Institute of Engineering & Technology

Devi Ahilya Vishwavidyalaya, Indore (M.P)

Department of Computer Science & Engineering

Abstraction and Paradigms for Programming (CER4C3)

Lab-Assignments

Submitted To:	Submitted By:
Er. Neeraj Sharma Sir	Tanishq Chauhan (21C4184)
CS-Dept	CS “B” 2nd Year
IET-DAVV	

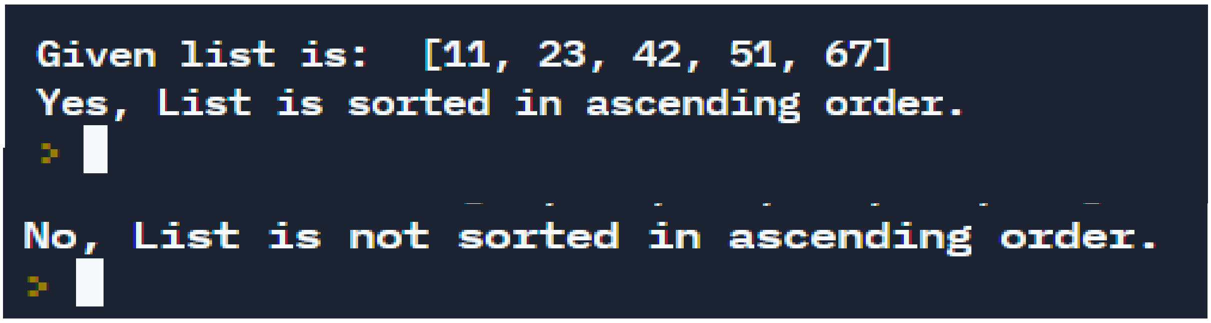
Lab Assignment-1

PYTHON LISTS

1. **Python program to check whether the elements in a list are sorted in ascending order or not.**

```
mylist = [11,23,42,51,67]
print("Given list is: ", mylist)
mylist_copy = mylist.copy()
mylist_copy.sort()
if (mylist == mylist_copy):
  print("Yes, List is sorted in ascending order.")
else:
  print("No, List is not sorted in ascending order.")
```

Output



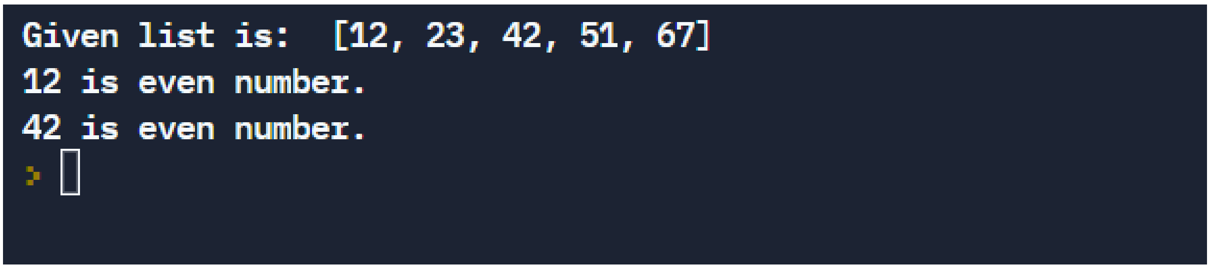
```
Given list is:  [11, 23, 42, 51, 67]
Yes, List is sorted in ascending order.
> 
```

```
No, List is not sorted in ascending order.
> 
```

2. Python program to find out even numbers in a list.

```
mylist = [12,23,42,51,67]
print("Given list is: ", mylist)
for num in mylist:
  if num % 2 == 0:
    print(f"{num} is even number.")
```

Output

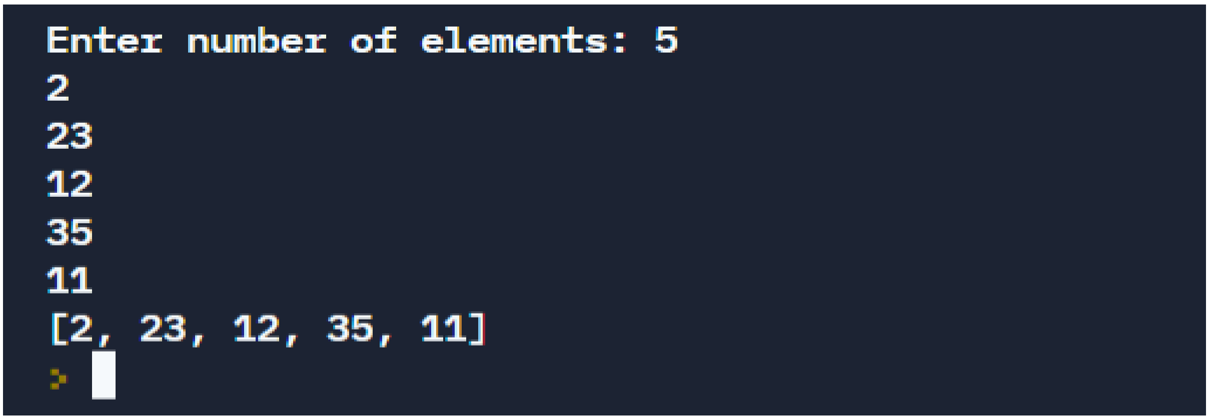


```
Given list is:  [12, 23, 42, 51, 67]
12 is even number.
42 is even number.
>
```

3. Python program to create list based on user input.

```
list = []
n = int(input("Enter number of elements: "))
for i in range(0, n):
  ele = int(input())
  list.append(ele)
print(list)
```

Output

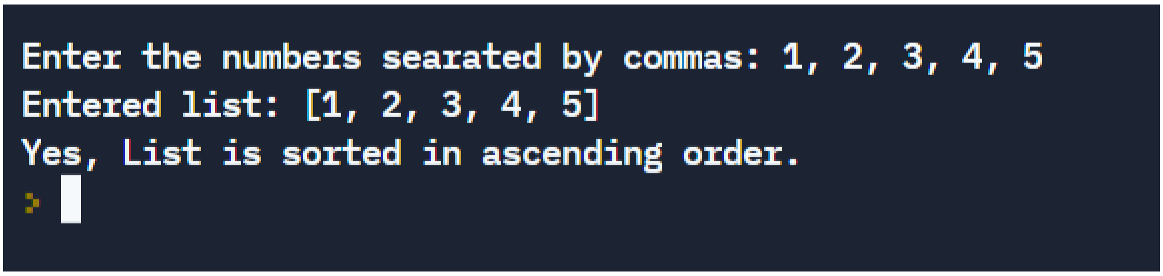


```
Enter number of elements: 5
2
23
12
35
11
[2, 23, 12, 35, 11]
> 
```

