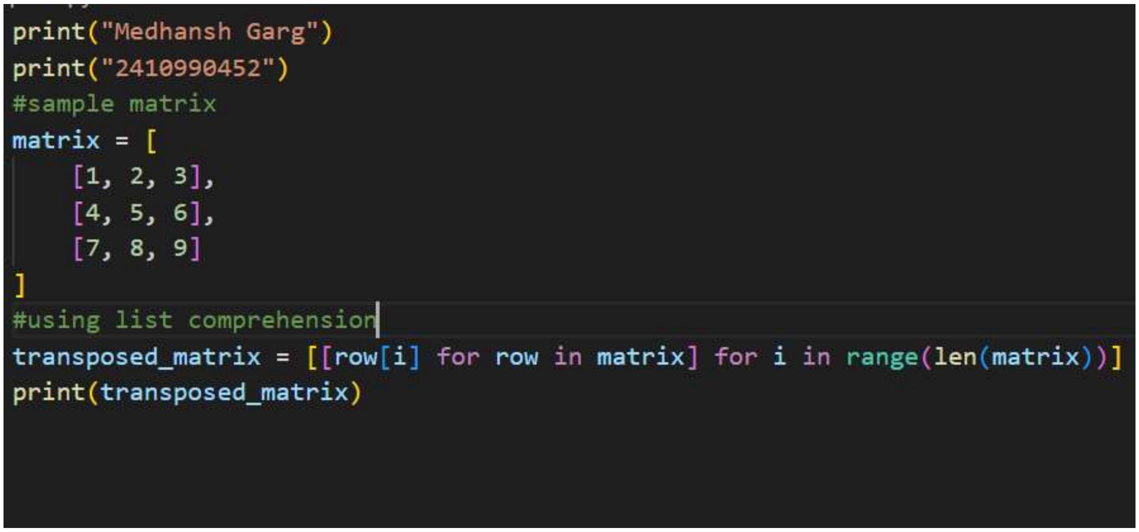
```python
print("Medhansh Garg")
print("2410990452")
l=eval(input("enter a list"))
for i,x in enumerate(l):
    print(i,x)
```

Output:



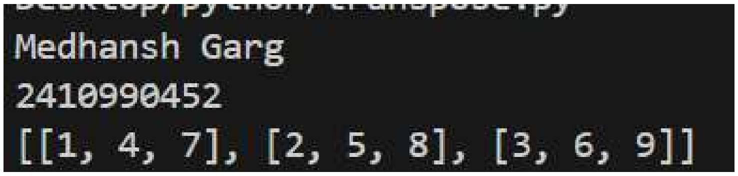
```
Desktop/python/enumerate.py
Medhansh Garg
2410990452
enter a list[10,30,40,40,56]
0 10
1 30
2 40
3 40
4 56
```

Transpose a 3*3 matrix using list comprehension.



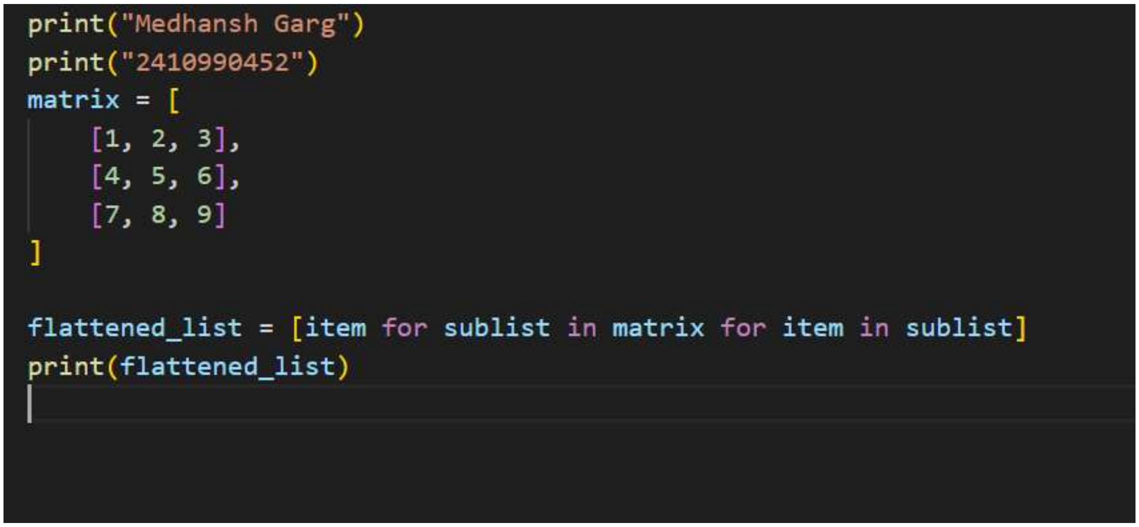
```python
print("Medhansh Garg")
print("2410990452")
#sample matrix
matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
#using list comprehension
transposed_matrix = [[row[i] for row in matrix] for i in range(len(matrix))]
print(transposed_matrix)
```

Output:



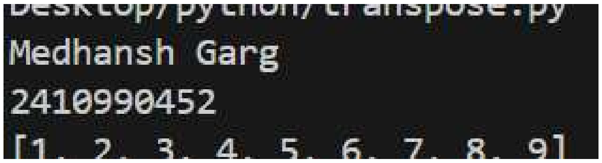
```
Desktop/python/transpose.py
Medhansh Garg
2410990452
[[1, 4, 7], [2, 5, 8], [3, 6, 9]]
```

Flatten a 2-d list to single list using list comprehension.



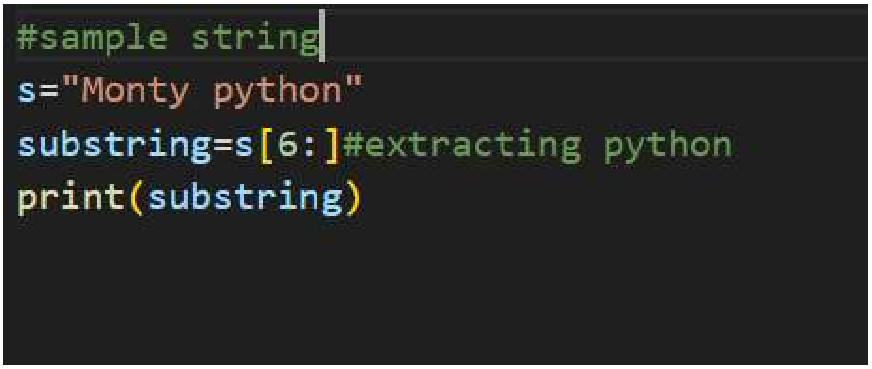
```python
print("Medhansh Garg")
print("2410990452")
matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]

flattened_list = [item for sublist in matrix for item in sublist]
print(flattened_list)
```

Output:



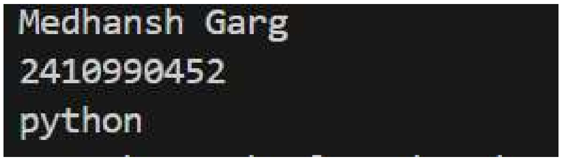
```
Desktop/python/transpose.py
Medhansh Garg
2410990452
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Program to slice a string to get a substring.



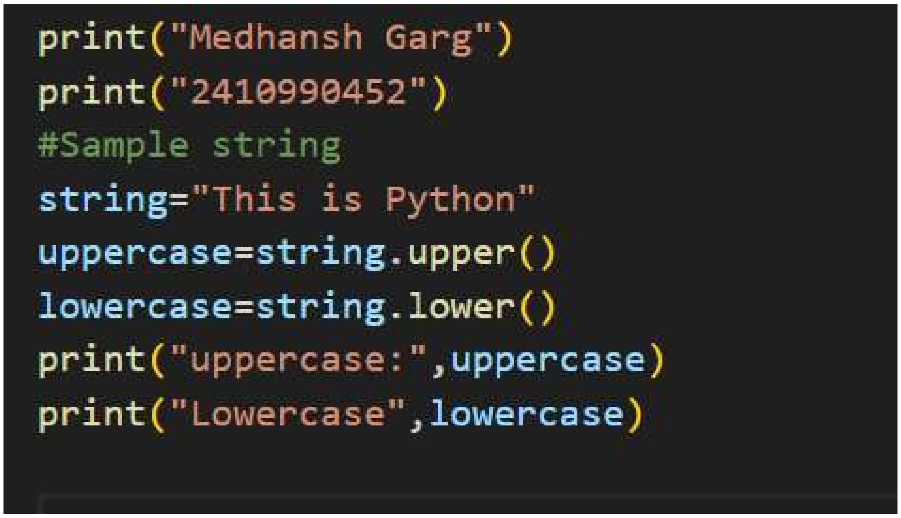
```
#sample string
s="Monty python"
substring=s[6:]#extracting python
print(substring)
```

Output:



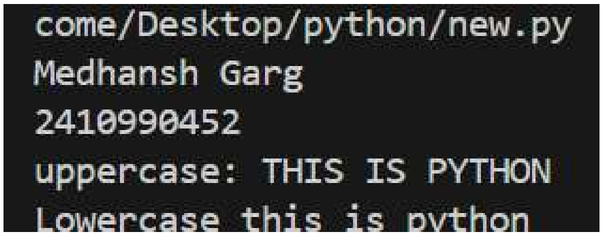
```
Medhansh Garg
2410990452
python
```

Program to convert a string to upper and lower case.



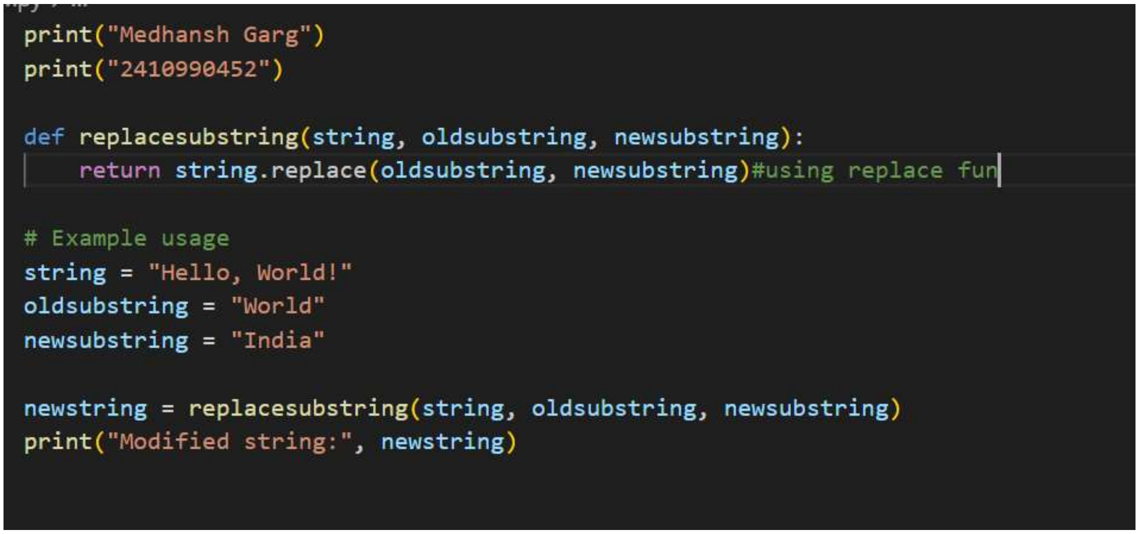
```python
print("Medhansh Garg")
print("2410990452")
#Sample string
string="This is Python"
uppercase=string.upper()
lowercase=string.lower()
print("uppercase:",uppercase)
print("Lowercase",lowercase)
```

Output:



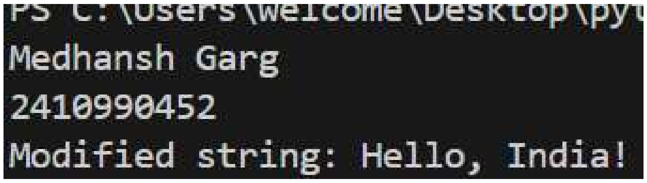
```
come/Desktop/python/new.py
Medhansh Garg
2410990452
uppercase: THIS IS PYTHON
Lowercase this is python
```

Program to replace a substring in a string.



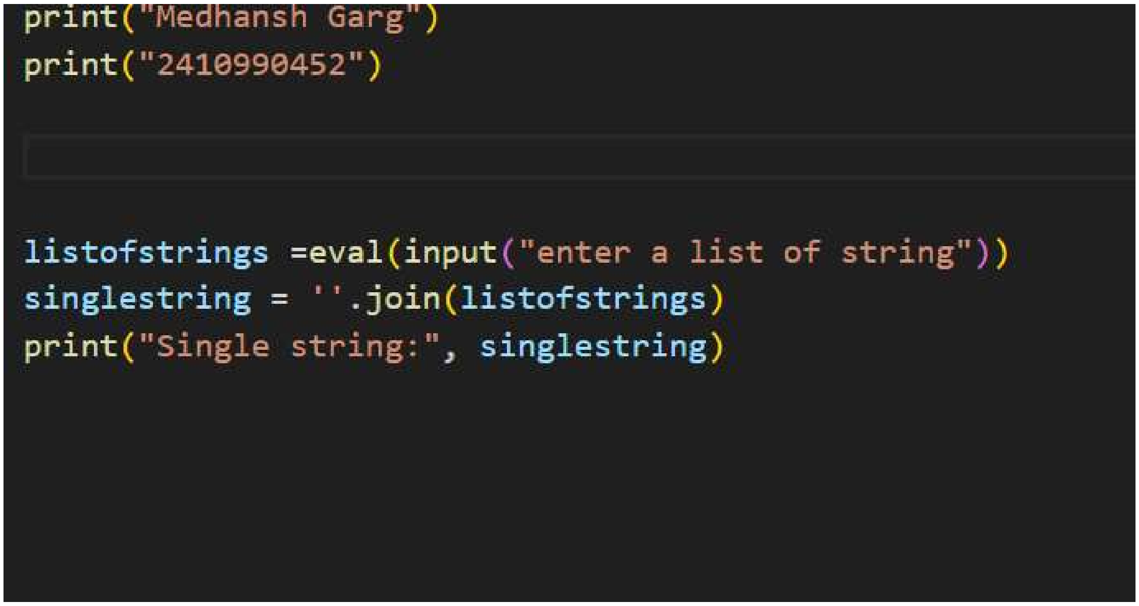
```
print("Medhansh Garg")
print("2410990452")

def replacesubstring(string, oldsubstring, newsubstring):
    return string.replace(oldsubstring, newsubstring)#using replace fun

# Example usage
string = "Hello, World!"
oldsubstring = "World"
newsubstring = "India"

newstring = replacesubstring(string, oldsubstring, newsubstring)
print("Modified string:", newstring)
```

Output:



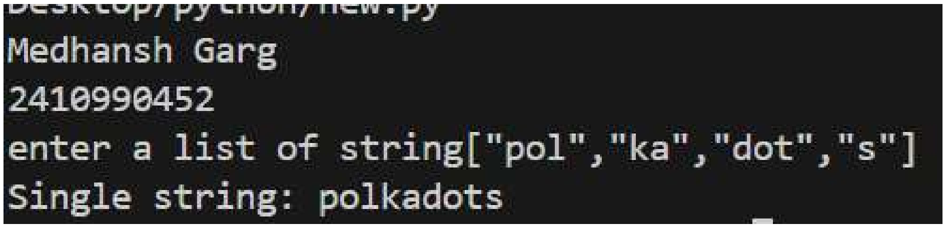
```
PS C:\Users\welcome\Desktop\pyt
Medhansh Garg
2410990452
Modified string: Hello, India!
```

Program to join a list of string to single string



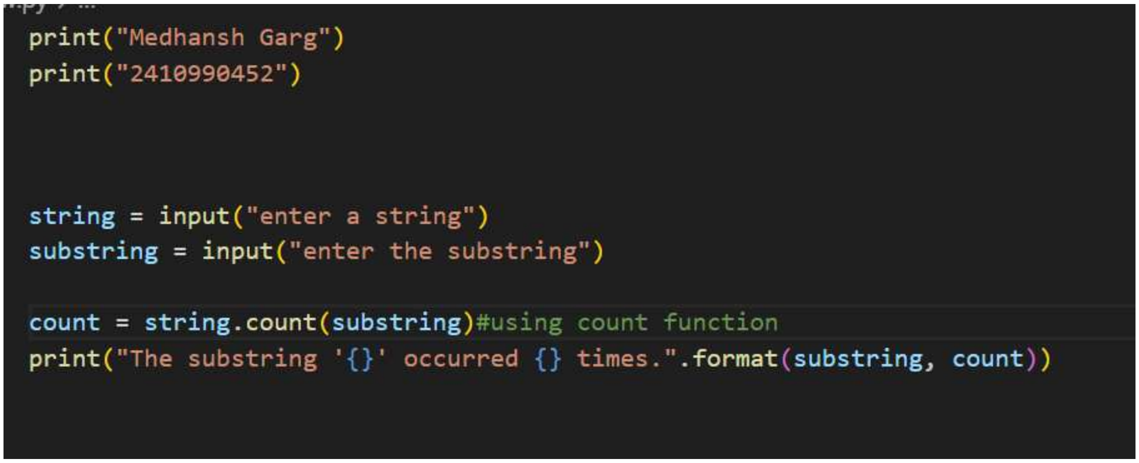
```python
print("Medhansh Garg")
print("2410990452")


listofstrings =eval(input("enter a list of string"))
singlestring = ''.join(listofstrings)
print("Single string:", singlestring)
```

Output:



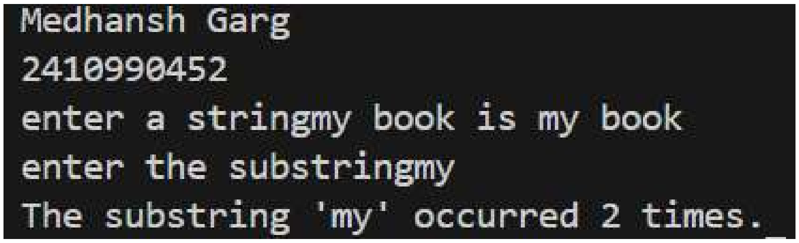
```
Desktop/python/new.py
Medhansh Garg
2410990452
enter a list of string["pol","ka","dot","s"]
Single string: polkadots
```

Program to count occurrences of substring in string



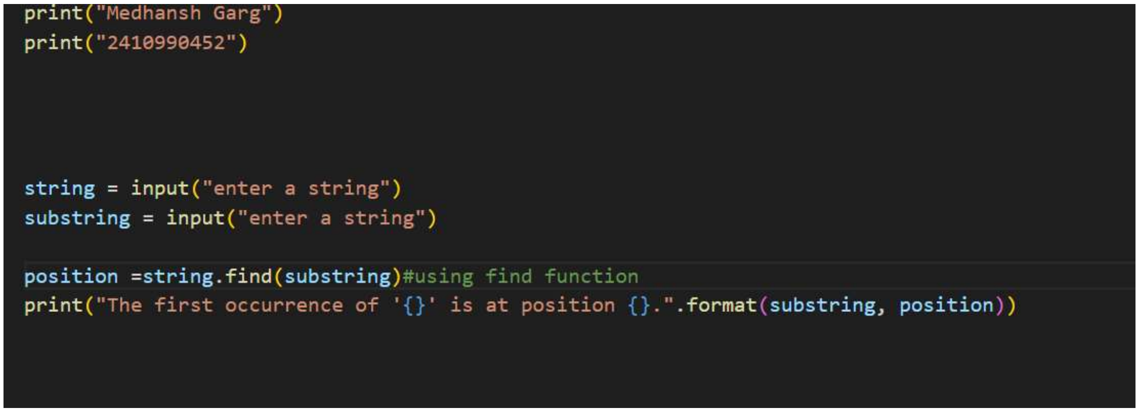
```python
print("Medhansh Garg")
print("2410990452")