4. Write a program to check whether the list elements are sorted in ascending order, given that the list is created using user inputs.

```
user_input = input("Enter the numbers searated by commas: ")
list = [int(num) for num in user_input.split(',')]
print(f"Entered list: {list}")

first_element = list[0]
last_element = list[len(list) - 1]

if first_element < last_element:
  print("Yes, List is sorted in ascending order.")
else:
  print("No, List is not sorted in ascending order.")
```

Output

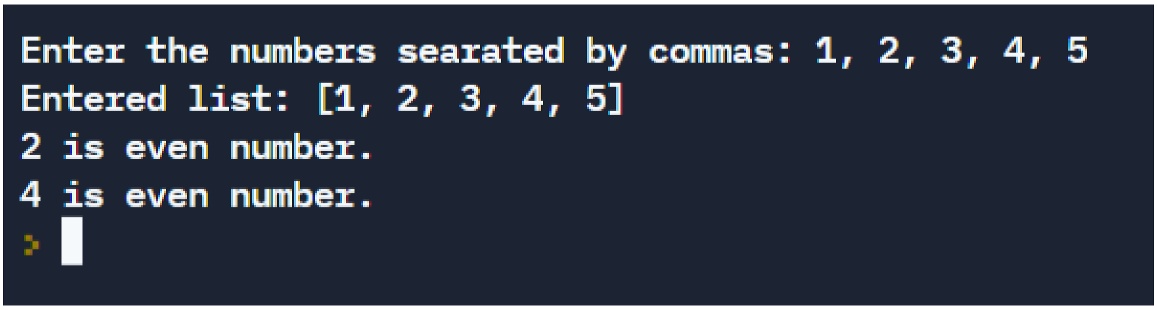


```
Enter the numbers searated by commas: 1, 2, 3, 4, 5
Entered list: [1, 2, 3, 4, 5]
Yes, List is sorted in ascending order.
> 
```

5. Write a program to find out even numbers in a list, given that the list is created using user inputs.

```
user_input = input("Enter the numbers searated by commas: ")
list = [int(num) for num in user_input.split(',')]
print(f"Entered list: {list}")

for num in list:
  if num%2 == 0:
    print(f"{num} is even number.")
```

Output

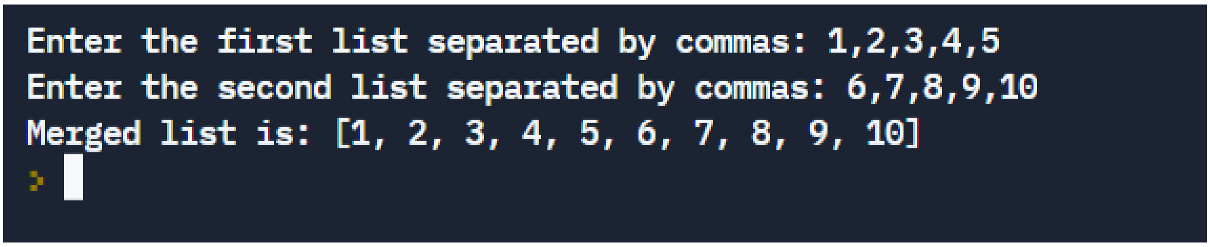


```
Enter the numbers searated by commas: 1, 2, 3, 4, 5
Entered list: [1, 2, 3, 4, 5]
2 is even number.
4 is even number.
>
```

6. Create two lists in python taking elements as input from user, merge those two lists.

```
first_user_input = input("Enter the first list separated by commas: ")
second_user_input = input("Enter the second list separated by commas: ")

first_list = [int(num) for num in first_user_input.split(',')]
second_list = [int(num) for num in second_user_input.split(',')]

merged_list = first_list + second_list
print(f"Merged list is: {merged_list}")
```

Output

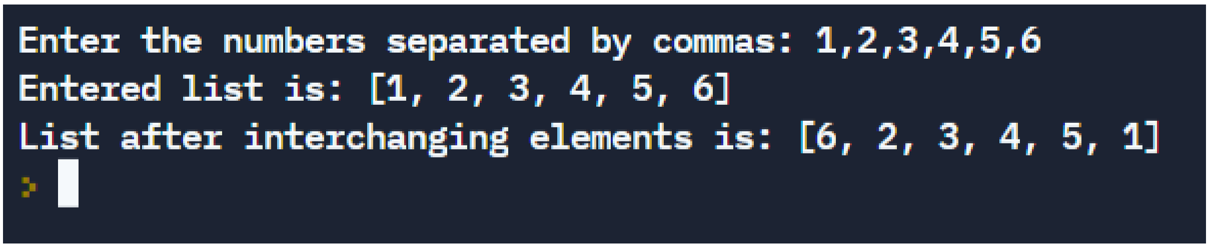


```
Enter the first list separated by commas: 1,2,3,4,5
Enter the second list separated by commas: 6,7,8,9,10
Merged list is: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>
```

7. Write a python program create a list based on user inputs interchange the first and last elements of the list.

```
user_input = input("Enter the numbers separated by commas: ")

list = [int(num) for num in user_input.split(',')]
print(f"Entered list is: {list}")

list[0], list[len(list) - 1] = list[len(list) - 1], list[0]

print(f"List after interchanging elements is: {list}")
```

Output



```
Enter the numbers separated by commas: 1,2,3,4,5,6
Entered list is: [1, 2, 3, 4, 5, 6]
List after interchanging elements is: [6, 2, 3, 4, 5, 1]
>
```

PROLOG

1. **The distance between different cities in KMs and the time taken by the bikers to cover the distance in hours is given below:-**

Source	**Destination**	**Distance(KMs)**	**Time(Hrs)**
Delhi	Mumbai	1414	23
Mumbai	Goa	587	14
Goa	Leh	2937	24
Leh	Kashmir	583	15

I. Write the clauses to create knowledge base in ProLog.

Knowledge Base:

```
distance(delhi, mumbai, 1414,23).
distance (mumbai, goa, 587, 14).
distance(goa, leh, 2937, 24).
distance(leh, kashmir, 583, 15).
```

II. Write query to determine the distance between Mumbai and Goa.

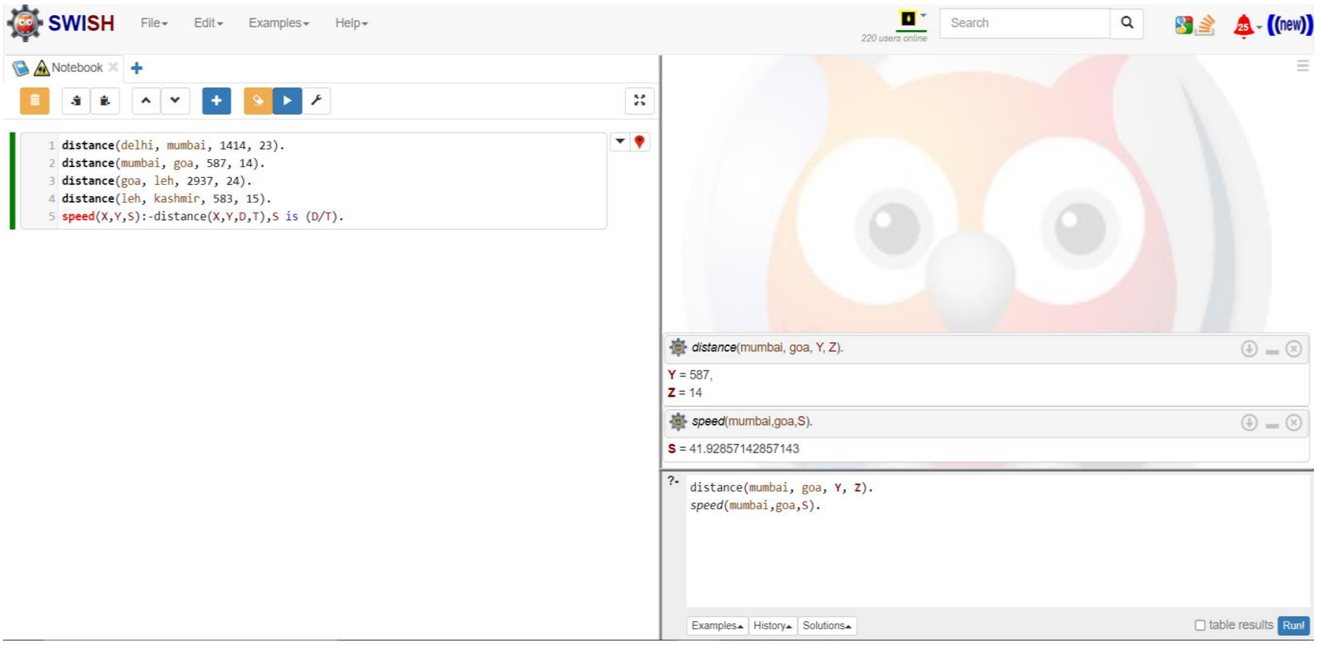
Query:

```
?-distance(mumbai, goa, Y, Z).
```

III. Establish a rule to determine the speed of biker between two given cities. Write query for the same.

Rule:

```
Speed(X,Y,S) :- distance(X,Y,D,T),
                S is (D/T).
```



```
distance(delhi, mumbai, 1414, 23).
distance(mumbai, goa, 587, 14).
distance(goa, leh, 2937, 24).
distance(leh, kashmir, 583, 15).
speed(X,Y,S):-distance(X,Y,D,T),S is (D/T).
```

```
distance(mumbai, goa, Y, Z).
Y = 587,
Z = 14
speed(mumbai,goa,S).
S = 41.92857142857143
```

```
?- distance(mumbai, goa, Y, Z).
   speed(mumbai,goa,S).
```

2. The linear measure, width and height of different living spaces in the house are given below.

Space	**Linear Measure**	**Width**	**Height**
Bedroom	10	12	18
Hall	12	20	18
Kitchen	8	8	18

I. Write Prolog program to create the knowledge base for the following.

Knowledge Base:

```
space(bedroom, 10, 12, 18).

space(hall, 12, 20, 18).

space(kitchen, 8, 8, 18).
```

II. Also design a rule to determine the volume of each individual living space.

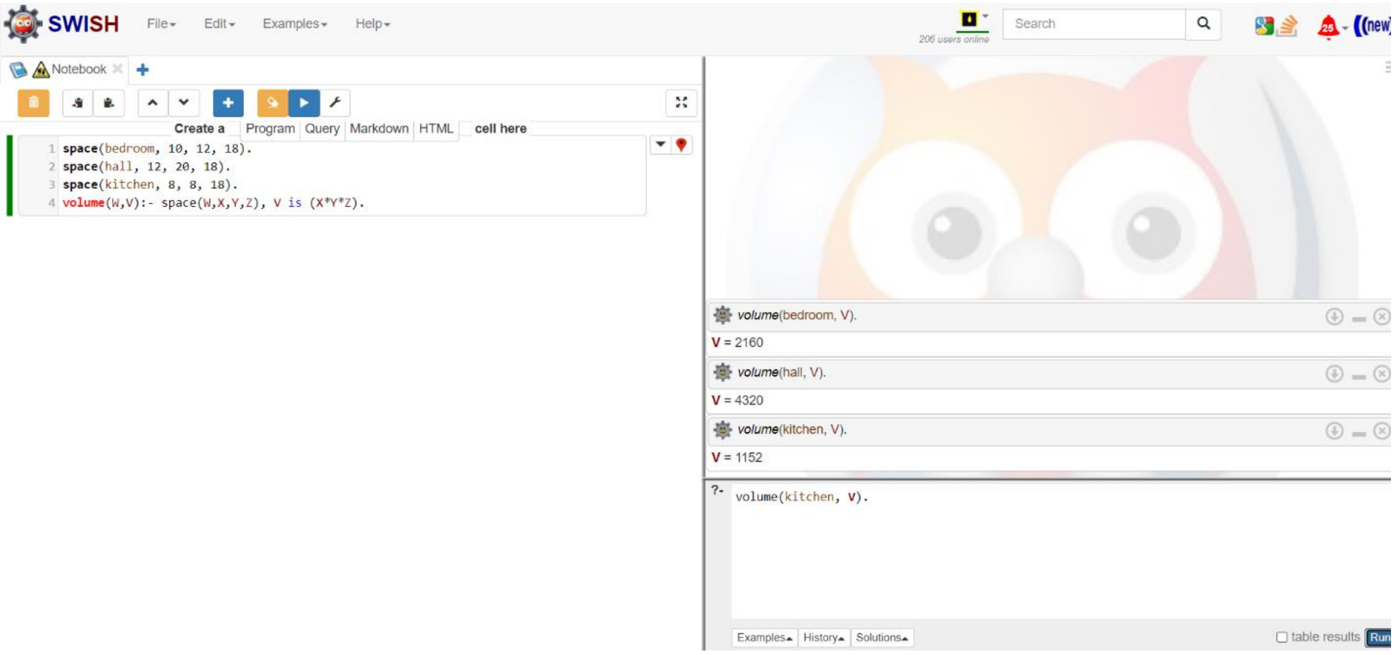
Rule:

```
volume(W,V):- space(W,X,Y,Z),
                      V is (W*Y*Z).
```