string = input("enter a string")
substring = input("enter the substring")

count = string.count(substring)#using count function
print("The substring '{}' occurred {} times.".format(substring, count))
```

Output:



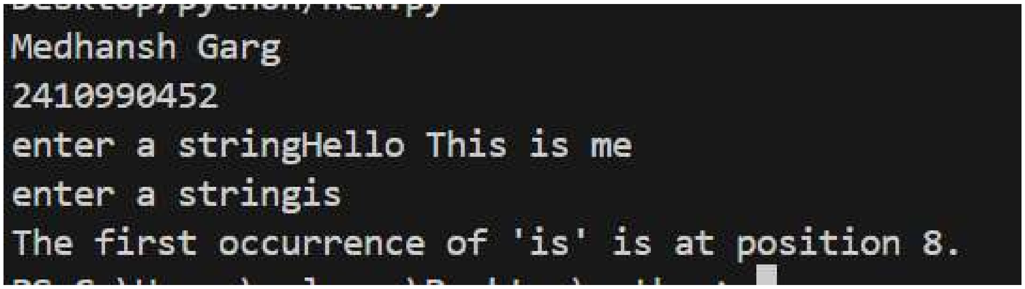
```
Medhansh Garg
2410990452
enter a stringmy book is my book
enter the substringmy
The substring 'my' occurred 2 times.
```

Program to find the position of substring in string.



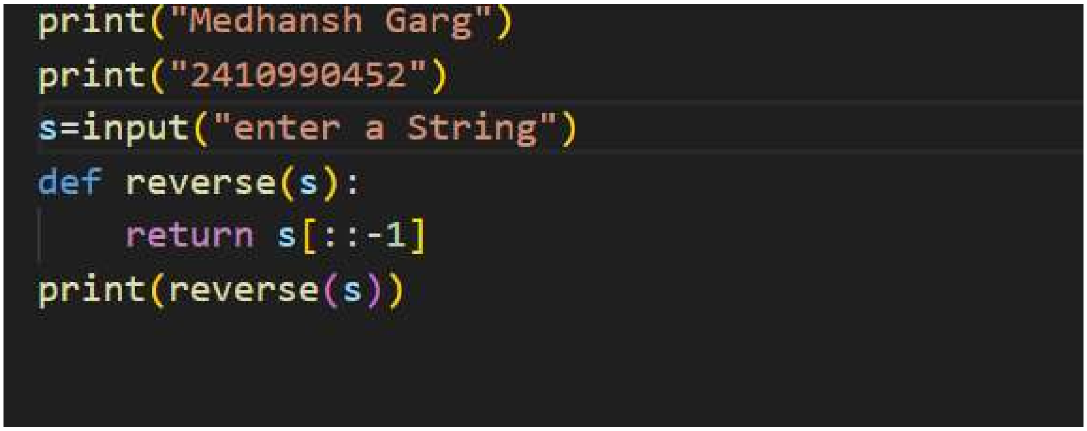
```
print("Medhansh Garg")
print("2410990452")



string = input("enter a string")
substring = input("enter a string")

position =string.find(substring)#using find function
print("The first occurrence of '{}' is at position {}.".format(substring, position))
```

Output:



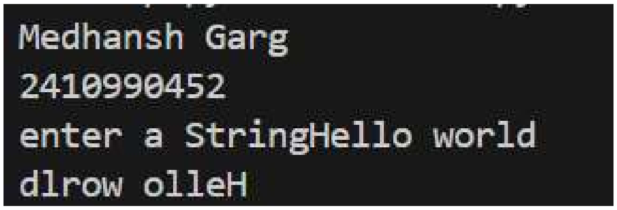
```
Desktop/python/new.py
Medhansh Garg
2410990452
enter a stringHello This is me
enter a stringis
The first occurrence of 'is' is at position 8.
```

Program to reverse string using slicing.



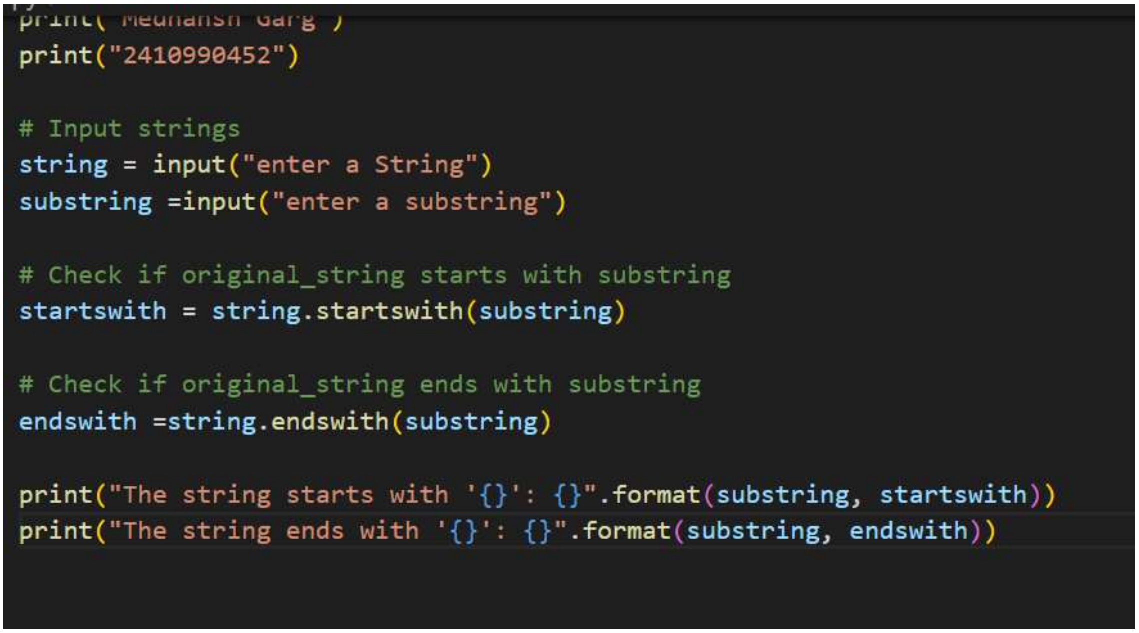
```
print("Medhansh Garg")
print("2410990452")
s=input("enter a String")
def reverse(s):
    return s[::-1]
print(reverse(s))
```

Output:



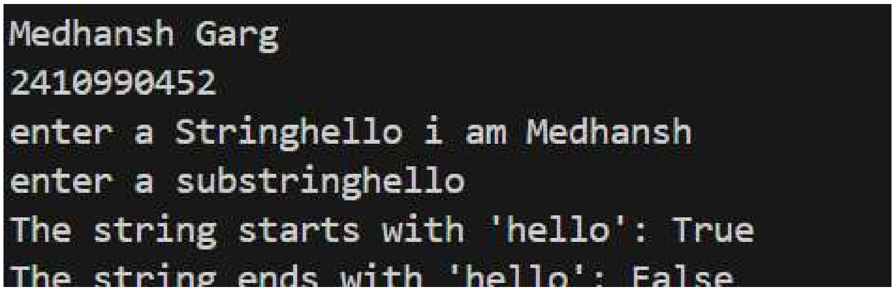
```
Medhansh Garg
2410990452
enter a StringHello world
dlrow olleH
```

Program to check if string starts with or ends with a substring.



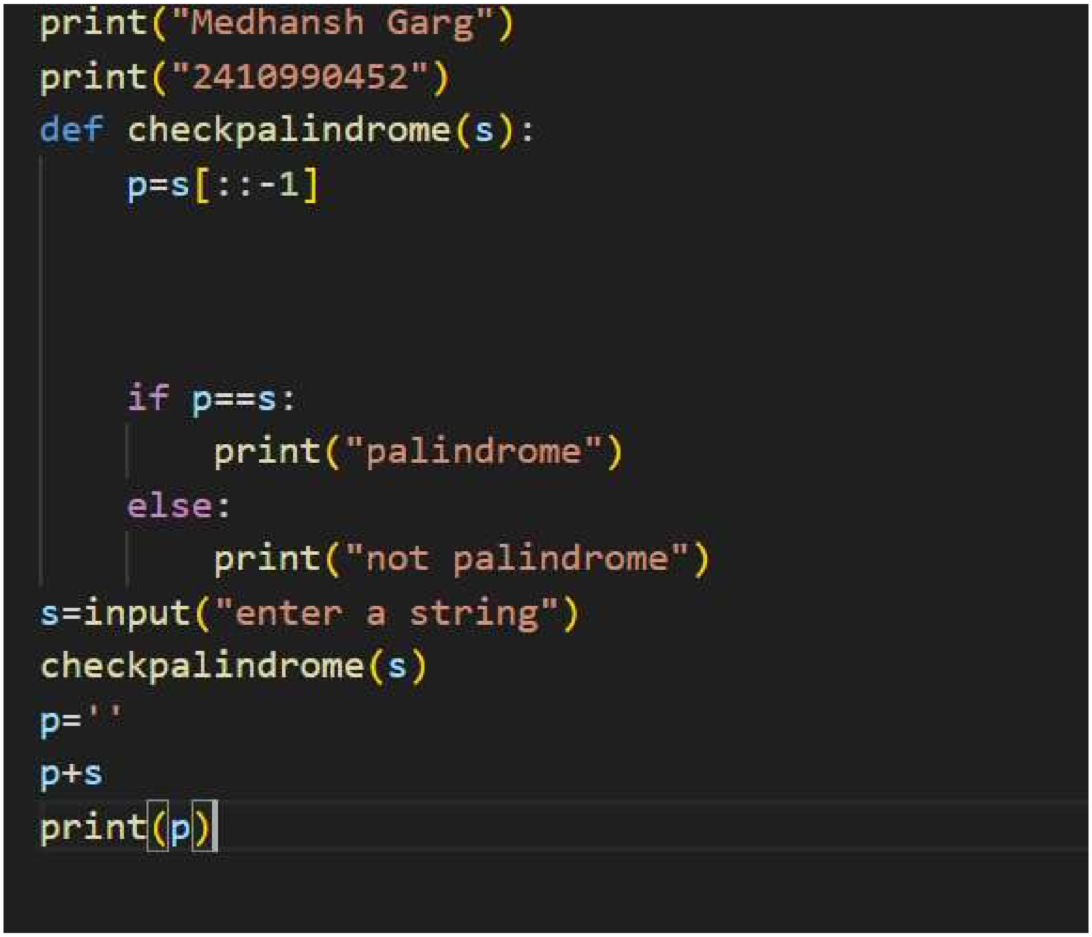
```
print("Medhansh Garg")
print("2410990452")

# Input strings
string = input("enter a String")
substring =input("enter a substring")

# Check if original_string starts with substring
startswith = string.startswith(substring)

# Check if original_string ends with substring
endswith =string.endswith(substring)

print("The string starts with '{}': {}".format(substring, startswith))
print("The string ends with '{}': {}".format(substring, endswith))
```

Output:



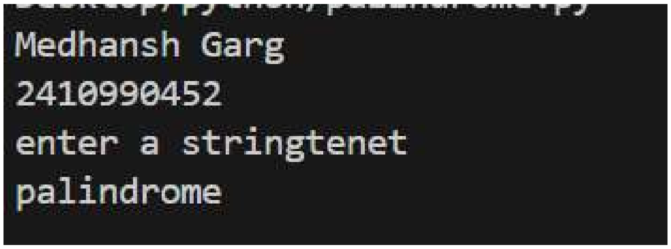
```
Medhansh Garg
2410990452
enter a Stringhello i am Medhansh
enter a substringhello
The string starts with 'hello': True
The string ends with 'hello': False
```

Program to check if a string is palindrome.



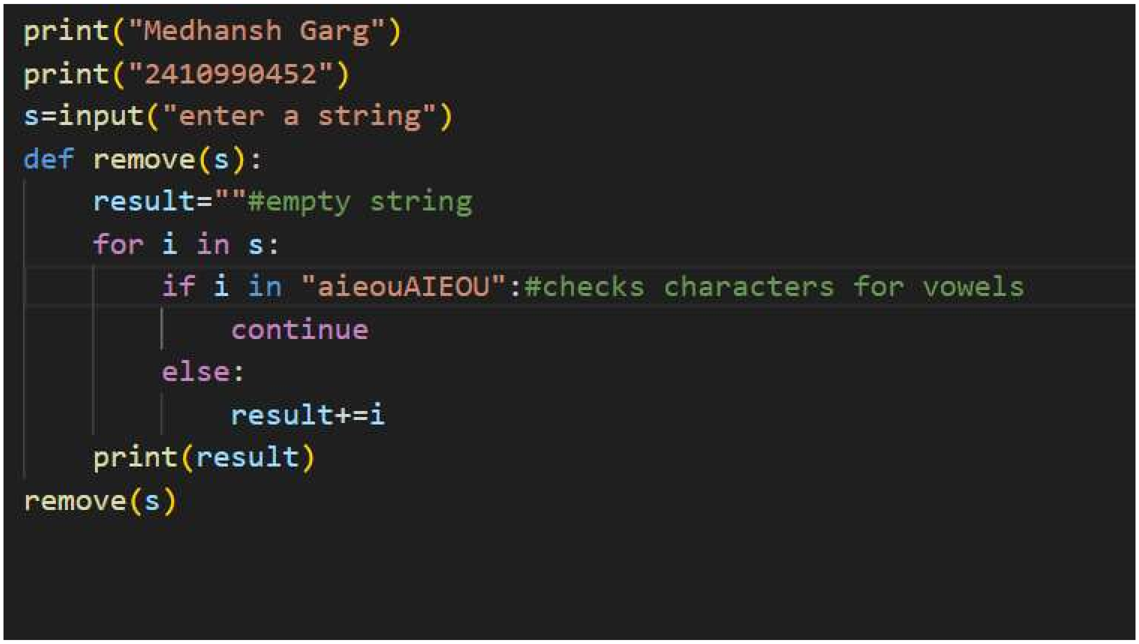
```
print("Medhansh Garg")
print("2410990452")
def checkpalindrome(s):
    p=s[::-1]



    if p==s:
        print("palindrome")
    else:
        print("not palindrome")
s=input("enter a string")
checkpalindrome(s)
p=''
p+s
print(p)
```

Output:



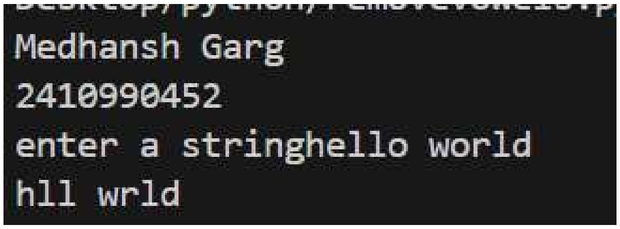
```
Medhansh Garg
2410990452
enter a stringtenet
palindrome
```

Programme to remove vowels from a string



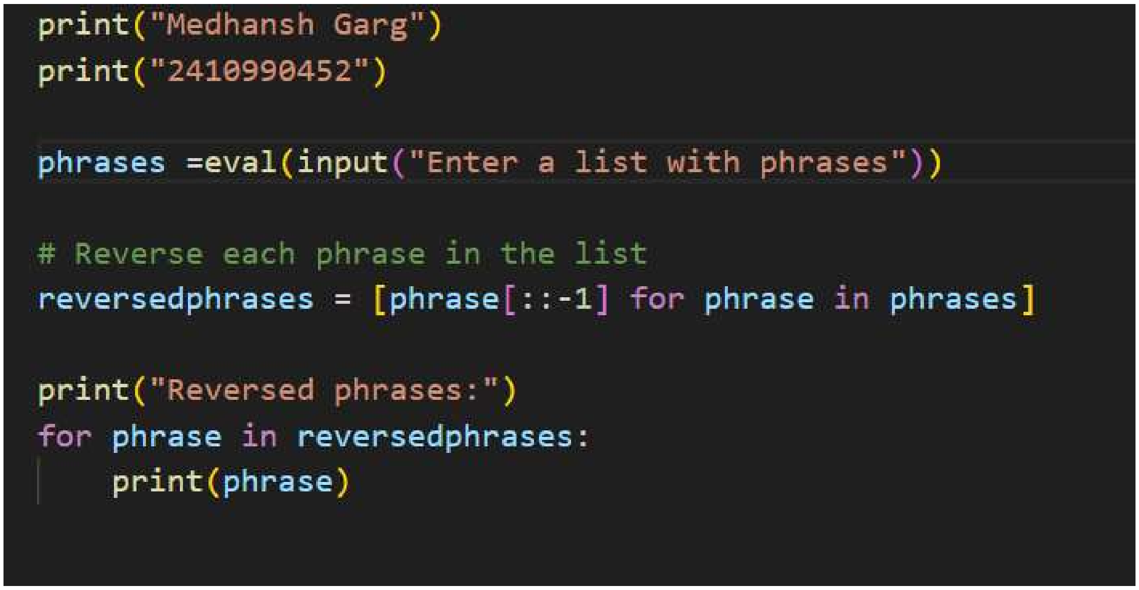
```python
print("Medhansh Garg")
print("2410990452")
s=input("enter a string")
def remove(s):
    result=""#empty string
    for i in s:
        if i in "aieouAIEOU":#checks characters for vowels
            continue
        else:
            result+=i
    print(result)
remove(s)
```

Output:



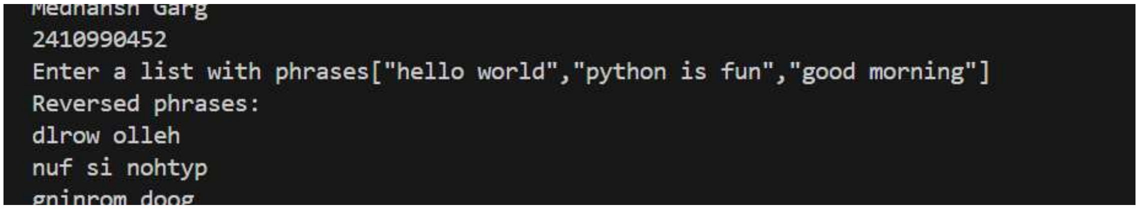
```
Medhansh Garg
2410990452
enter a stringhello world
hll wrld
```

Programme to reverse each string in a list of phrases.



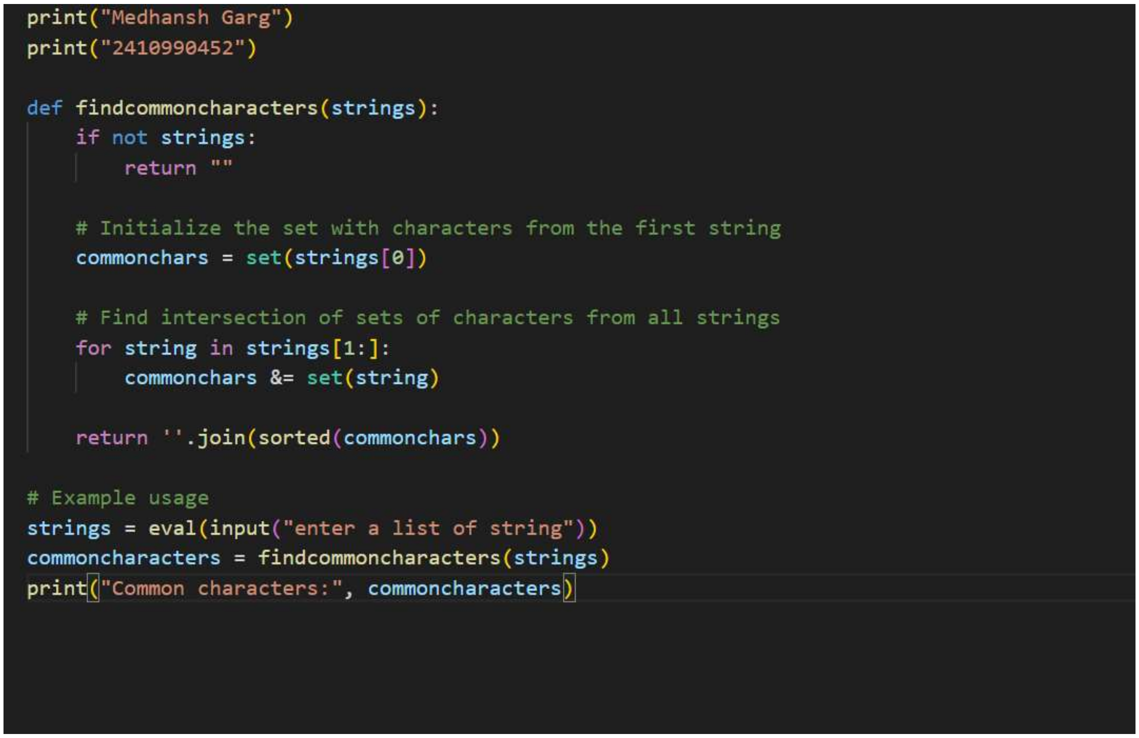
```
print("Medhansh Garg")
print("2410990452")

phrases =eval(input("Enter a list with phrases"))