III. Write queries to find out the volume of individual living spaces.

Query:

?-volume(bedroom, V).
?-volume(hall, V).
?-volume(kitchen, V).



```
space(bedroom, 10, 12, 18).
space(hall, 12, 20, 18).
space(kitchen, 8, 8, 18).
volume(W,V):- space(W,X,Y,Z), V is (X*Y*Z).
```

```
volume(bedroom, V).
V = 2160
volume(hall, V).
V = 4320
volume(kitchen, V).
V = 1152
```

```
?- volume(kitchen, V).
```

3. **Following details about the employees of an organization-**

Name	**EmpId**	**Dept**	**Salary**
Alex	20	Marketing	25000
Ronald	24	Marketing	25000
Fred	21	IT	30000
George	22	HR	50000

I. **How will you represent the knowledge base using structure in ProLog.**

Knowledge Base using Structure:

details(name(alex), empid(20), dept(marketing), salary(25000)).

details(name(ronald), empid(24), dept(marketing), salary(25000)).

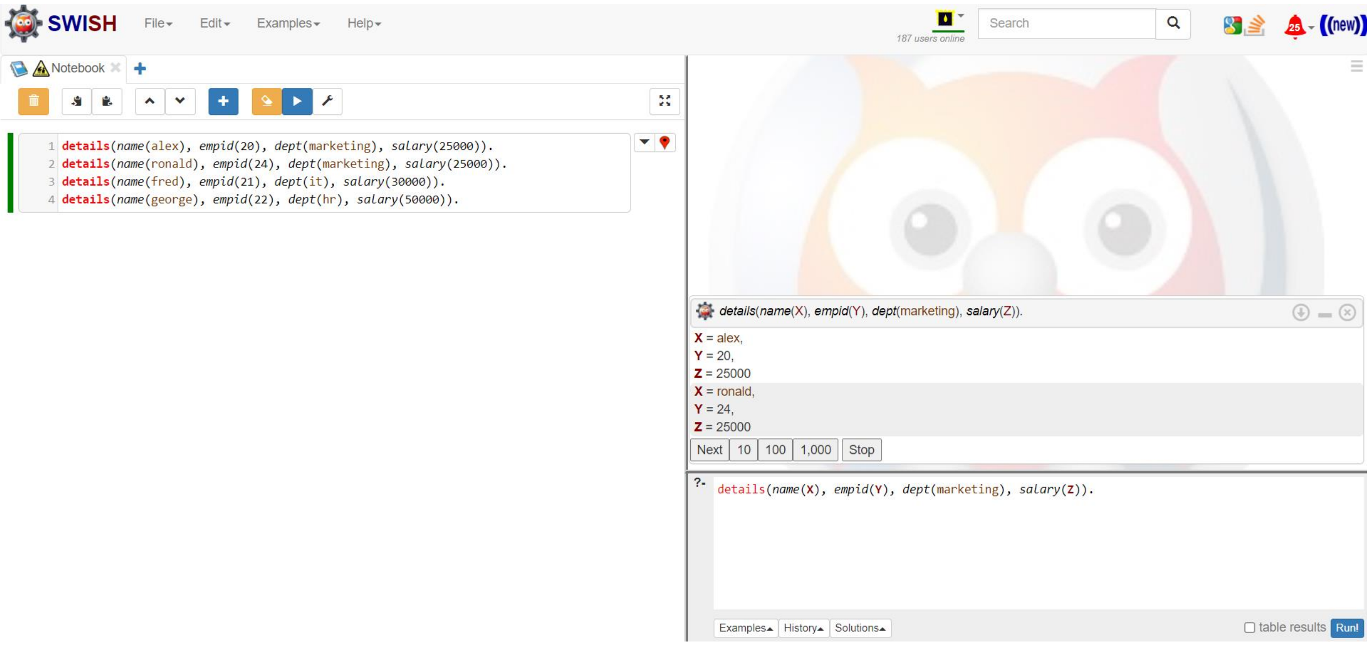
details(name(fred), empid(21), dept(it), salary(30000)).

details(name(george), empid(22), dept(hr), salary(50000)).

II. Write query to display employees of marketing department.

Query:

details(name(X), empid(Y), dept(marketing), salary(Z)).



```
details(name(alex), empid(20), dept(marketing), salary(25000)).
details(name(ronald), empid(24), dept(marketing), salary(25000)).
details(name(fred), empid(21), dept(it), salary(30000)).
details(name(george), empid(22), dept(hr), salary(50000)).
```

```
details(name(X), empid(Y), dept(marketing), salary(Z)).
X = alex,
Y = 20,
Z = 25000
X = ronald,
Y = 24,
Z = 25000
```

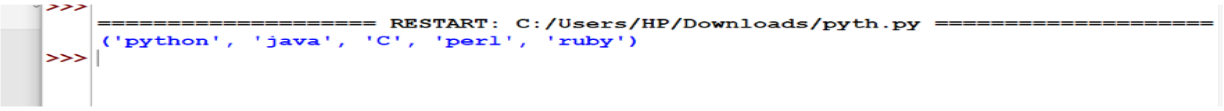
```
?- details(name(X), empid(Y), dept(marketing), salary(Z)).
```

Lab Assignment-2

Tuples in Python

#Printing Tuple

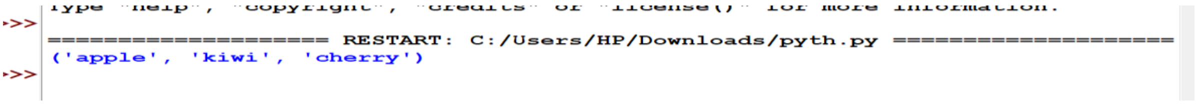
```
myTuple = (“python”, “java”, “C#”, “perl”, “ruby”)
print(myTuple)
```



```
>>>
==================== RESTART: C:/Users/HP/Downloads/pyth.py ====================
('python', 'java', 'C', 'perl', 'ruby')
>>> |
```

#Updating in Tuple

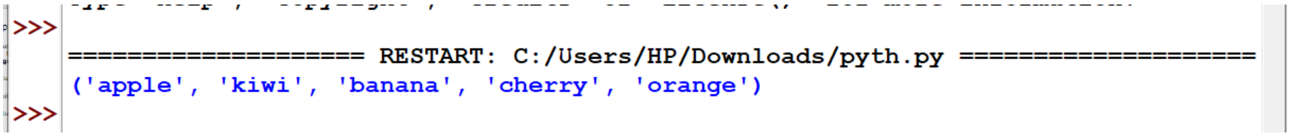
```
x = ("apple", "banana", "cherry")
y = list(x)
y[1] = "kiwi"
x = tuple(y)
print(x)
```



```
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==================== RESTART: C:/Users/HP/Downloads/pyth.py ====================
('apple', 'kiwi', 'cherry')
>>>
```

#Inserting in Tuple

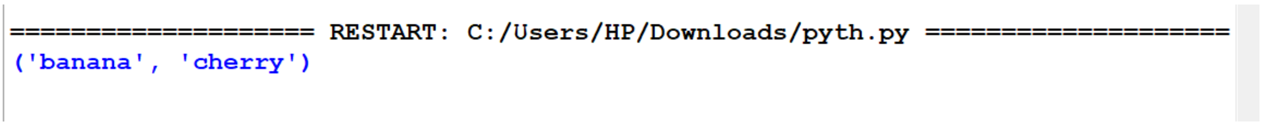
```
x = ("apple", "banana", "cherry")
y = list(x)
y.append("orange")
y.insert(1,"kiwi")
x = tuple(y)
print(x)
```



```
>>>
==================== RESTART: C:/Users/HP/Downloads/pyth.py ====================
('apple', 'kiwi', 'banana', 'cherry', 'orange')
>>>
```

#Remove and Delete Items From Tuple

```
x = ("apple", "banana", "cherry")
y = list(x)
y.remove("apple")
x = tuple(y)
print(x)
```



```
==================== RESTART: C:/Users/HP/Downloads/pyth.py ====================
('banana', 'cherry')
```

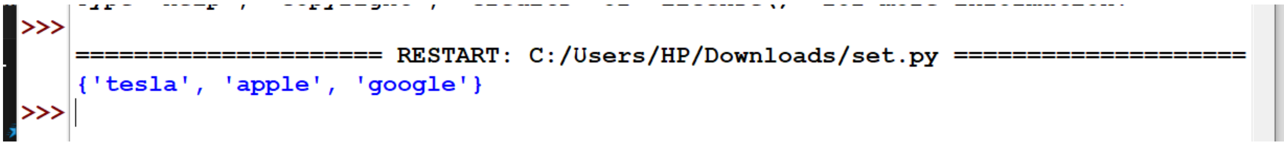
# The del keyword can be used to delete the tuple completely

```
x = ("apple", "banana", "cherry")
del x
```

Set in Python

#Printing Set

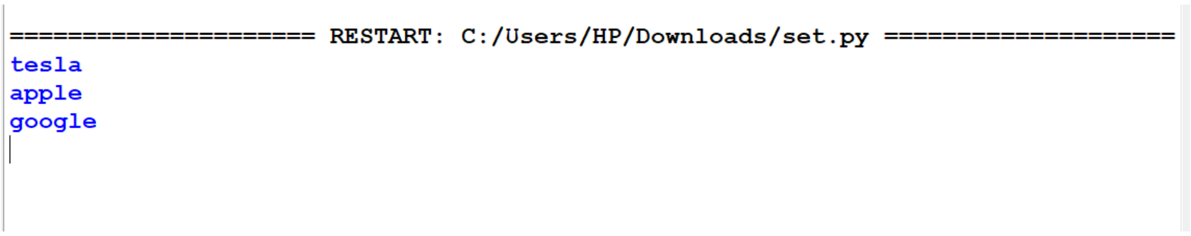
```
thisset = {"apple","tesla","google"}
print(thisset)
```



```
>>>
    ===================== RESTART: C:/Users/HP/Downloads/set.py ====================
    {'tesla', 'apple', 'google'}
>>>
```

#Access Set Items

```
thisset = {"apple","tesla","google"}
for x in thisset:
    print(x)
```



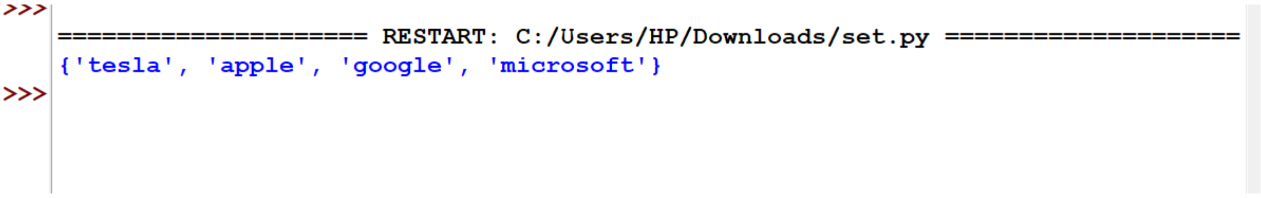
```
===================== RESTART: C:/Users/HP/Downloads/set.py ====================
tesla
apple
google
```

#Adding items to a set

thisset = {"apple","tesla","google"}

thisset.add("microsoft")

print(thisset)



```
>>>
==================== RESTART: C:/Users/HP/Downloads/set.py ====================
{'tesla', 'apple', 'google', 'microsoft'}
>>>
```