# Reverse each phrase in the list
reversedphrases = [phrase[::-1] for phrase in phrases]

print("Reversed phrases:")
for phrase in reversedphrases:
    print(phrase)
```

Output:



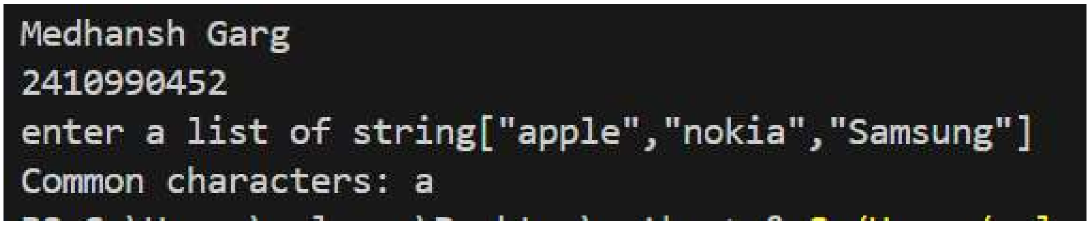
```
Medhansh Garg
2410990452
Enter a list with phrases["hello world","python is fun","good morning"]
Reversed phrases:
dlrow olleh
nuf si nohtyp
gninrom doog
```

To find the common characters in a list of string



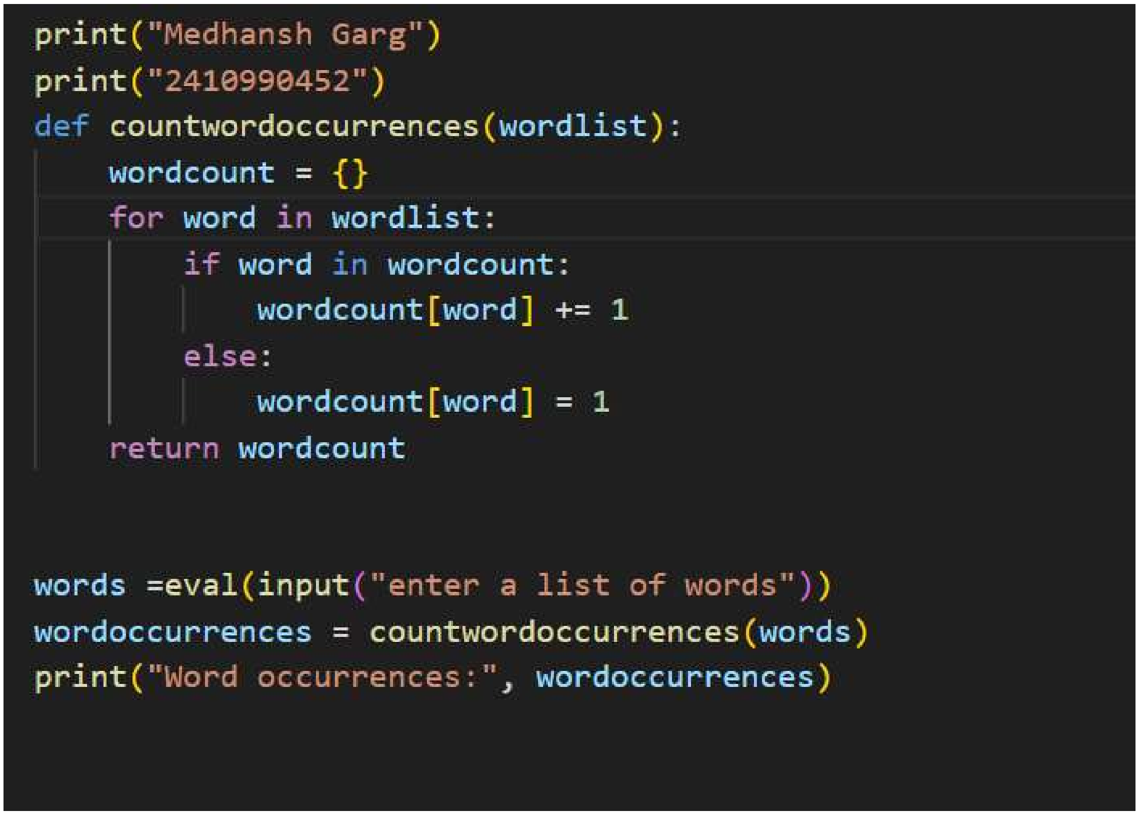
```python
print("Medhansh Garg")
print("2410990452")

def findcommoncharacters(strings):
    if not strings:
        return ""

    # Initialize the set with characters from the first string
    commonchars = set(strings[0])

    # Find intersection of sets of characters from all strings
    for string in strings[1:]:
        commonchars &= set(string)

    return ''.join(sorted(commonchars))

# Example usage
strings = eval(input("enter a list of string"))
commoncharacters = findcommoncharacters(strings)
print("Common characters:", commoncharacters)
```

Output:



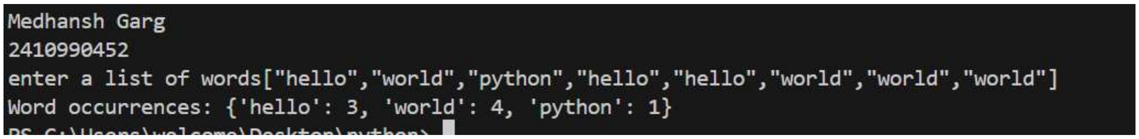
```
Medhansh Garg
2410990452
enter a list of string["apple","nokia","Samsung"]
Common characters: a
```

Count the occurrence of each word in a list



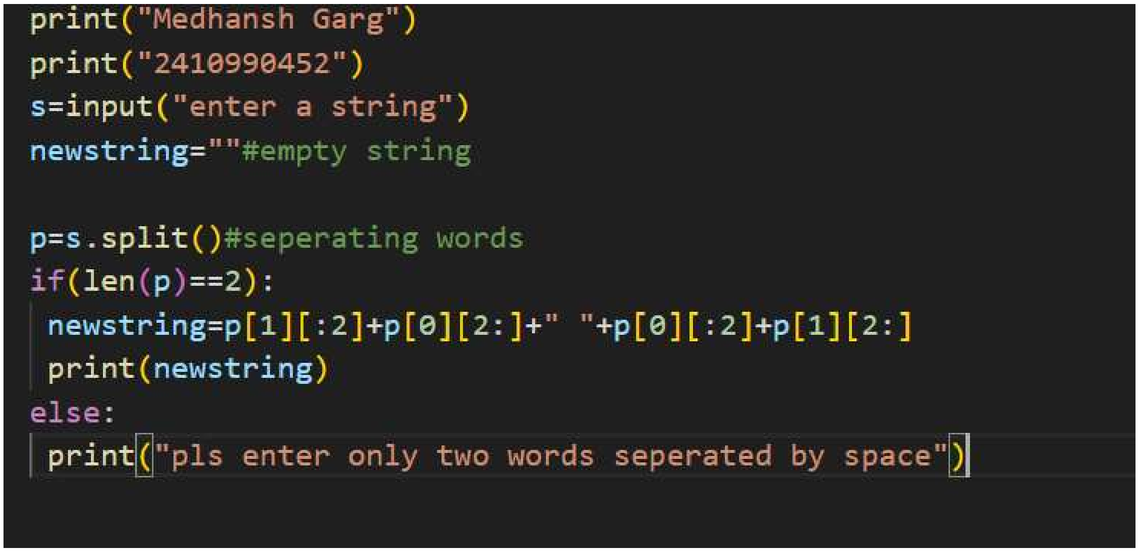
```
print("Medhansh Garg")
print("2410990452")
def countwordoccurrences(wordlist):
    wordcount = {}
    for word in wordlist:
        if word in wordcount:
            wordcount[word] += 1
        else:
            wordcount[word] = 1
    return wordcount


words =eval(input("enter a list of words"))
wordoccurrences = countwordoccurrences(words)
print("Word occurrences:", wordoccurrences)
```

Output:



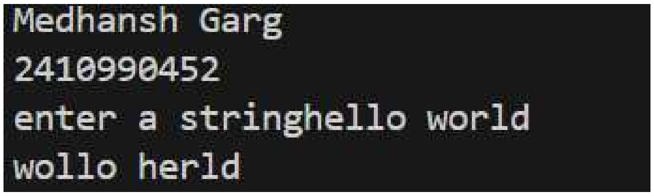
```
Medhansh Garg
2410990452
enter a list of words["hello","world","python","hello","hello","world","world","world"]
Word occurrences: {'hello': 3, 'world': 4, 'python': 1}
PS C:\Users\welcome\Desktop\python>
```

Take a string with two words and then first two letters.



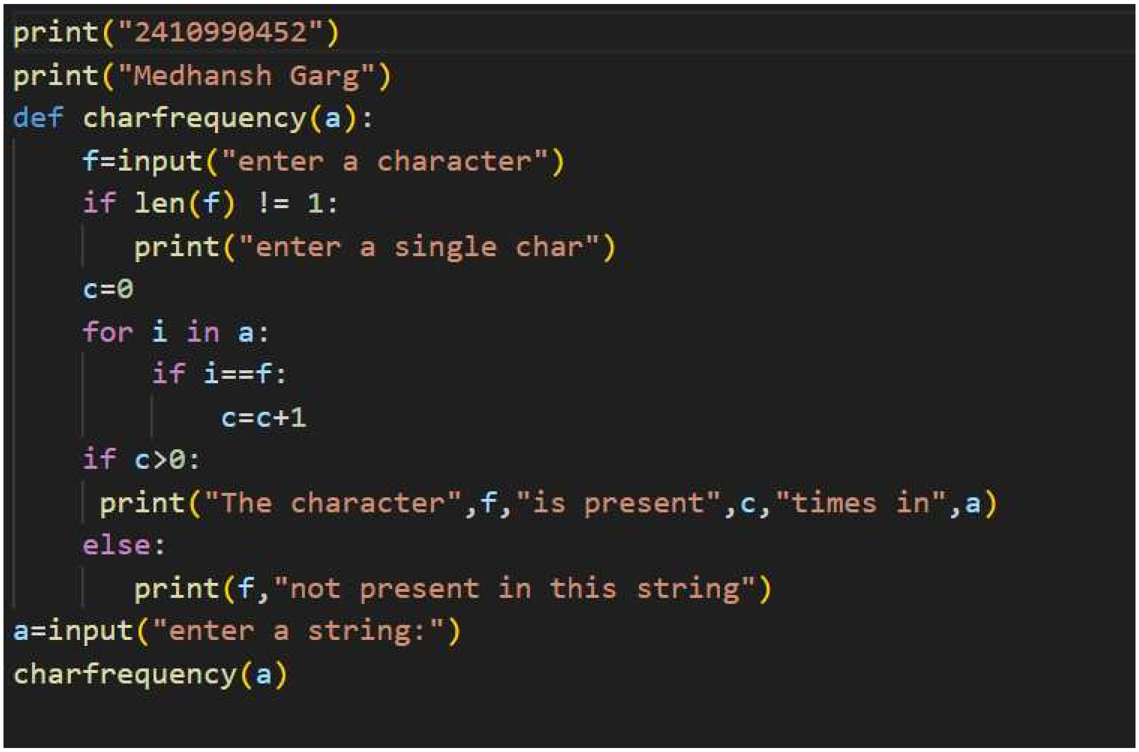
```
print("Medhansh Garg")
print("2410990452")
s=input("enter a string")
newstring=""#empty string

p=s.split()#seperating words
if(len(p)==2):
    newstring=p[1][:2]+p[0][2:]+" "+p[0][:2]+p[1][2:]
    print(newstring)
else:
    print("pls enter only two words seperated by space")
```

Output:



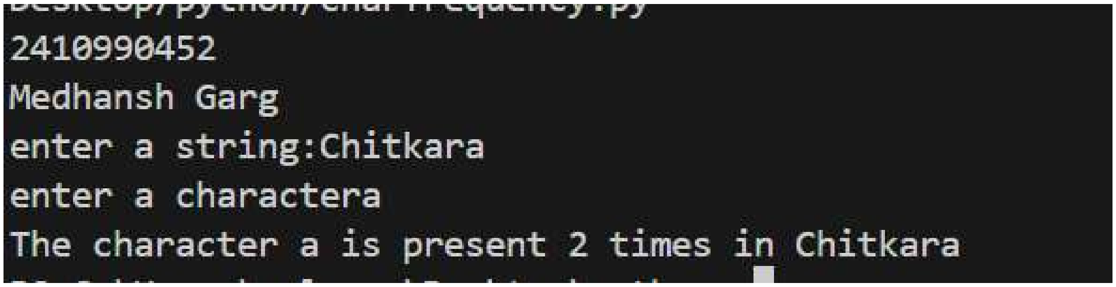
```
Medhansh Garg
2410990452
enter a stringhello world
wollo herld
```

To find the frequency of given character in string.



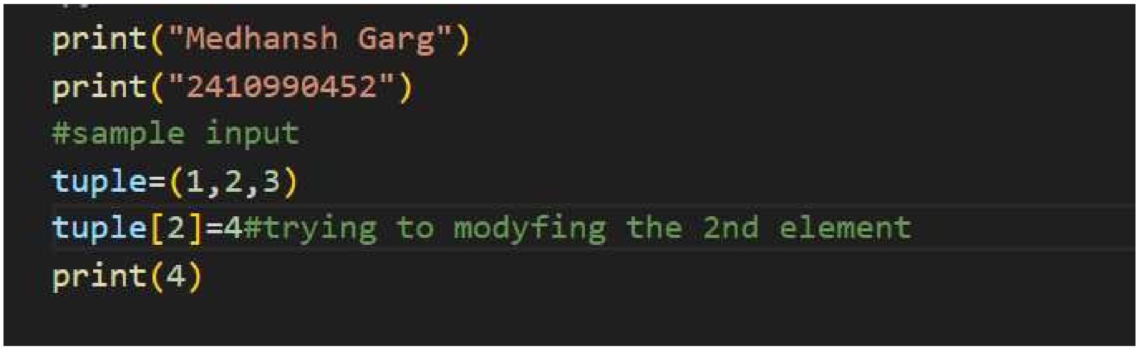
```
print("2410990452")
print("Medhansh Garg")
def charfrequency(a):
    f=input("enter a character")
    if len(f) != 1:
       print("enter a single char")
    c=0
    for i in a:
        if i==f:
            c=c+1
    if c>0:
     print("The character",f,"is present",c,"times in",a)
    else:
       print(f,"not present in this string")
a=input("enter a string:")
charfrequency(a)
```

Output:



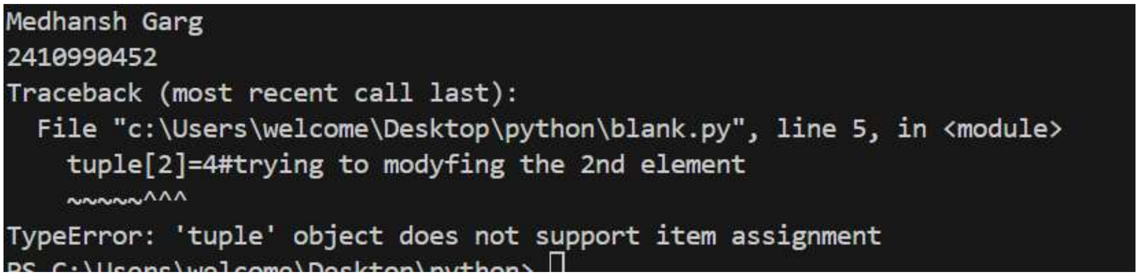
```
Desktop/python/charfrequency.py
2410990452
Medhansh Garg
enter a string:Chitkara
enter a charactera
The character a is present 2 times in Chitkara
```

Demonstrate Tuples are immutable



```
print("Medhansh Garg")
print("2410990452")
#sample input
tuple=(1,2,3)
tuple[2]=4#trying to modyfing the 2nd element
print(4)
```

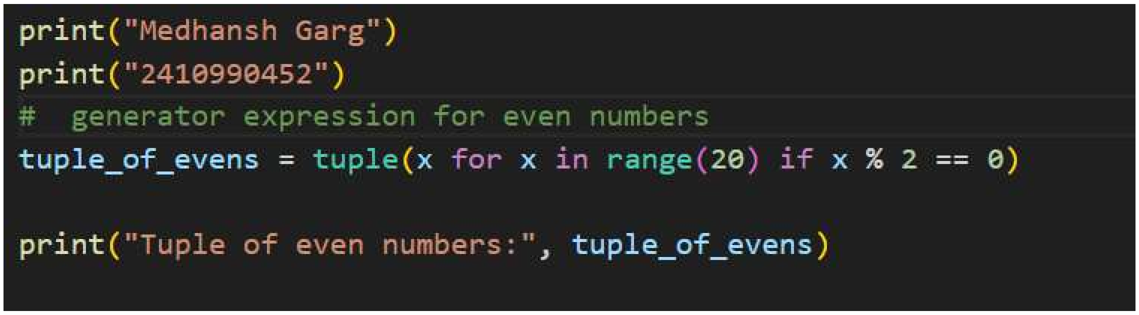
Output:



```
Medhansh Garg
2410990452
Traceback (most recent call last):
  File "c:\Users\welcome\Desktop\python\blank.py", line 5, in <module>
    tuple[2]=4#trying to modyfing the 2nd element
    ~~~~~^^^
TypeError: 'tuple' object does not support item assignment
PS C:\Users\welcome\Desktop\python> 
```

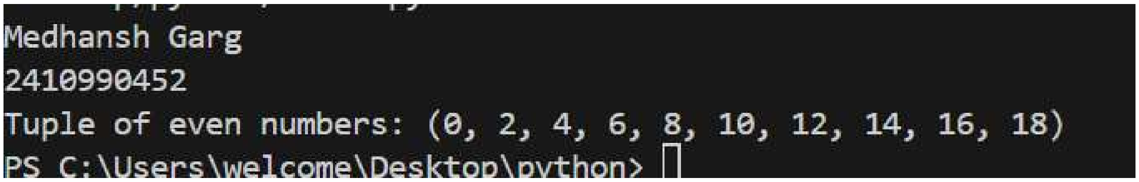
This shows error as tuple is immutable

Create a tuple using a generator expression



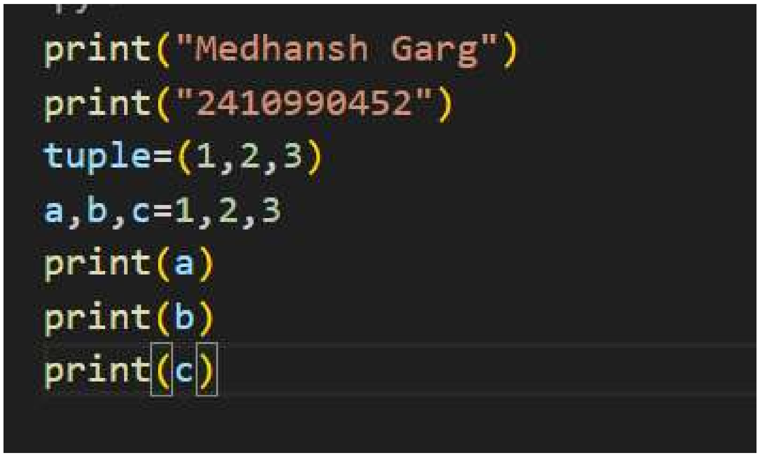
```python
print("Medhansh Garg")
print("2410990452")
#  generator expression for even numbers
tuple_of_evens = tuple(x for x in range(20) if x % 2 == 0)

print("Tuple of even numbers:", tuple_of_evens)
```

Output:



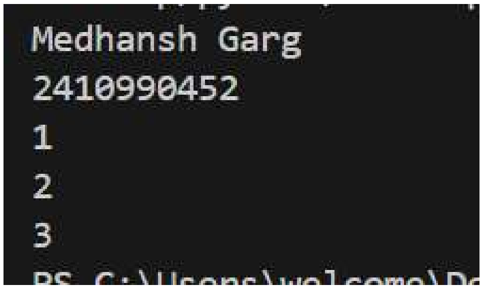
```
Medhansh Garg
2410990452
Tuple of even numbers: (0, 2, 4, 6, 8, 10, 12, 14, 16, 18)
PS C:\Users\welcome\Desktop\python> 
```

Unpacking a tuple



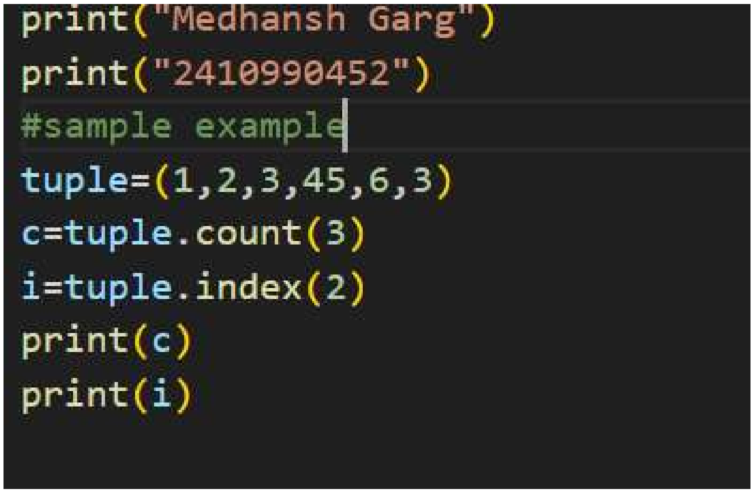
```
print("Medhansh Garg")
print("2410990452")
tuple=(1,2,3)
a,b,c=1,2,3
print(a)
print(b)
print(c)
```

Output:



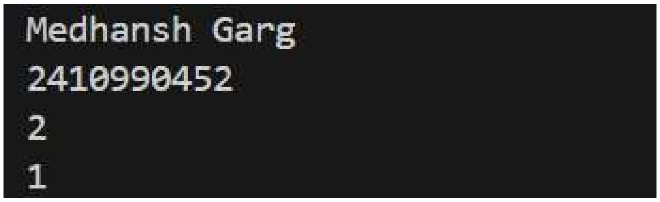
```
Medhansh Garg
2410990452
1
2
3
PS C:\Users\welcome\De
```

Use tuple methods count and index



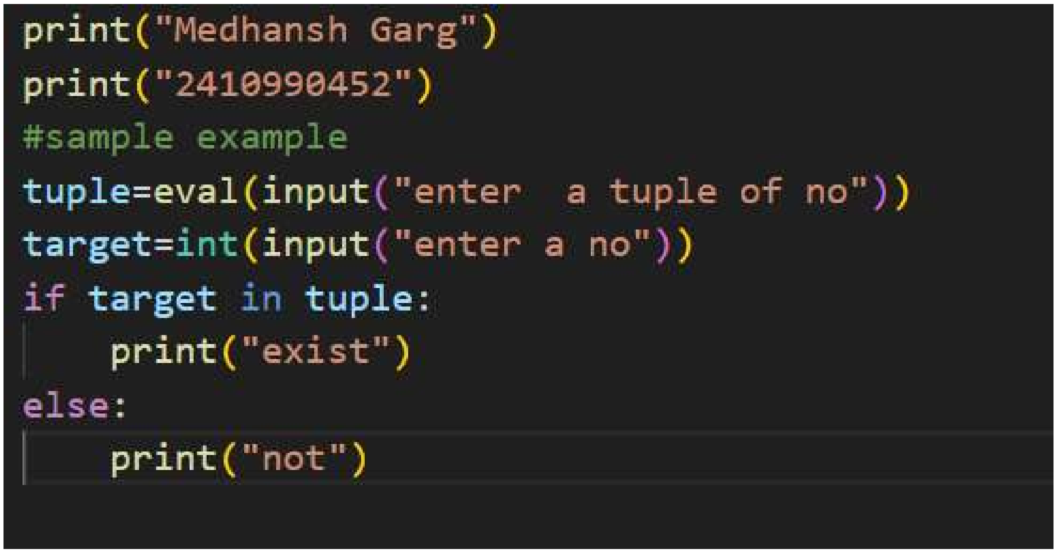
```
print("Medhansh Garg")
print("2410990452")
#sample example
tuple=(1,2,3,45,6,3)
c=tuple.count(3)
i=tuple.index(2)
print(c)
print(i)
```

Output:



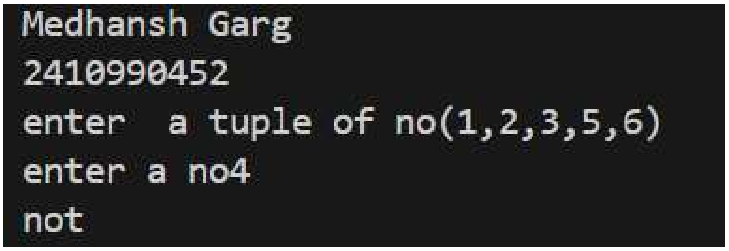
```
Medhansh Garg
2410990452
2
1
```

Check whether element is present in tuple or not



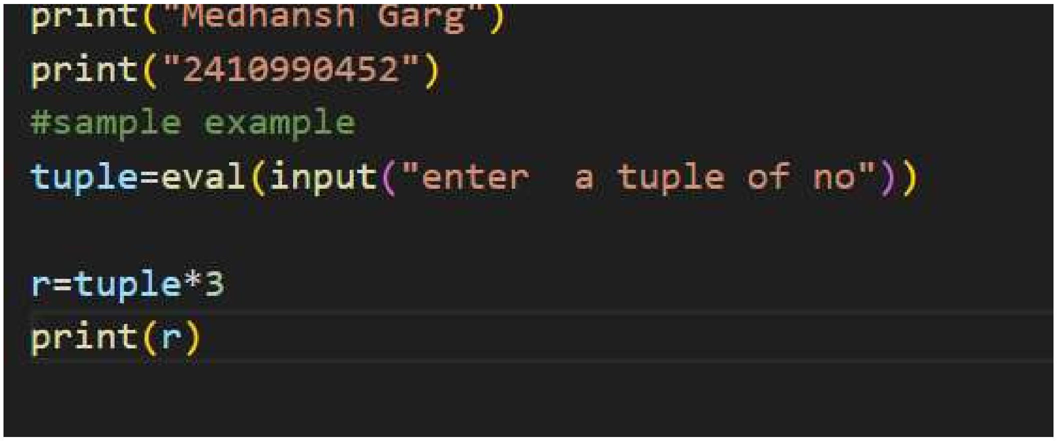
```
print("Medhansh Garg")
print("2410990452")
#sample example
tuple=eval(input("enter  a tuple of no"))
target=int(input("enter a no"))
if target in tuple:
    print("exist")
else:
    print("not")
```

Output:



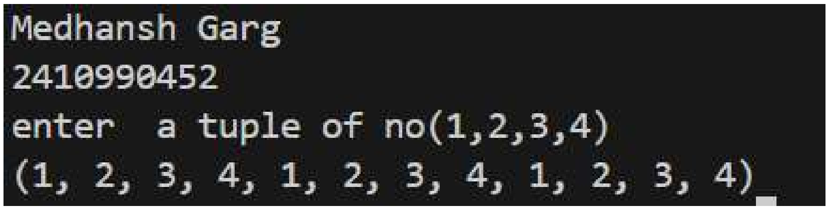
```
Medhansh Garg
2410990452
enter  a tuple of no(1,2,3,5,6)
enter a no4
not
```

Repeat a multiple times



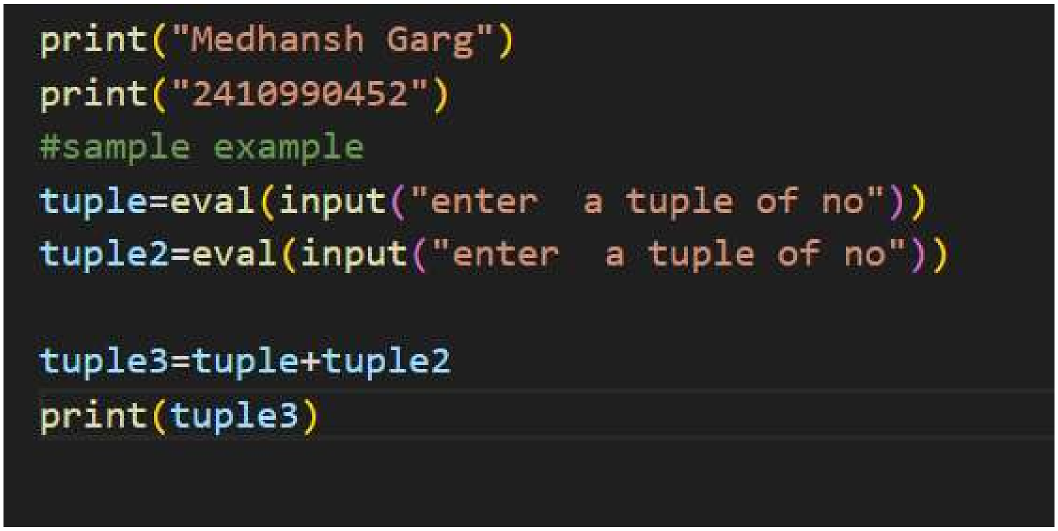
```
print("Medhansh Garg")
print("2410990452")
#sample example
tuple=eval(input("enter  a tuple of no"))

r=tuple*3
print(r)
```

Output:



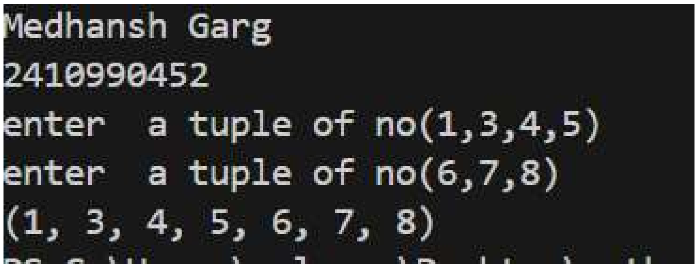
```
Medhansh Garg
2410990452
enter  a tuple of no(1,2,3,4)
(1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4)
```

Concatenate two tuples



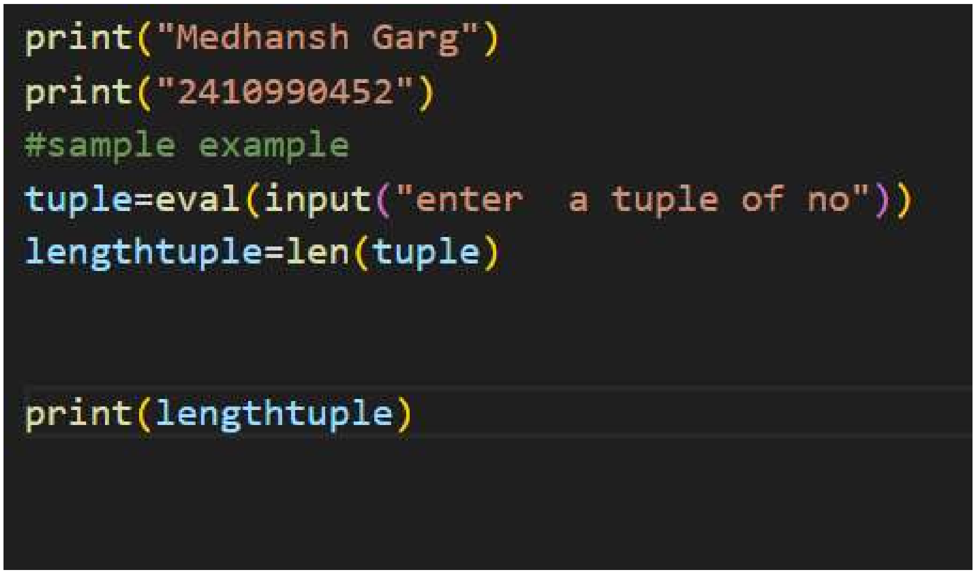
```
print("Medhansh Garg")
print("2410990452")
#sample example
tuple=eval(input("enter  a tuple of no"))
tuple2=eval(input("enter  a tuple of no"))

tuple3=tuple+tuple2
print(tuple3)
```

Output:



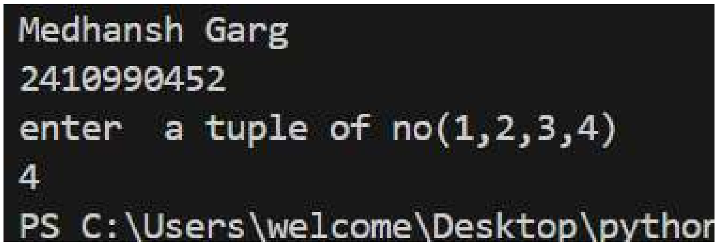
```
Medhansh Garg
2410990452
enter  a tuple of no(1,3,4,5)
enter  a tuple of no(6,7,8)
(1, 3, 4, 5, 6, 7, 8)
```

Length of tuple



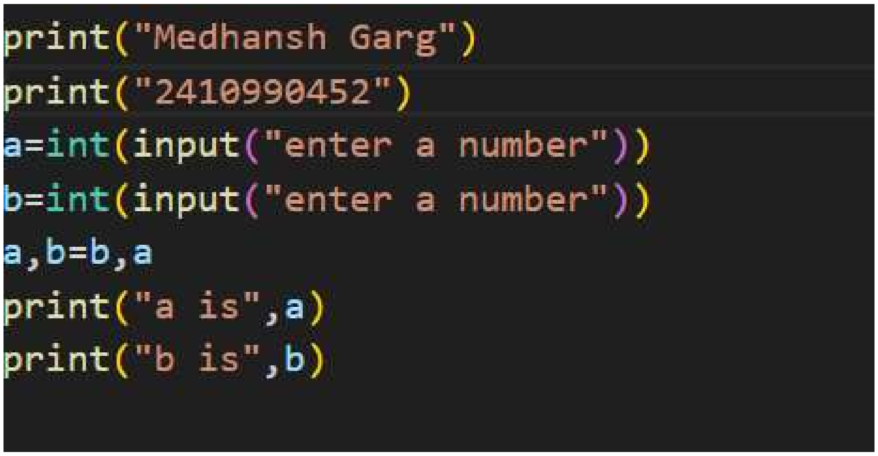
```
print("Medhansh Garg")
print("2410990452")
#sample example
tuple=eval(input("enter  a tuple of no"))
lengthtuple=len(tuple)