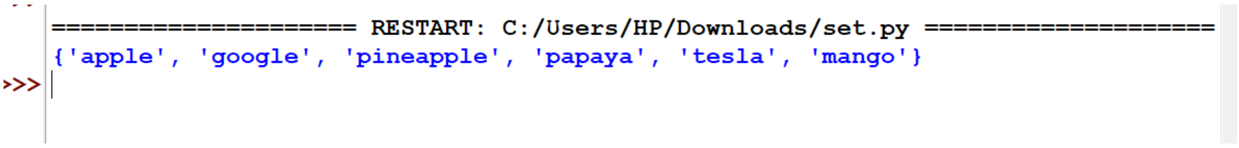
#Adding two sets

thisset = {"apple","tesla","google"}

tropical = {"pineapple","mango","papaya"}

thisset.update(tropical)

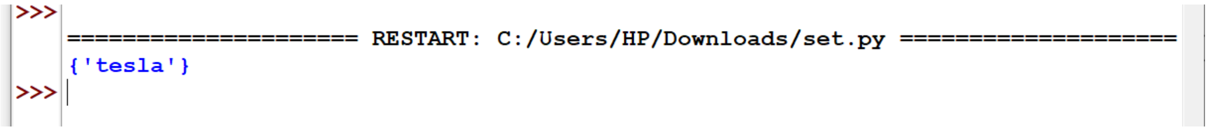
print(thisset)



```
==================== RESTART: C:/Users/HP/Downloads/set.py ====================
{'apple', 'google', 'pineapple', 'papaya', 'tesla', 'mango'}
>>>
```

#Removing Set Items

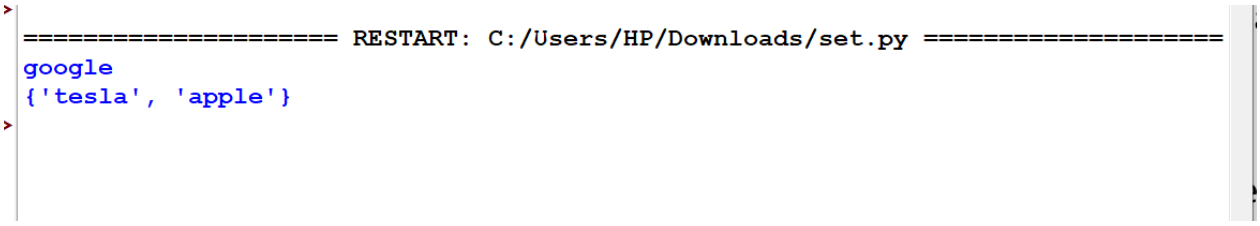
```
thisset = {"apple","tesla","google"}
thisset.remove("apple")
thisset.discard("google")
print(thisset)
```



```
>>>
===================== RESTART: C:/Users/HP/Downloads/set.py ====================
{'tesla'}
>>>
```

#Using Pop() Method

```
thisset = {"apple","tesla","google"}
x=thisset.pop()
print(x)
print(thisset)
```

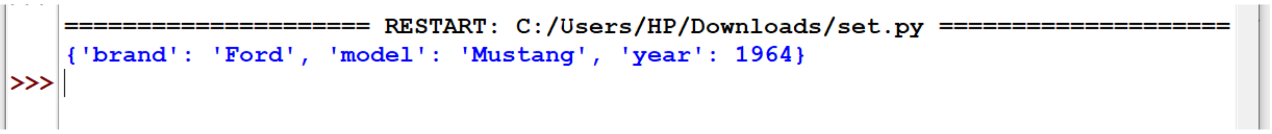


```
>
===================== RESTART: C:/Users/HP/Downloads/set.py ====================
google
{'tesla', 'apple'}
>
```

Python Dictionaries

#Printing Dictionary

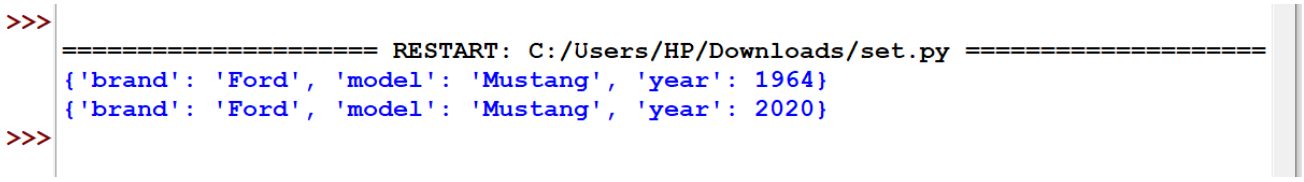
```
thisdict = {"brand": "Ford",
            "model": "Mustang",
            "year": 1964}
print(thisdict)
```



```
==================== RESTART: C:/Users/HP/Downloads/set.py ====================
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
>>> 
```

#Updating Dictionary

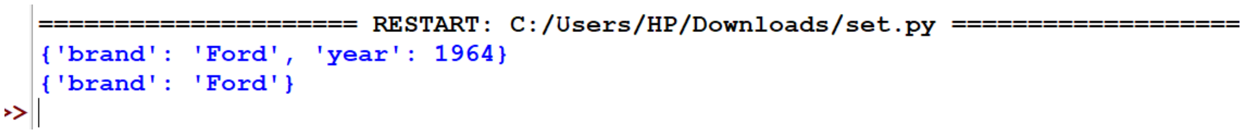
```
thisdict = {"brand": "Ford",
            "model": "Mustang",
            "year": 1964}
print(thisdict) #before update
thisdict.update({"year": 2020})
print(thisdict) #after update
```



```
>>>
    ===================== RESTART: C:/Users/HP/Downloads/set.py ====================
    {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
    {'brand': 'Ford', 'model': 'Mustang', 'year': 2020}
>>>
```

#Removing Items from Dictionary

```
thisdict = {"brand": "Ford",
            "model": "Mustang",
            "year": 1964}
thisdict.pop("model")
print(thisdict) #after poping "model"
thisdict.popitem()
print(thisdict)
```