print(lengthtuple)
```

Output:



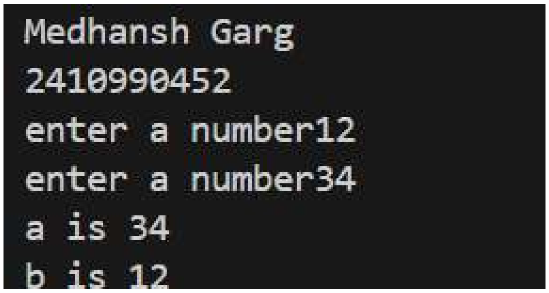
```
Medhansh Garg
2410990452
enter  a tuple of no(1,2,3,4)
4
PS C:\Users\welcome\Desktop\python
```

Swap two numbers using tuple



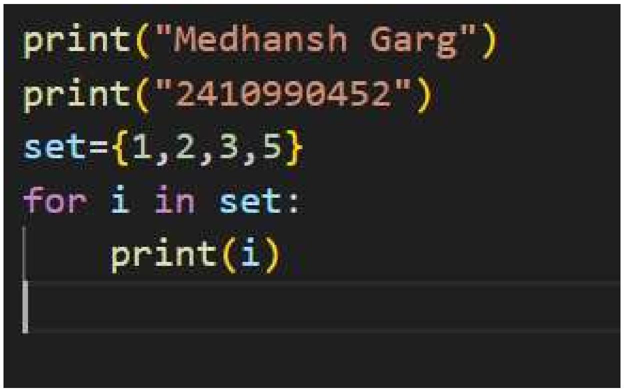
```
print("Medhansh Garg")
print("2410990452")
a=int(input("enter a number"))
b=int(input("enter a number"))
a,b=b,a
print("a is",a)
print("b is",b)
```

Output:



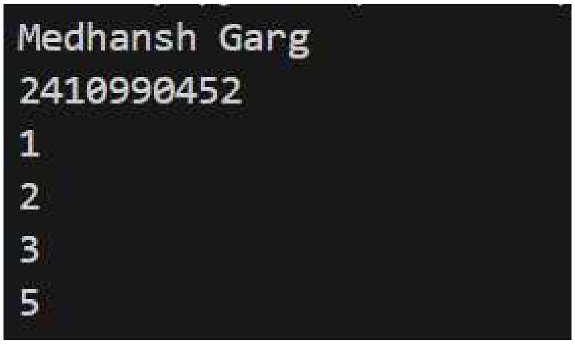
```
Medhansh Garg
2410990452
enter a number12
enter a number34
a is 34
b is 12
```

Create a set and print its elements



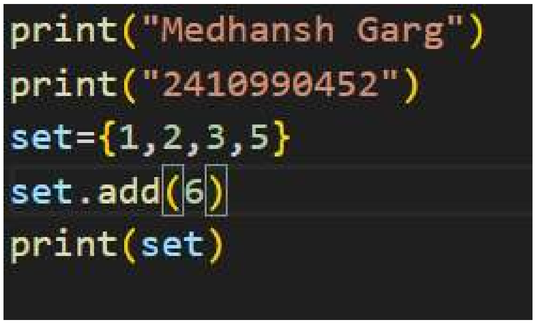
```python
print("Medhansh Garg")
print("2410990452")
set={1,2,3,5}
for i in set:
    print(i)
```

Output:



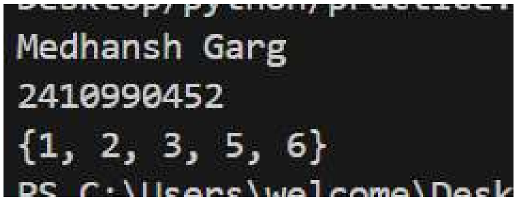
```
Medhansh Garg
2410990452
1
2
3
5
```

Add an element to set



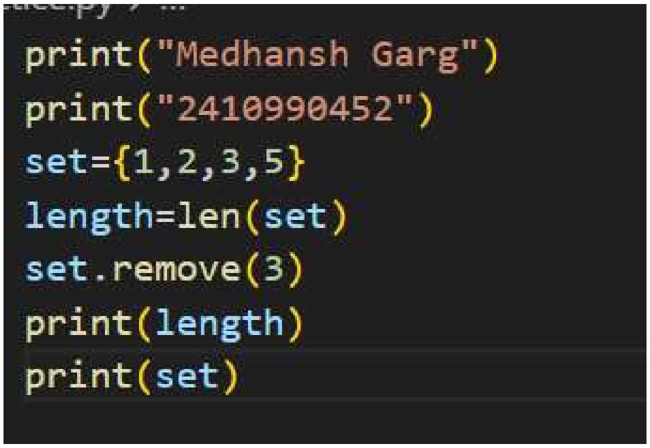
```
print("Medhansh Garg")
print("2410990452")
set={1,2,3,5}
set.add(6)
print(set)
```

Output:



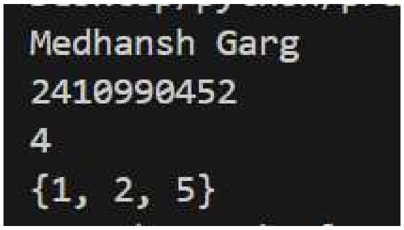
```
Desktop/python/practice.
Medhansh Garg
2410990452
{1, 2, 3, 5, 6}
PS C:\Users\welcome\Desk
```

find length of the set and then remove the element



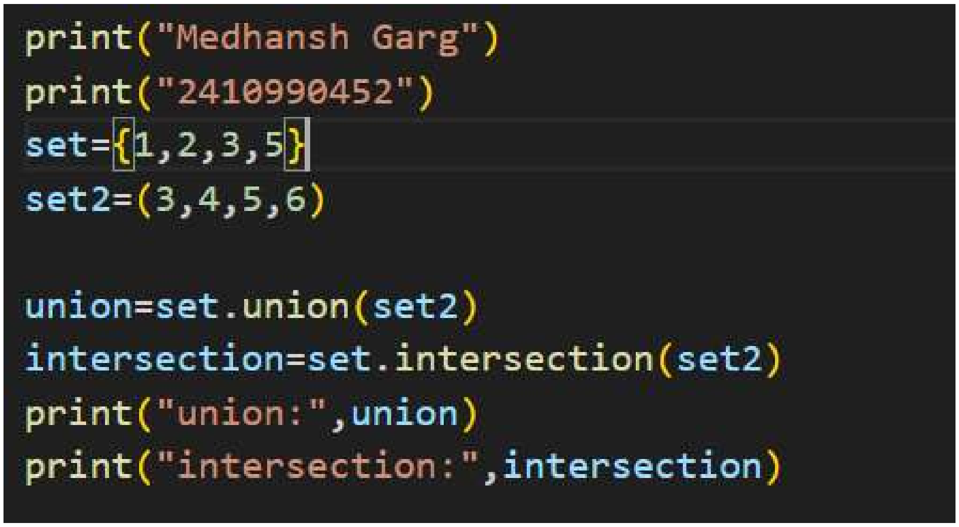
```
print("Medhansh Garg")
print("2410990452")
set={1,2,3,5}
length=len(set)
set.remove(3)
print(length)
print(set)
```

Output:



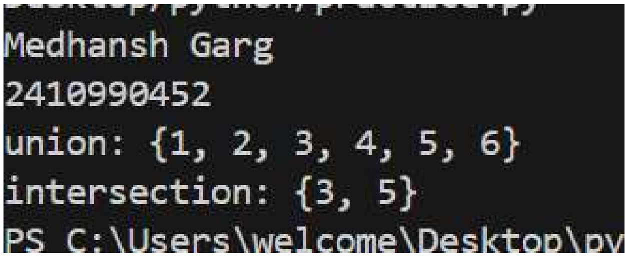
```
Medhansh Garg
2410990452
4
{1, 2, 5}
```

Perform union and intersection on two sets



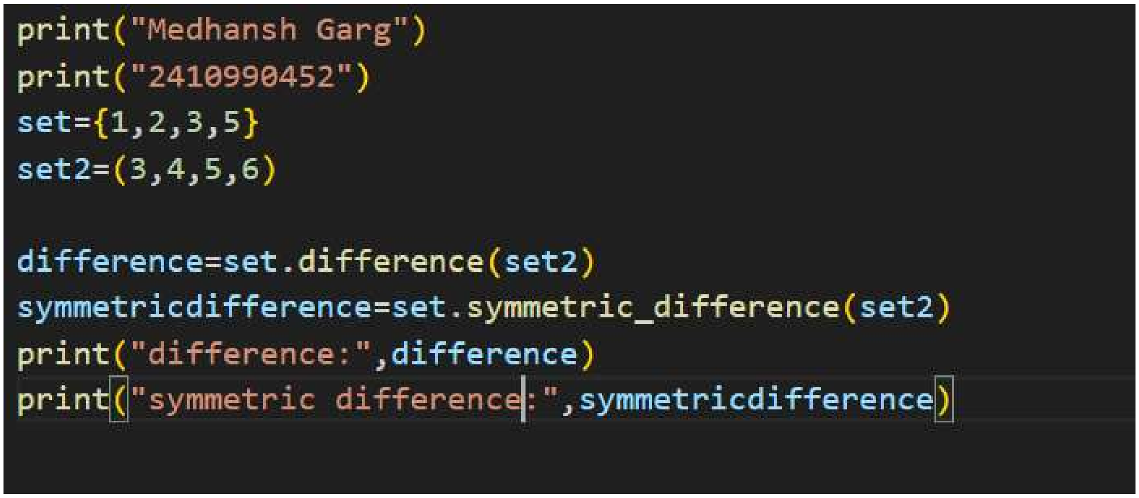
```
print("Medhansh Garg")
print("2410990452")
set={1,2,3,5}
set2=(3,4,5,6)

union=set.union(set2)
intersection=set.intersection(set2)
print("union:",union)
print("intersection:",intersection)
```

Output:



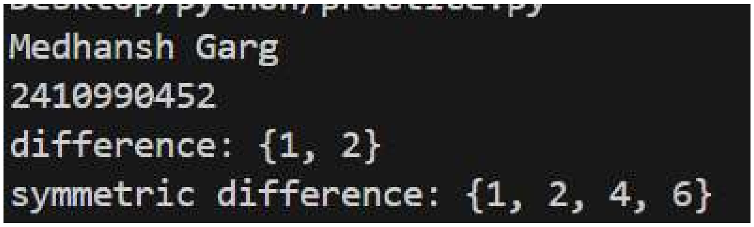
```
Medhansh Garg
2410990452
union: {1, 2, 3, 4, 5, 6}
intersection: {3, 5}
PS C:\Users\welcome\Desktop\py
```

Perform difference and symmetric difference on sets



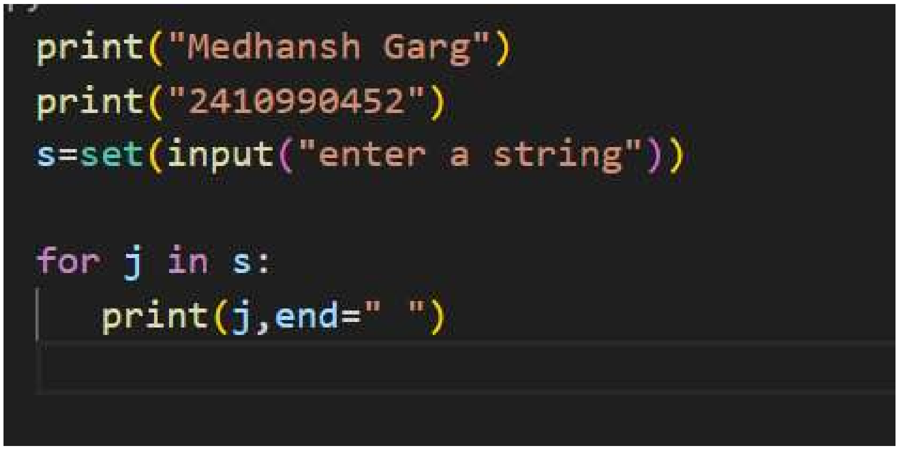
```
print("Medhansh Garg")
print("2410990452")
set={1,2,3,5}
set2=(3,4,5,6)

difference=set.difference(set2)
symmetricdifference=set.symmetric_difference(set2)
print("difference:",difference)
print("symmetric difference:",symmetricdifference)
```

Output:



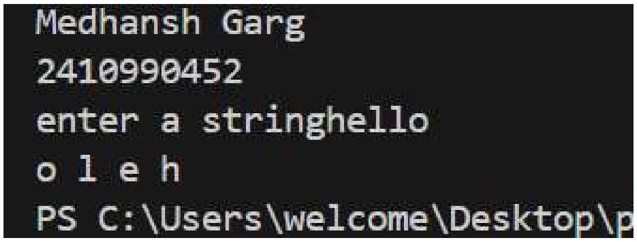
```
Desktop/python/practice.py
Medhansh Garg
2410990452
difference: {1, 2}
symmetric difference: {1, 2, 4, 6}
```

Accept a string in set and print it,s elements.



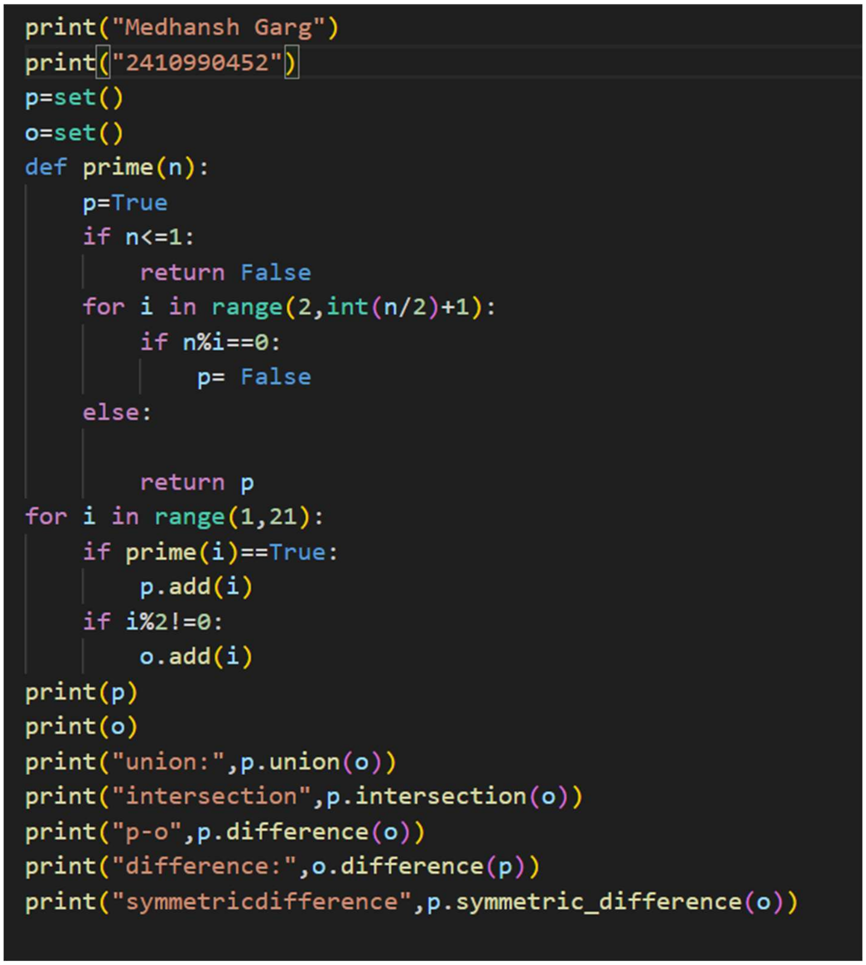
```python
print("Medhansh Garg")
print("2410990452")
s=set(input("enter a string"))

for j in s:
    print(j,end=" ")
```

Output:



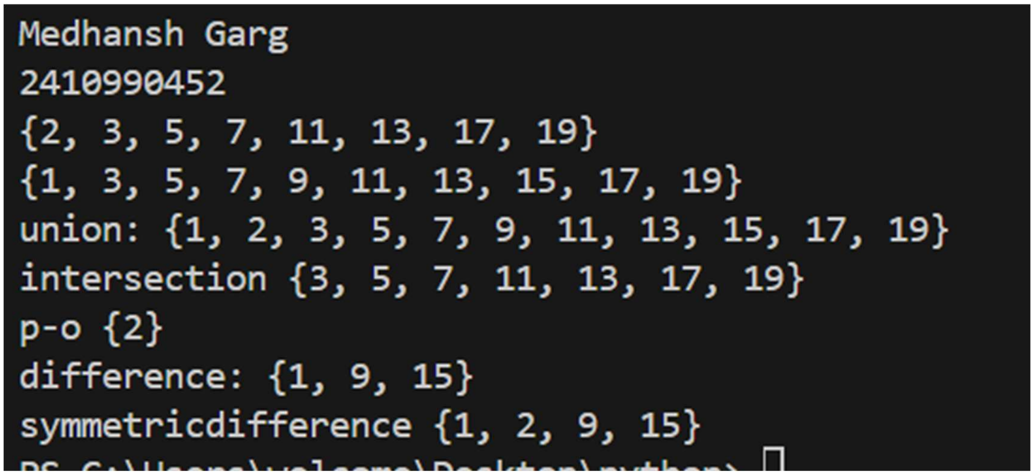
```
Medhansh Garg
2410990452
enter a stringhello
o l e h
PS C:\Users\welcome\Desktop\p
```

Generate sets containing prime no and odd no below 20 and find perform various operations on them.



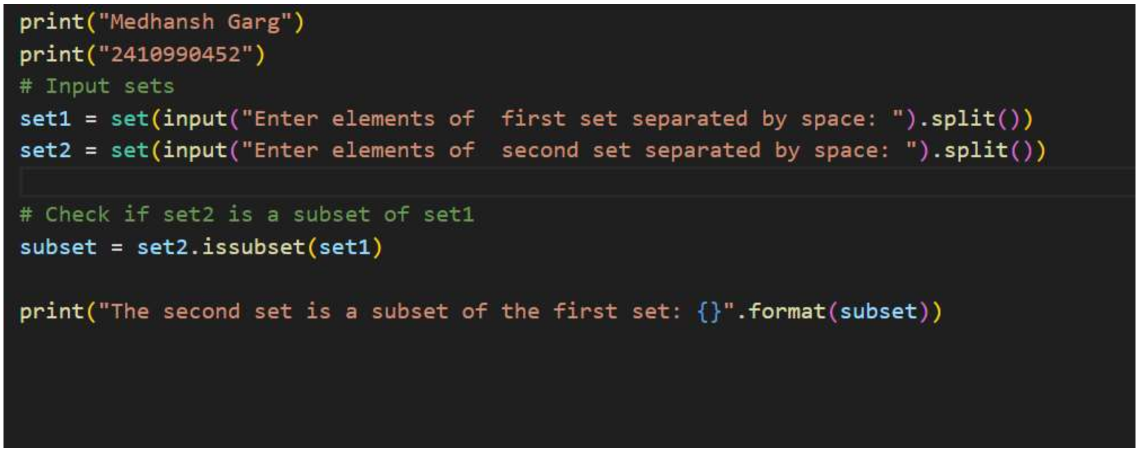
```
print("Medhansh Garg")
print("2410990452")
p=set()
o=set()
def prime(n):
    p=True
    if n<=1:
        return False
    for i in range(2,int(n/2)+1):
        if n%i==0:
            p= False
    else:

        return p
for i in range(1,21):
    if prime(i)==True:
        p.add(i)
    if i%2!=0:
        o.add(i)
print(p)
print(o)
print("union:",p.union(o))
print("intersection",p.intersection(o))
print("p-o",p.difference(o))
print("difference:",o.difference(p))
print("symmetricdifference",p.symmetric_difference(o))
```

Output:



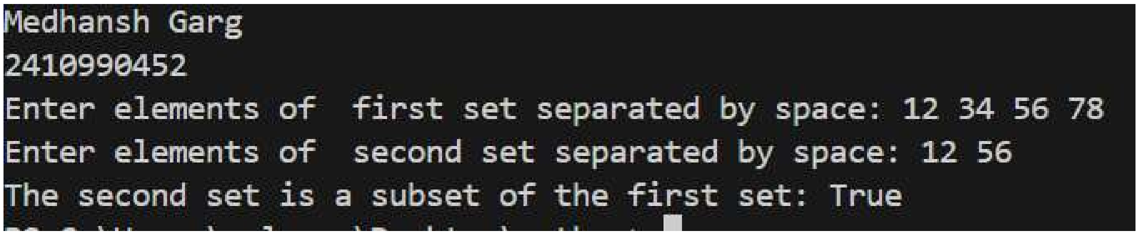
```
Medhansh Garg
2410990452
{2, 3, 5, 7, 11, 13, 17, 19}
{1, 3, 5, 7, 9, 11, 13, 15, 17, 19}
union: {1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19}
intersection {3, 5, 7, 11, 13, 17, 19}
p-o {2}
difference: {1, 9, 15}
symmetricdifference {1, 2, 9, 15}
PS C:\Users\welcome\Desktop\python> 
```

Input a set and check if other set is subset of it.



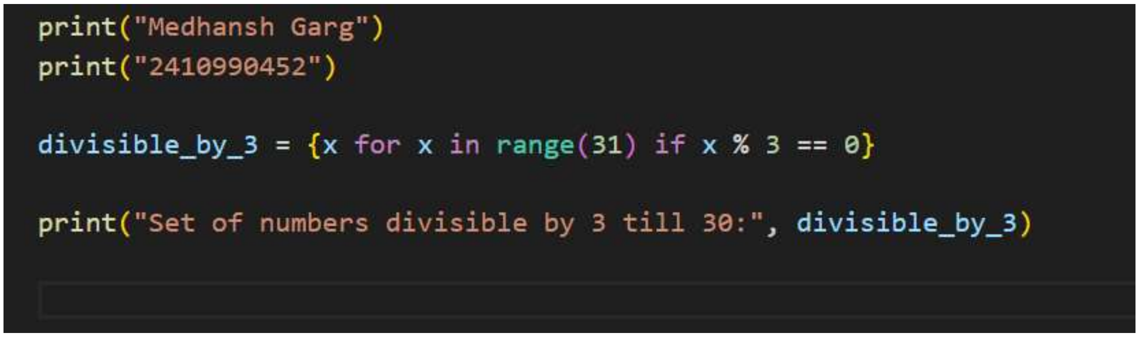
```
print("Medhansh Garg")
print("2410990452")
# Input sets
set1 = set(input("Enter elements of  first set separated by space: ").split())
set2 = set(input("Enter elements of  second set separated by space: ").split())

# Check if set2 is a subset of set1
subset = set2.issubset(set1)

print("The second set is a subset of the first set: {}".format(subset))
```

Output:



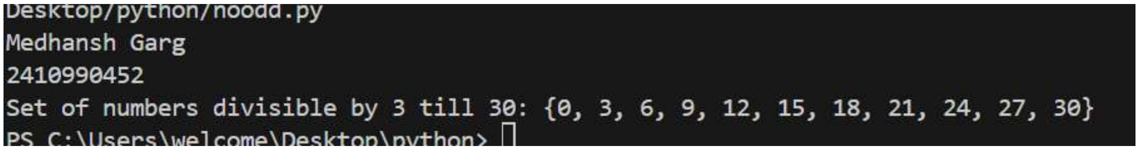
```
Medhansh Garg
2410990452
Enter elements of  first set separated by space: 12 34 56 78
Enter elements of  second set separated by space: 12 56
The second set is a subset of the first set: True
```

Create a set using set comprehension comprised of numbers divisible by 3 till 30.



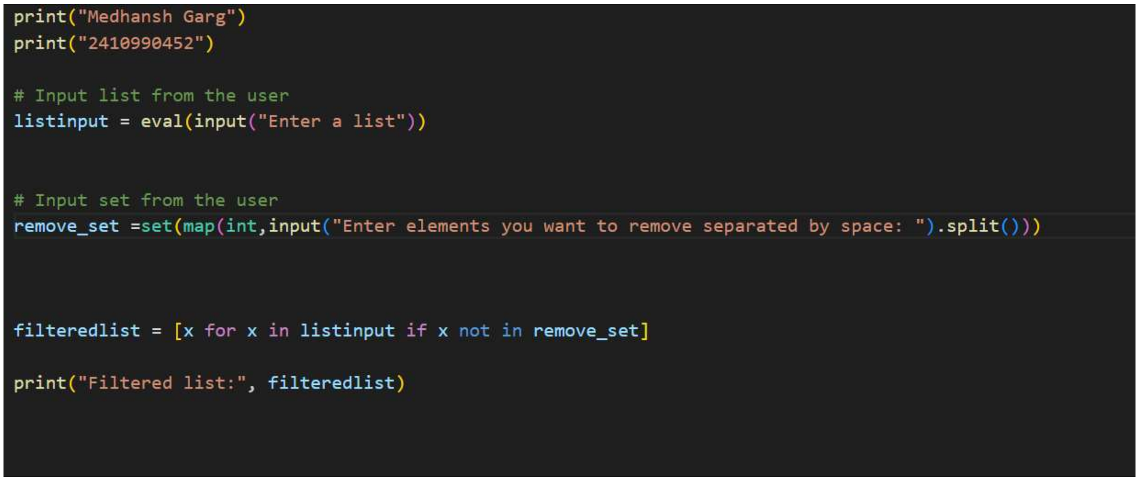
```
print("Medhansh Garg")
print("2410990452")

divisible_by_3 = {x for x in range(31) if x % 3 == 0}

print("Set of numbers divisible by 3 till 30:", divisible_by_3)
```

Output:



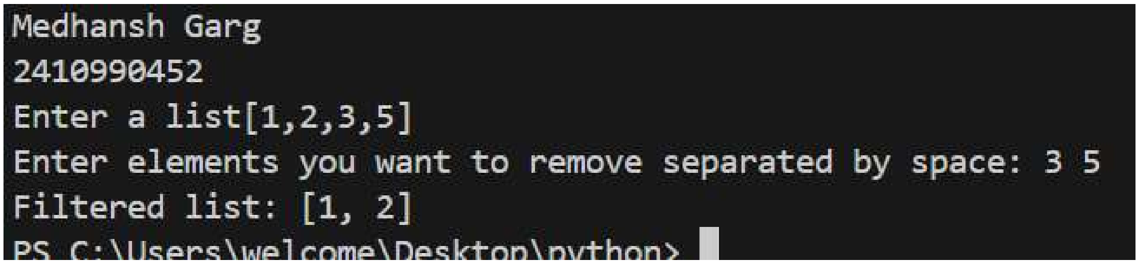
```
Desktop/python/noodd.py
Medhansh Garg
2410990452
Set of numbers divisible by 3 till 30: {0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30}
PS C:\Users\welcome\Desktop\python> 
```

Remove elements from list using a set.



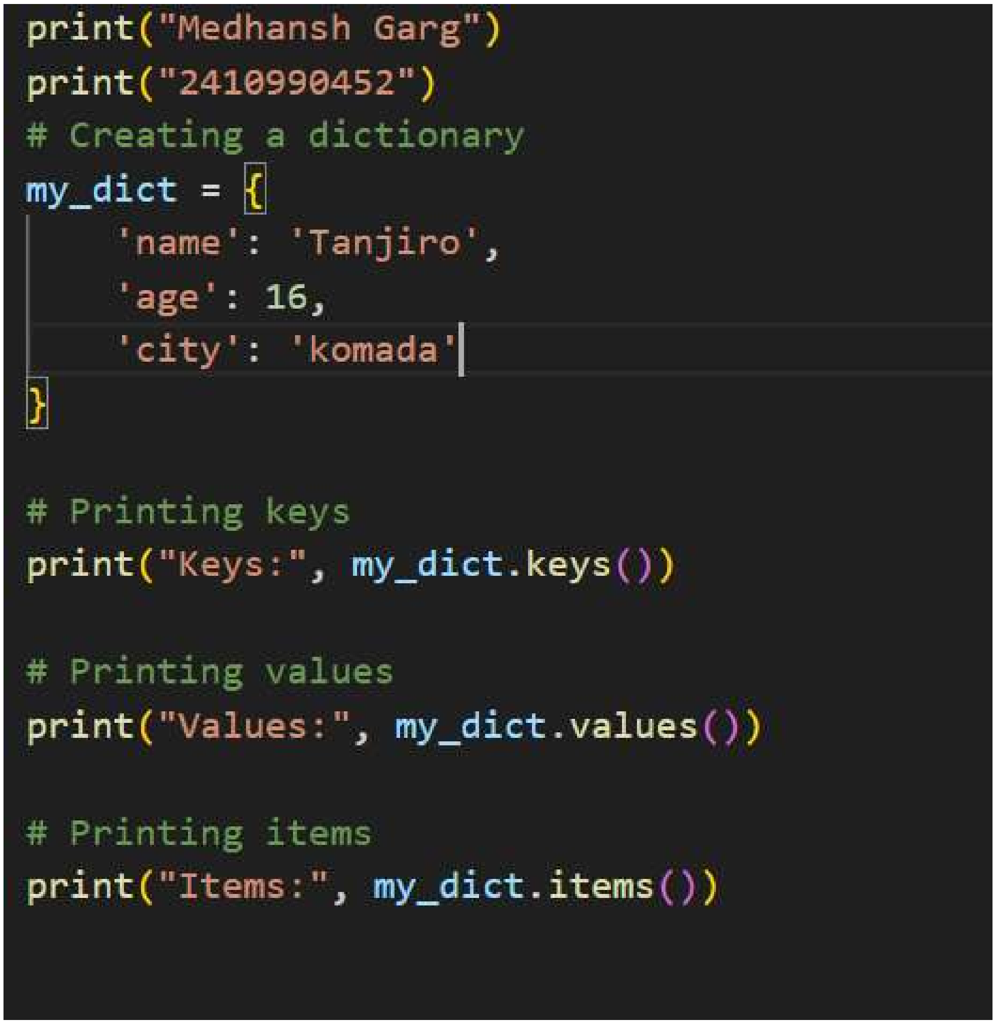
```python
print("Medhansh Garg")
print("2410990452")

# Input list from the user
listinput = eval(input("Enter a list"))


# Input set from the user
remove_set =set(map(int,input("Enter elements you want to remove separated by space: ").split()))



filteredlist = [x for x in listinput if x not in remove_set]

print("Filtered list:", filteredlist)
```

Output:



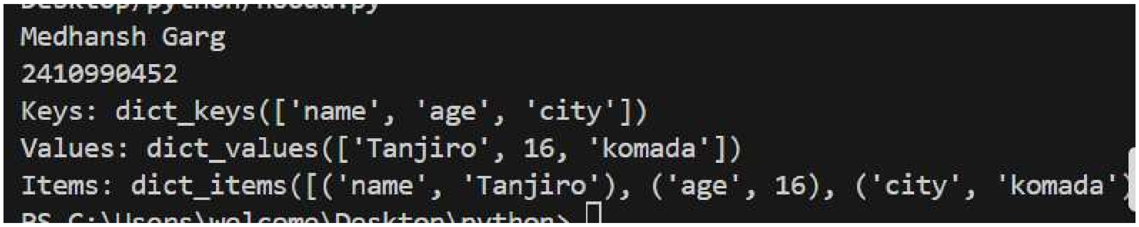
```
Medhansh Garg
2410990452
Enter a list[1,2,3,5]
Enter elements you want to remove separated by space: 3 5
Filtered list: [1, 2]
PS C:\Users\welcome\Desktop\python>
```

Create a dictionary a print its keys ,values and items



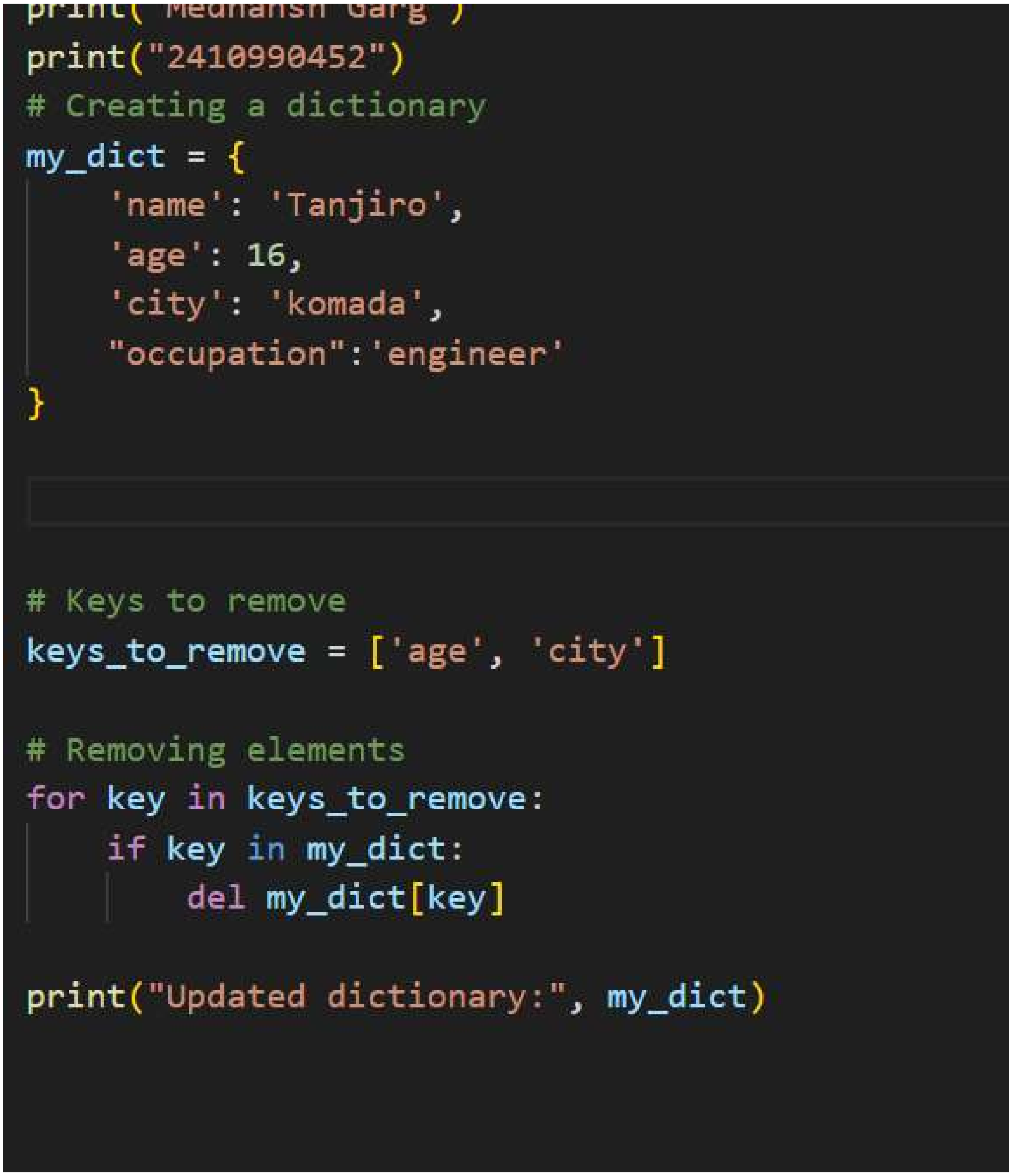
```
print("Medhansh Garg")
print("2410990452")
# Creating a dictionary
my_dict = {
    'name': 'Tanjiro',
    'age': 16,
    'city': 'komada'
}

# Printing keys
print("Keys:", my_dict.keys())

# Printing values
print("Values:", my_dict.values())

# Printing items
print("Items:", my_dict.items())
```

Output:



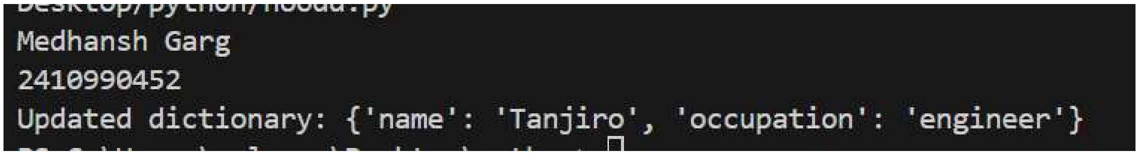
```
Medhansh Garg
2410990452
Keys: dict_keys(['name', 'age', 'city'])
Values: dict_values(['Tanjiro', 16, 'komada'])
Items: dict_items([('name', 'Tanjiro'), ('age', 16), ('city', 'komada')
PS C:\Users\welcome\Desktop\python>
```

Remove elements from dictionary



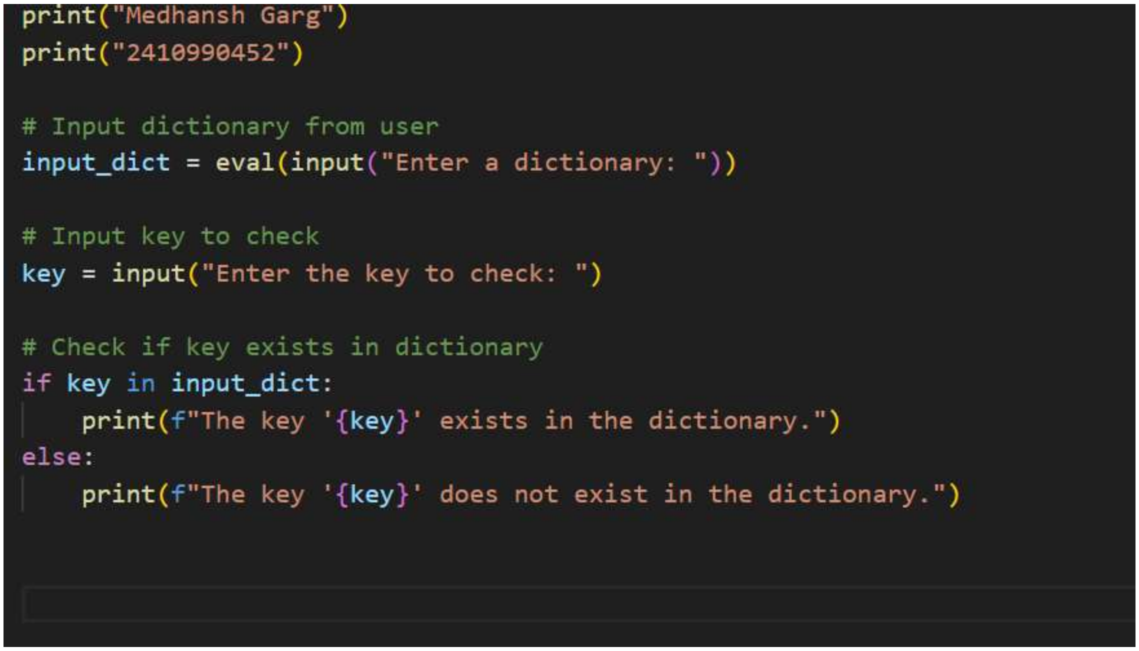
```
print("Medhansh Garg")
print("2410990452")
# Creating a dictionary
my_dict = {
    'name': 'Tanjiro',
    'age': 16,
    'city': 'komada',
    "occupation":'engineer'
}



# Keys to remove
keys_to_remove = ['age', 'city']

# Removing elements
for key in keys_to_remove:
    if key in my_dict:
        del my_dict[key]

print("Updated dictionary:", my_dict)
```

Output:



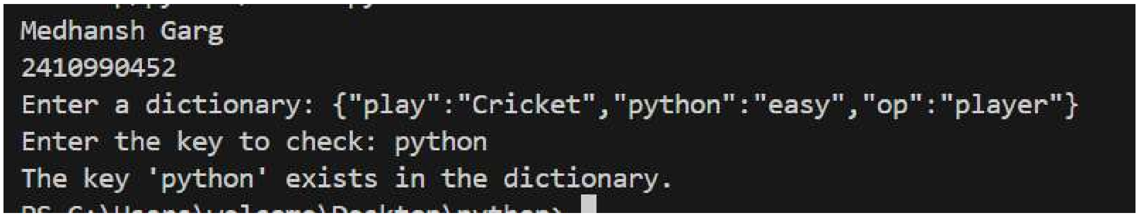
```
Desktop/python/noodu.py
Medhansh Garg
2410990452
Updated dictionary: {'name': 'Tanjiro', 'occupation': 'engineer'}
```

To check if a key exists in a dictionary



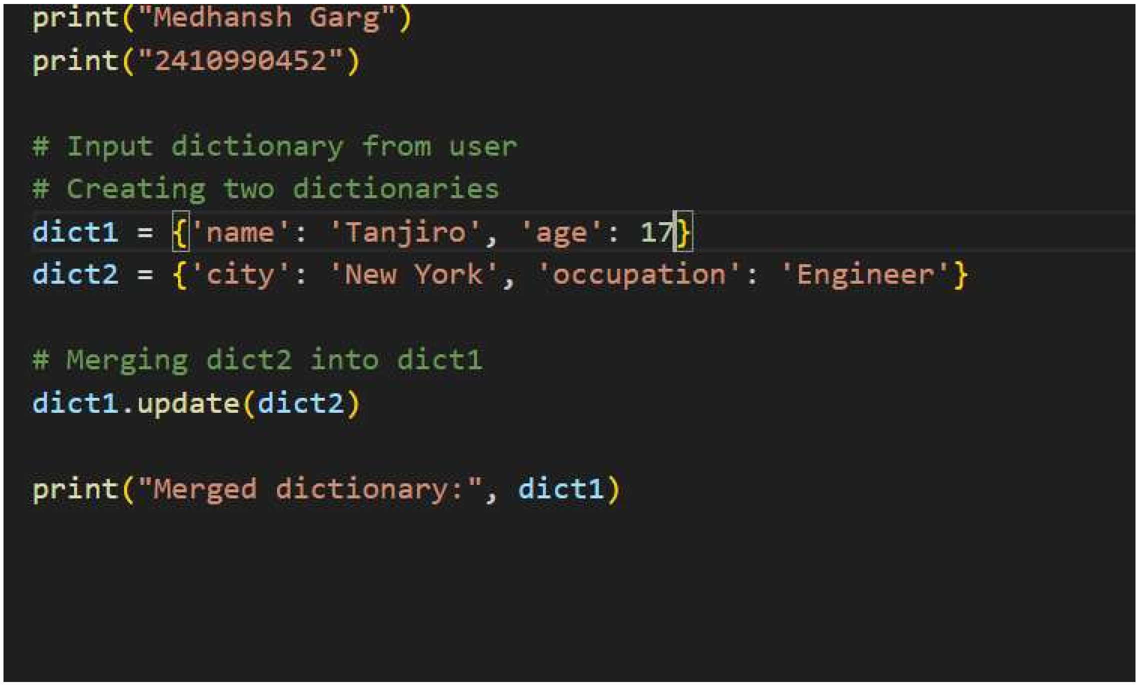
```python
print("Medhansh Garg")
print("2410990452")