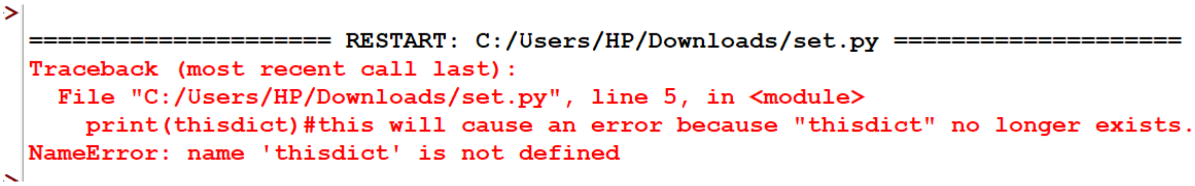


```
===================== RESTART: C:/Users/HP/Downloads/set.py ===================
{'brand': 'Ford', 'year': 1964}
{'brand': 'Ford'}
>>
```

#Using Del keyword

```
thisdict = {"brand": "Ford",
            "model": "Mustang",
            "year": 1964}
del thisdict
```

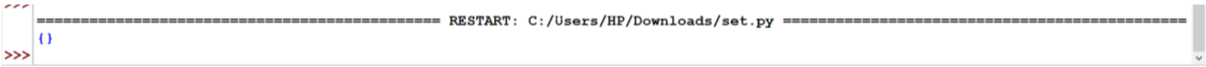
print(thisdict) **#this will cause an error because "thisdict" no longer exists.**



```
===================== RESTART: C:/Users/HP/Downloads/set.py ====================
Traceback (most recent call last):
  File "C:/Users/HP/Downloads/set.py", line 5, in <module>
    print(thisdict)#this will cause an error because "thisdict" no longer exists.
NameError: name 'thisdict' is not defined
```

#Using clear() method

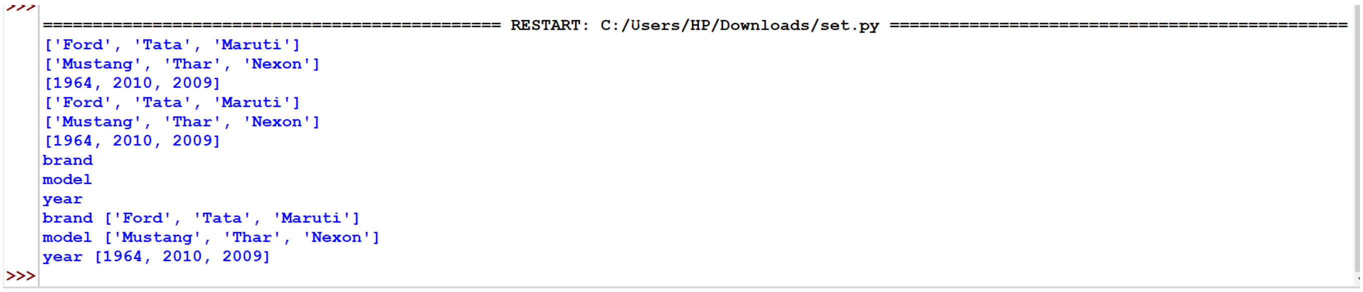
```
thisdict = {"brand": "Ford",
            "model": "Mustang",
            "year": 1964}
thisdict.clear()
print(thisdict)
```



```
============================================ RESTART: C:/Users/HP/Downloads/set.py ============================================
{}
>>>
```

#Looping through a dictionary

```
thisdict = {"brand": ["Ford","Tata","Maruti"],
            "model": ["Mustang","Thar","Nexon"],
            "year": [1964,2010,2009]
            }

for x in thisdict:
    print(thisdict[x]) #Print all the values in the dictionary one by one
for x in thisdict.values():
    print(x) #The values() method can be used to return all the values.
for x in thisdict.keys():
    print(x) #keys() method can be used to print all the keys of a dictionary.
for x, y in thisdict.items():
    print(x, y) #To print both the keys and values.
```

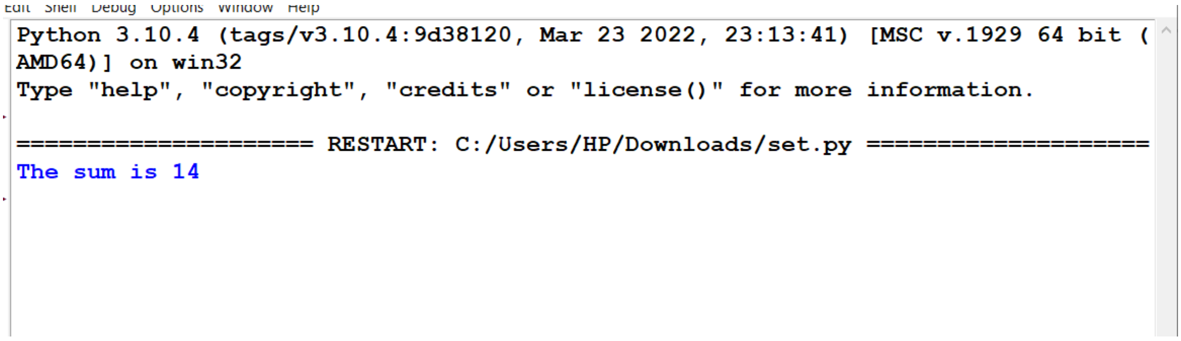


```
>>>
============================================ RESTART: C:/Users/HP/Downloads/set.py ============================================
['Ford', 'Tata', 'Maruti']
['Mustang', 'Thar', 'Nexon']
[1964, 2010, 2009]
['Ford', 'Tata', 'Maruti']
['Mustang', 'Thar', 'Nexon']
[1964, 2010, 2009]
brand
model
year
brand ['Ford', 'Tata', 'Maruti']
model ['Mustang', 'Thar', 'Nexon']
year [1964, 2010, 2009]
>>>
```

Python Function

1. Function that calculates sum of two numbers

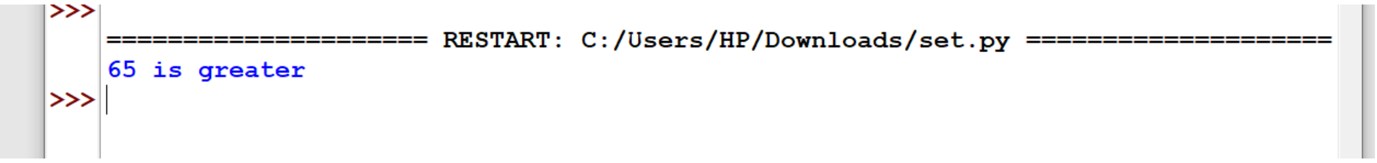
```
def sum(x,y):
    return x+y


x=5
y=9
sum = sum(x,y)
print("The sum is", sum)
```



```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

===================== RESTART: C:/Users/HP/Downloads/set.py ====================
The sum is 14
```