# Input dictionary from user
input_dict = eval(input("Enter a dictionary: "))

# Input key to check
key = input("Enter the key to check: ")

# Check if key exists in dictionary
if key in input_dict:
    print(f"The key '{key}' exists in the dictionary.")
else:
    print(f"The key '{key}' does not exist in the dictionary.")
```

Output:



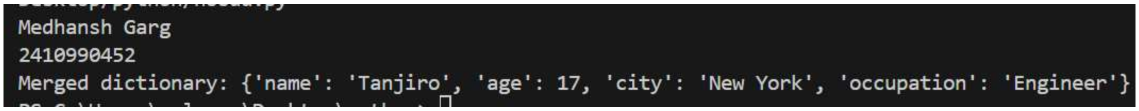
```
Medhansh Garg
2410990452
Enter a dictionary: {"play":"Cricket","python":"easy","op":"player"}
Enter the key to check: python
The key 'python' exists in the dictionary.
PS C:\Users\welcome\Desktop\python>
```

Merge two dictionaries using update()



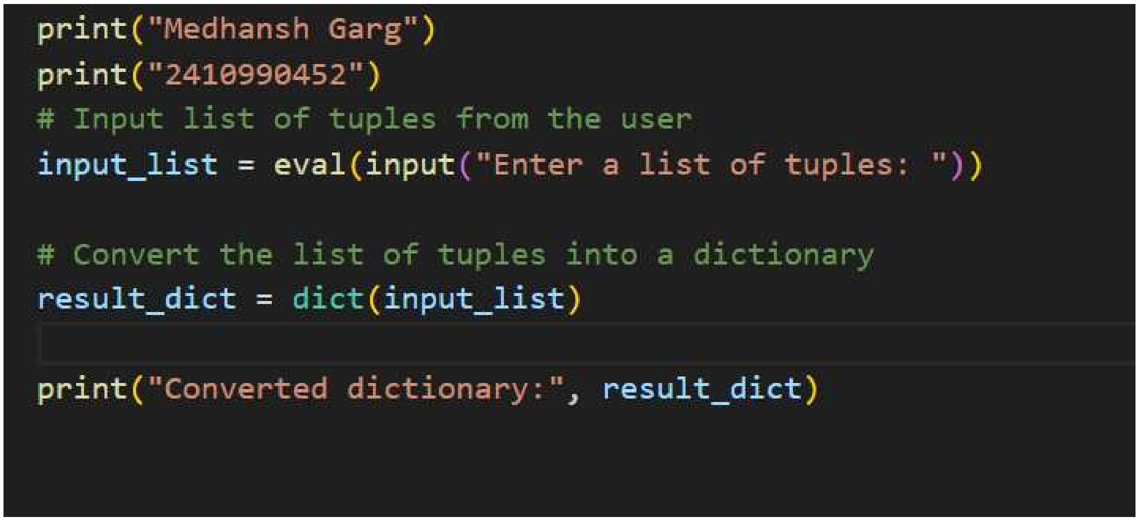
```python
print("Medhansh Garg")
print("2410990452")

# Input dictionary from user
# Creating two dictionaries
dict1 = {'name': 'Tanjiro', 'age': 17}
dict2 = {'city': 'New York', 'occupation': 'Engineer'}

# Merging dict2 into dict1
dict1.update(dict2)

print("Merged dictionary:", dict1)
```

Output:



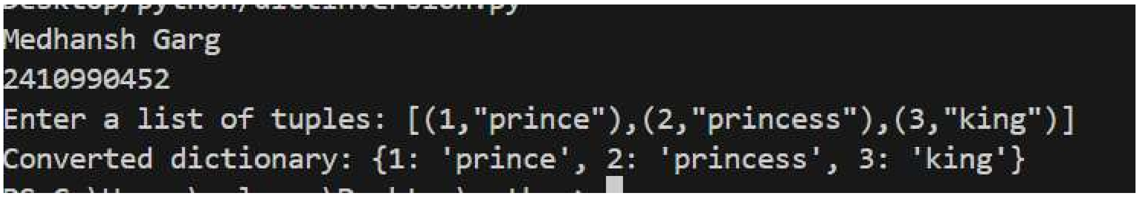
```
Medhansh Garg
2410990452
Merged dictionary: {'name': 'Tanjiro', 'age': 17, 'city': 'New York', 'occupation': 'Engineer'}
```

Convert list of tuples into dictionary



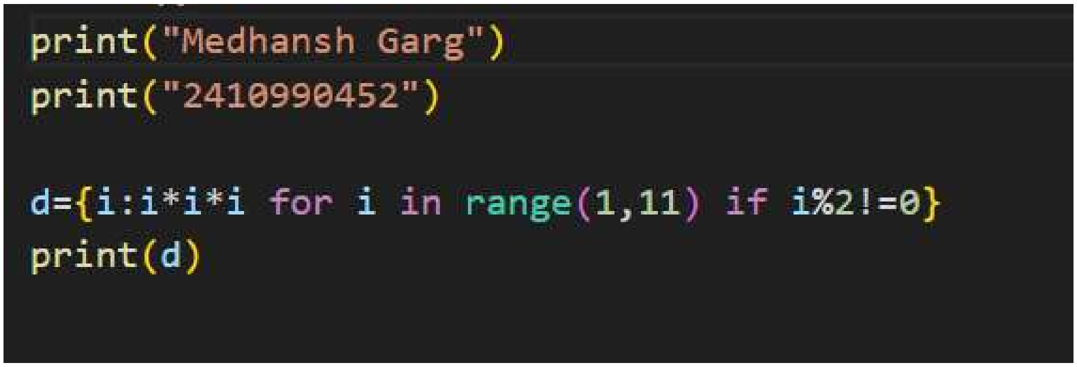
```
print("Medhansh Garg")
print("2410990452")
# Input list of tuples from the user
input_list = eval(input("Enter a list of tuples: "))

# Convert the list of tuples into a dictionary
result_dict = dict(input_list)

print("Converted dictionary:", result_dict)
```

Output:



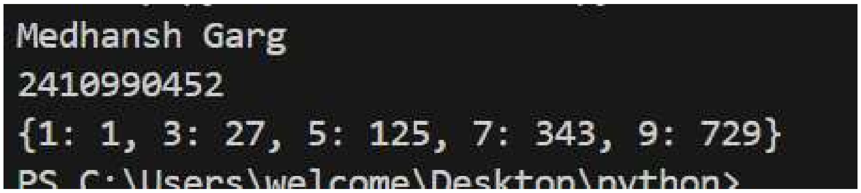
```
Desktop/python/dictinversion.py
Medhansh Garg
2410990452
Enter a list of tuples: [(1,"prince"),(2,"princess"),(3,"king")]
Converted dictionary: {1: 'prince', 2: 'princess', 3: 'king'}
```

Create a dictionary using dictionary comprehension



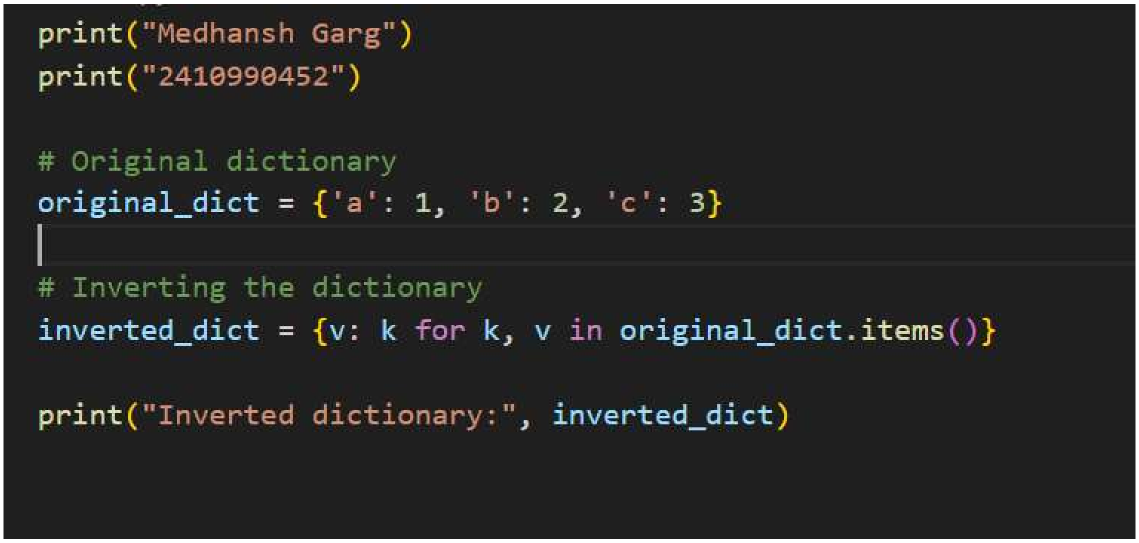
```
print("Medhansh Garg")
print("2410990452")

d={i:i*i*i for i in range(1,11) if i%2!=0}
print(d)
```

Output:



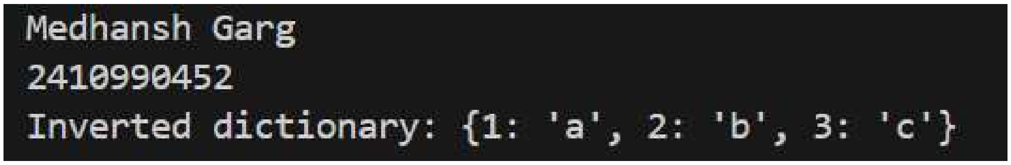
```
Medhansh Garg
2410990452
{1: 1, 3: 27, 5: 125, 7: 343, 9: 729}
PS C:\Users\welcome\Desktop\python>
```

Program to invert a dictionary using dictionary comprehension



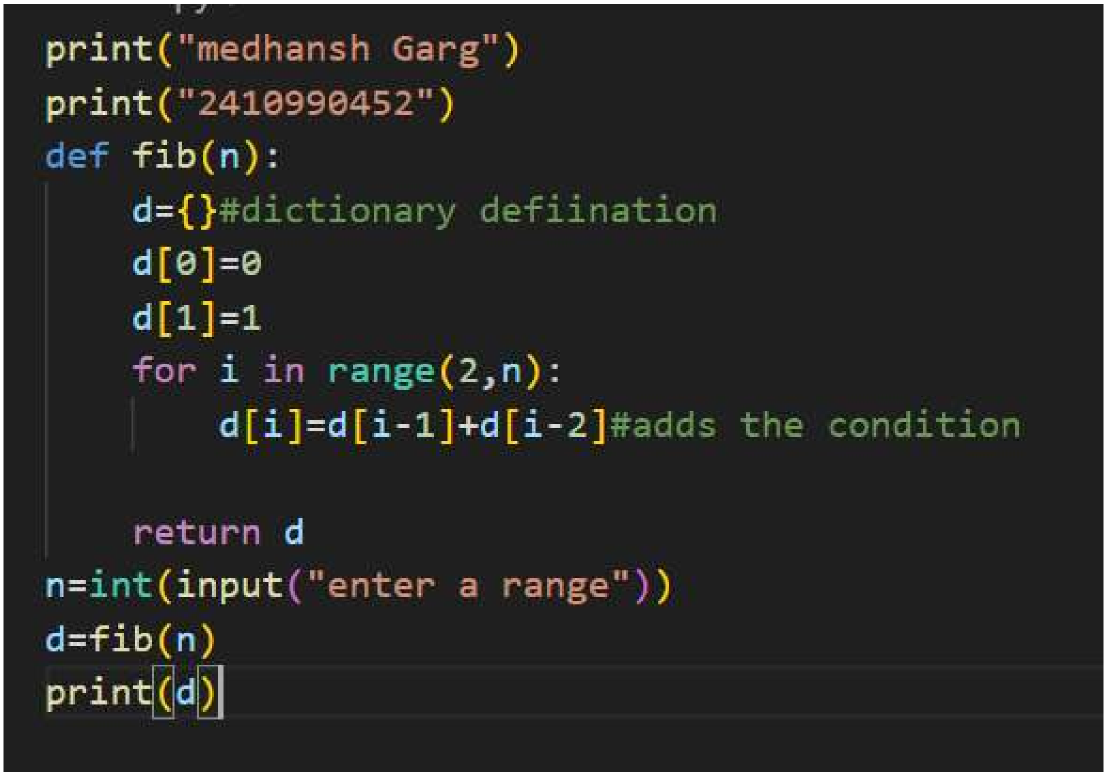
```python
print("Medhansh Garg")
print("2410990452")

# Original dictionary
original_dict = {'a': 1, 'b': 2, 'c': 3}

# Inverting the dictionary
inverted_dict = {v: k for k, v in original_dict.items()}

print("Inverted dictionary:", inverted_dict)
```

Output:



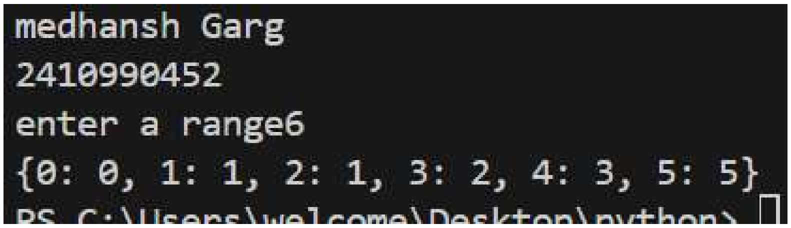
```
Medhansh Garg
2410990452
Inverted dictionary: {1: 'a', 2: 'b', 3: 'c'}
```

Programme to calculate the n Fibonacci terms using dictionary.



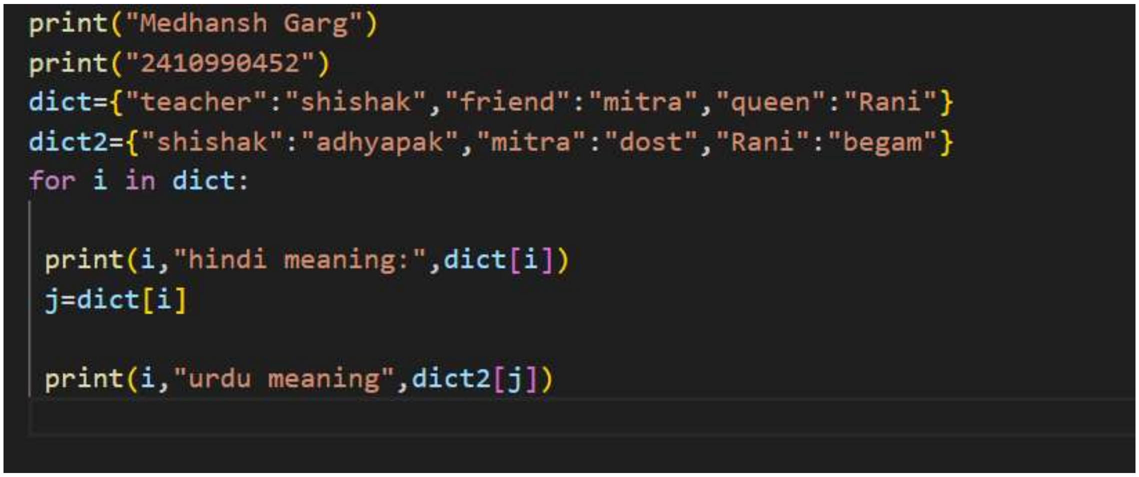
```
print("medhansh Garg")
print("2410990452")
def fib(n):
    d={}#dictionary defiination
    d[0]=0
    d[1]=1
    for i in range(2,n):
        d[i]=d[i-1]+d[i-2]#adds the condition

    return d
n=int(input("enter a range"))
d=fib(n)
print(d)
```

Output:



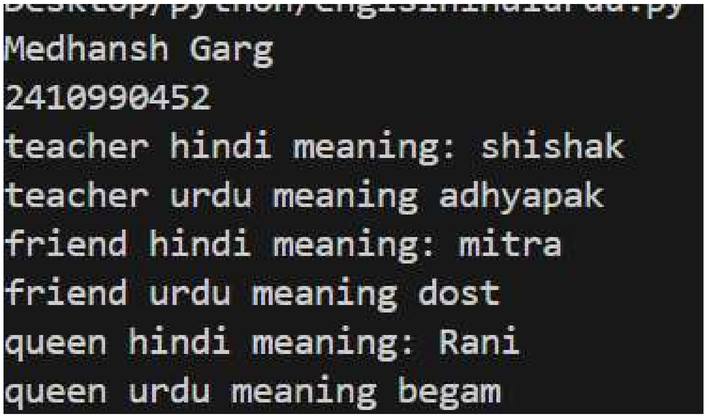
```
medhansh Garg
2410990452
enter a range6
{0: 0, 1: 1, 2: 1, 3: 2, 4: 3, 5: 5}
PS C:\Users\welcome\Desktop\python> 
```

Write a program that has a dictionary of English and corresponding words in hindi and other dictionary with hindi and corresponding words in urdu.Take all words in English and display their meaning in both languages.



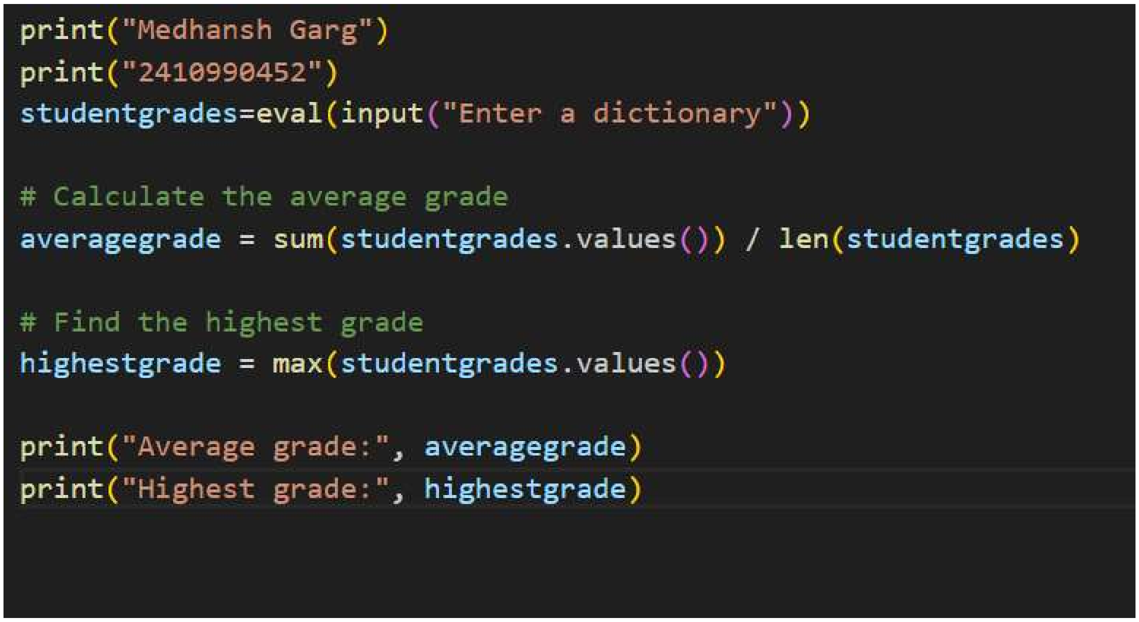
```
print("Medhansh Garg")
print("2410990452")
dict={"teacher":"shishak","friend":"mitra","queen":"Rani"}
dict2={"shishak":"adhyapak","mitra":"dost","Rani":"begam"}
for i in dict:

 print(i,"hindi meaning:",dict[i])
 j=dict[i]

 print(i,"urdu meaning",dict2[j])
```

Output:



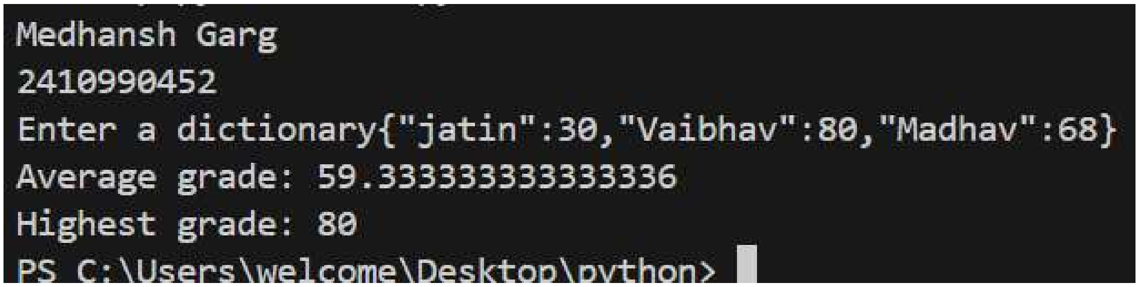
```
Desktop/python/englishhindiurdu.py
Medhansh Garg
2410990452
teacher hindi meaning: shishak
teacher urdu meaning adhyapak
friend hindi meaning: mitra
friend urdu meaning dost
queen hindi meaning: Rani
queen urdu meaning begam
```

given a dictionary with student names as keys and their grades as values, write a program to compute the average grade and the highest grade.



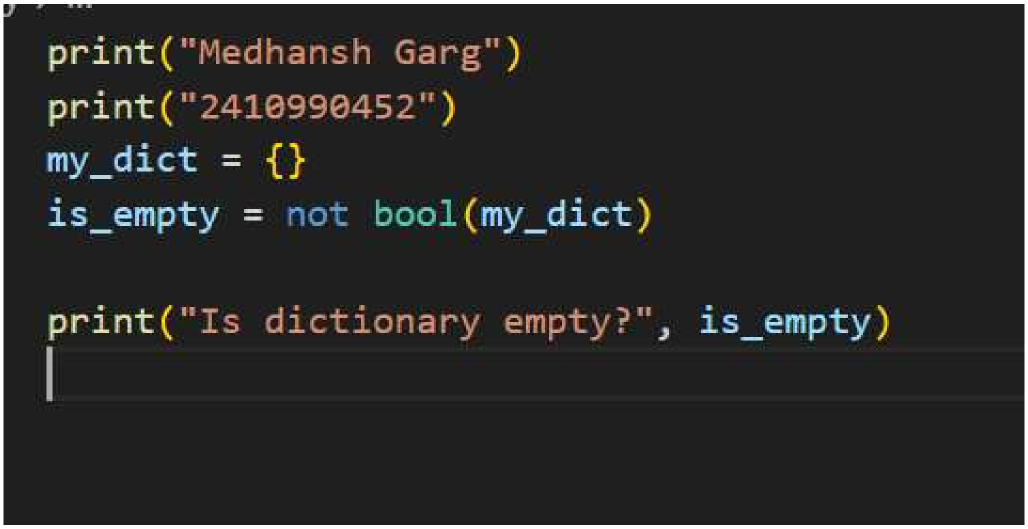
```
print("Medhansh Garg")
print("2410990452")
studentgrades=eval(input("Enter a dictionary"))

# Calculate the average grade
averagegrade = sum(studentgrades.values()) / len(studentgrades)

# Find the highest grade
highestgrade = max(studentgrades.values())

print("Average grade:", averagegrade)
print("Highest grade:", highestgrade)
```

Output:



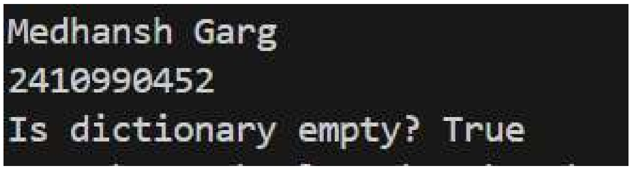
```
Medhansh Garg
2410990452
Enter a dictionary{"jatin":30,"Vaibhav":80,"Madhav":68}
Average grade:  59.333333333333336
Highest grade:  80
PS C:\Users\welcome\Desktop\python>
```

Program to check if a dictionary is empty.



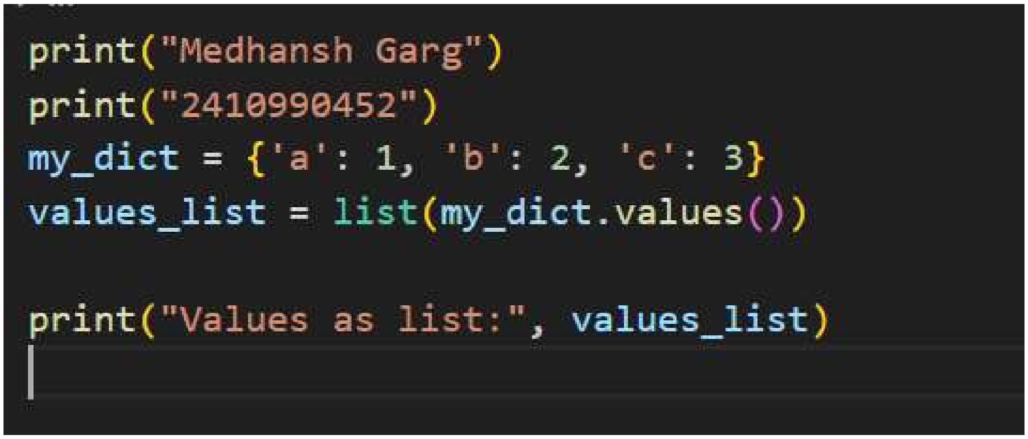
```
print("Medhansh Garg")
print("2410990452")
my_dict = {}
is_empty = not bool(my_dict)

print("Is dictionary empty?", is_empty)
```

Output:



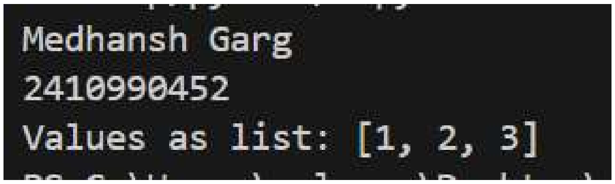
```
Medhansh Garg
2410990452
Is dictionary empty? True
```

Convert dictionary values to list.



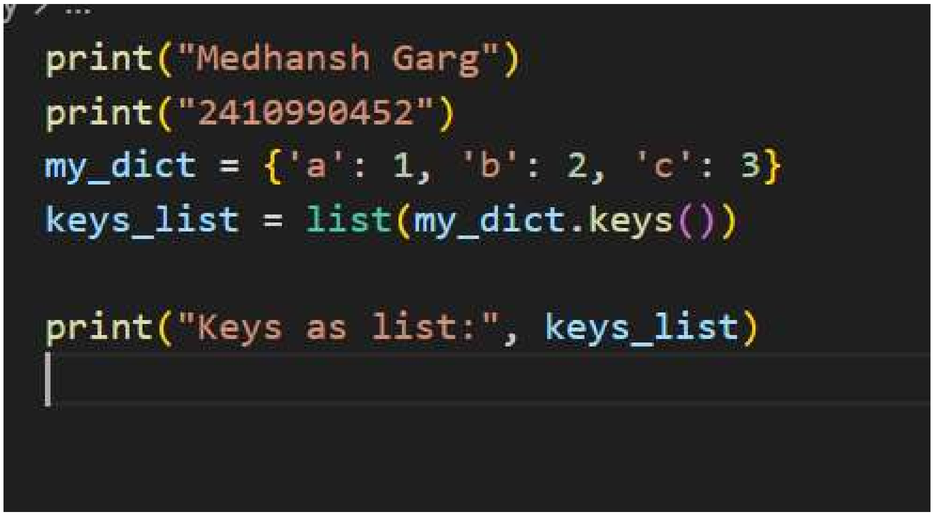
```
print("Medhansh Garg")
print("2410990452")
my_dict = {'a': 1, 'b': 2, 'c': 3}
values_list = list(my_dict.values())

print("Values as list:", values_list)
```

Output:



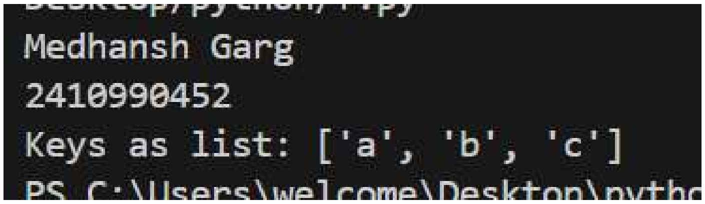
```
Medhansh Garg
2410990452
Values as list: [1, 2, 3]
```

Convert dictionary keys to list.



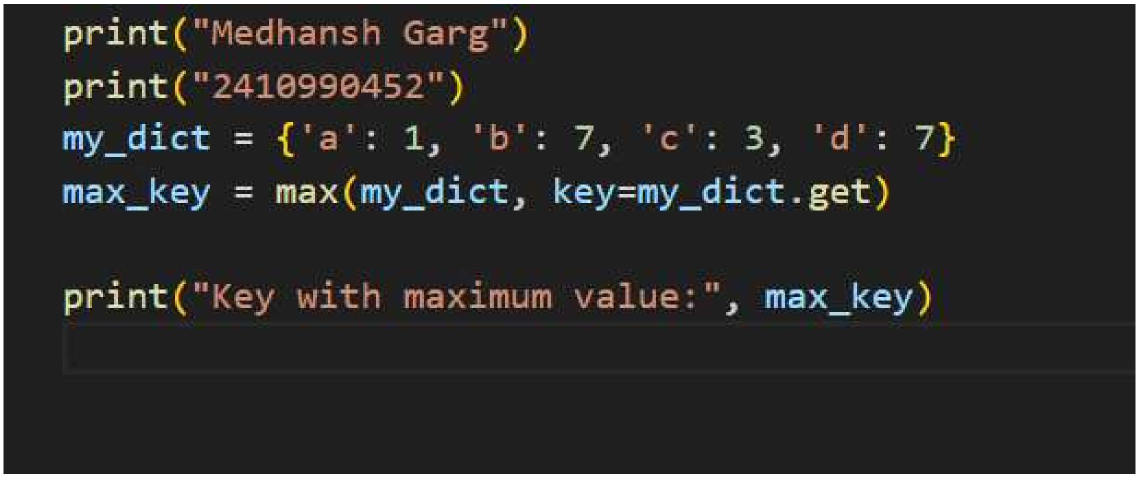
```
print("Medhansh Garg")
print("2410990452")
my_dict = {'a': 1, 'b': 2, 'c': 3}
keys_list = list(my_dict.keys())

print("Keys as list:", keys_list)
```

Output:



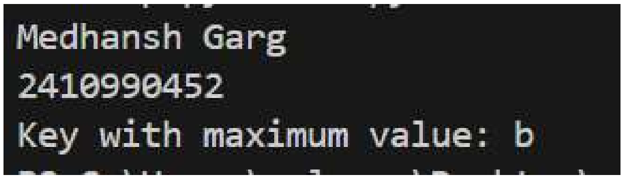
```
Desktop/python/1.py
Medhansh Garg
2410990452
Keys as list: ['a', 'b', 'c']
PS C:\Users\welcome\Desktop\pytho
```

Program to get key with maximum value



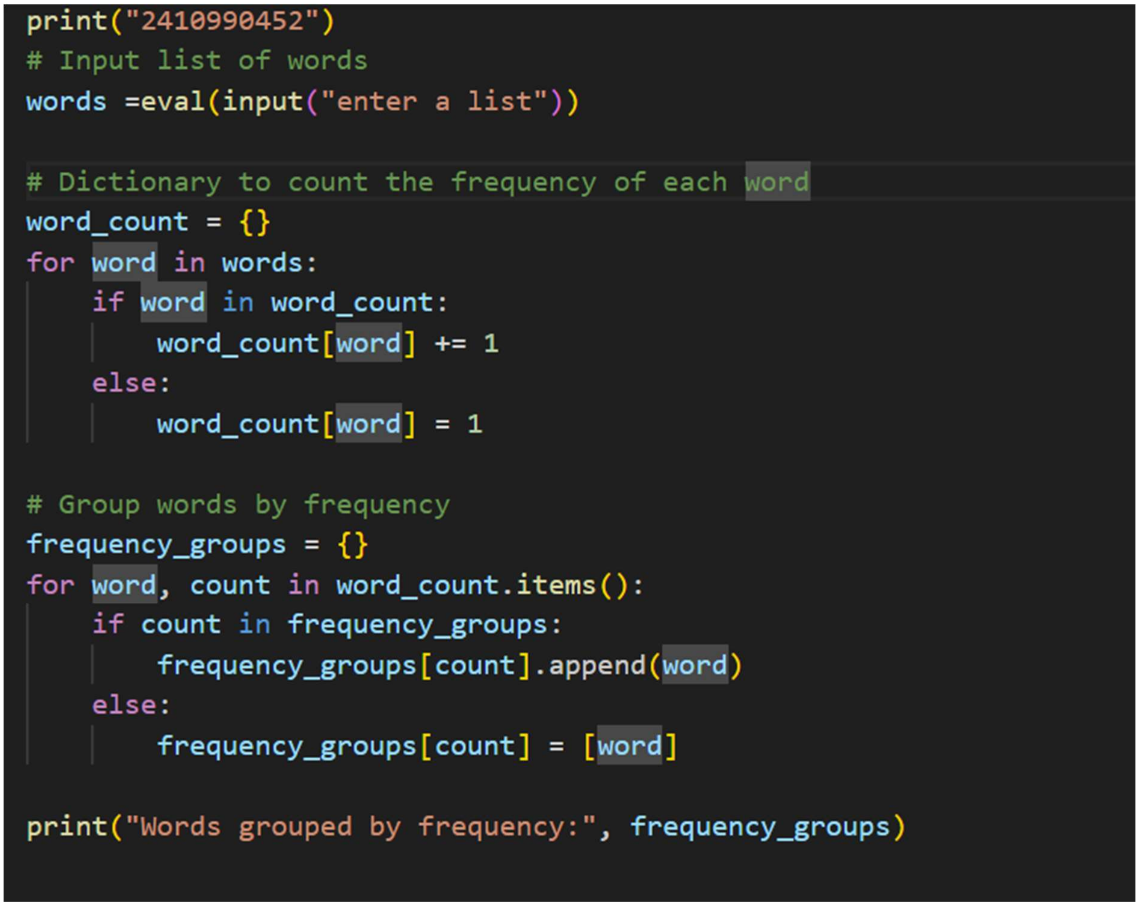
```python
print("Medhansh Garg")
print("2410990452")
my_dict = {'a': 1, 'b': 7, 'c': 3, 'd': 7}
max_key = max(my_dict, key=my_dict.get)

print("Key with maximum value:", max_key)
```

Output:



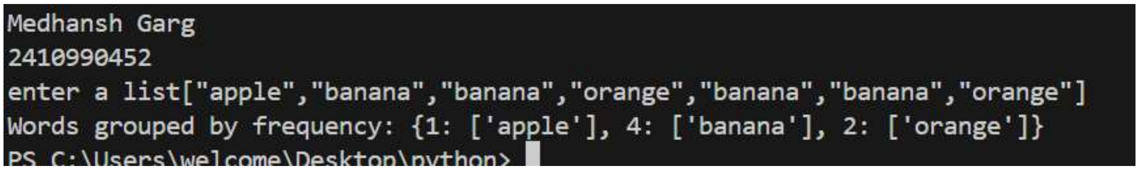
```
Medhansh Garg
2410990452
Key with maximum value: b
```

Create a dictionary which groups words by their frequency from a list.



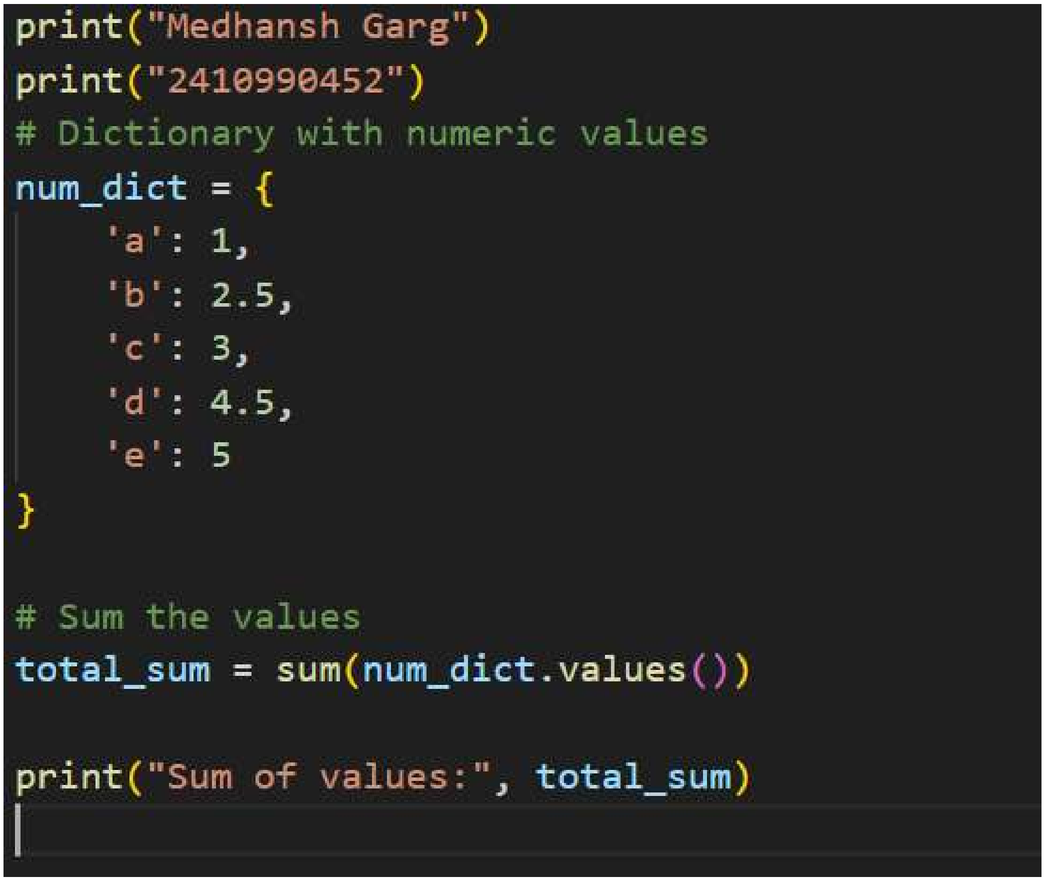
```
print("2410990452")
# Input list of words
words =eval(input("enter a list"))

# Dictionary to count the frequency of each word
word_count = {}
for word in words:
    if word in word_count:
        word_count[word] += 1
    else:
        word_count[word] = 1

# Group words by frequency
frequency_groups = {}
for word, count in word_count.items():
    if count in frequency_groups:
        frequency_groups[count].append(word)
    else:
        frequency_groups[count] = [word]

print("Words grouped by frequency:", frequency_groups)
```

Output:



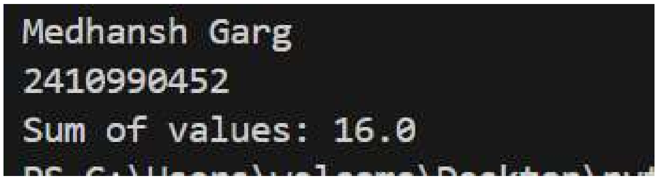
```
Medhansh Garg
2410990452
enter a list["apple","banana","banana","orange","banana","banana","orange"]
Words grouped by frequency: {1: ['apple'], 4: ['banana'], 2: ['orange']}
PS C:\Users\welcome\Desktop\python>
```

Write a program to sum of values of a dictionary where the values are integers or floats.



```python
print("Medhansh Garg")
print("2410990452")
# Dictionary with numeric values
num_dict = {
    'a': 1,
    'b': 2.5,
    'c': 3,
    'd': 4.5,
    'e': 5
}

# Sum the values
total_sum = sum(num_dict.values())

print("Sum of values:", total_sum)
```

Output:



```
Medhansh Garg
2410990452
Sum of values: 16.0
PS C:\Users\welcome\Desktop\pyt
```

Python file end