2. Function that finds the maximum of two numbers

```
def maximum(x,y):
    if(x>y):
        print(x, "is greater")
    else:
        print(y, "is greater")



x=65
y=34
maximum(x,y)
```



```
>>>
==================== RESTART: C:/Users/HP/Downloads/set.py ====================
65 is greater
>>>
```

Lab Assignment-3

Natural Language Processing

Text Pre Processing

- ### Tokenization

```
import nltk
import pandas as pd
data=pd.read_pd(“data.csv”)
sentences = nltk.sent_tokenize(paragraph)
for sentence in sentences:
 words=nltk.word_tokenize(sentence)
```

- ### Lemmatization

```
import nltk
import pandas as pd
data=pd.read_pd(“data.csv”)
sentences = nltk.sent_tokenize(paragraph)
lemmatizer = WordNetLemmatizer()

# Lemmatization
for i in range(len(sentences)):
    words = nltk.word_tokenize(sentences[i])
    words = [lemmatizer.lemmatize(word) for word in words if word not in set(stopwords.words('english'))]
    sentences[i] = ' '.join(words)
```

- ## Stop Words Removal

```
import nltk
import pandas as pd
data=pd.read_pd(“data.csv”)
# Preprocessing the data
text = re.sub(r'\[[0-9]*\]',' ',paragraph)
text = re.sub(r'\s+',' ',text)
text = text.lower()
text = re.sub(r'\d',' ',text)
text = re.sub(r'\s+',' ',text)

# Preparing the dataset
sentences = nltk.sent_tokenize(text)
sentences = [nltk.word_tokenize(sentence) for sentence in sentences]
for i in range(len(sentences)):
    sentences[i] = [word for word in sentences[i] if word not in stopwords.words('english')]
```

Vectorization

- ## Bag Of Words

```
import nltk
import pandas as pd
data=pd.read_pd(“data.csv”)
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer

ps = PorterStemmer()
wordnet=WordNetLemmatizer()
sentences = nltk.sent_tokenize(paragraph)
corpus = []
for i in range(len(sentences)):
    review = re.sub('[^a-zA-Z]', ' ', sentences[i])
    review = review.lower()
    review = review.split()
    review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
```

- ## TF-IDF

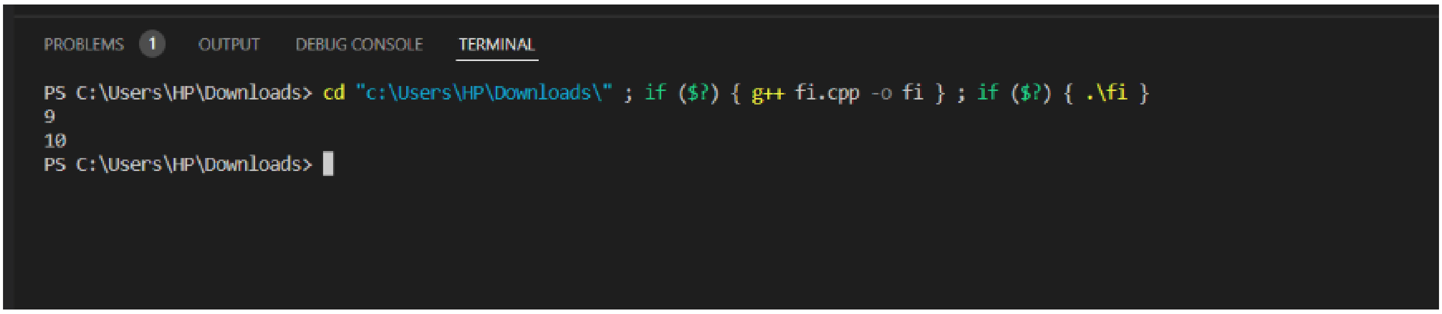
```
import nltk
import pandas as pd
data=pd.read_pd(“data.csv”)

import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer

ps = PorterStemmer()
wordnet=WordNetLemmatizer()
sentences = nltk.sent_tokenize(paragraph)
corpus = []
for i in range(len(sentences)):
    review = re.sub('[^a-zA-Z]', ' ', sentences[i])
    review = review.lower()
    review = review.split()
    review = [wordnet.lemmatize(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)

# Creating the TF-IDF model
from sklearn.feature_extraction.text import TfidfVectorizer
cv = TfidfVectorizer()
X = cv.fit_transform(corpus).toarray()
```

Lab Assignment-4

Generic Programming

1. **Write a Generic function to add two numbers.**

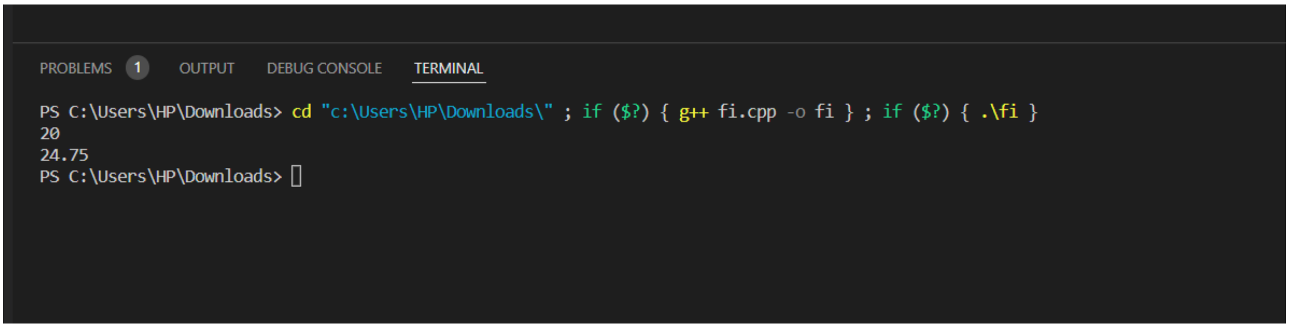
```
#include<iostream>
using namespace std;
template<typename T>T sum(T x, T y)
{
    return x+y;
}
int main()
{
    cout<<sum(4,5)<<endl;
    cout<<sum(4.5,5.5)<<endl;
    return 0;
}
```



```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
9
10
PS C:\Users\HP\Downloads> ▌
```

2. Write a generic function for product of two numbers.

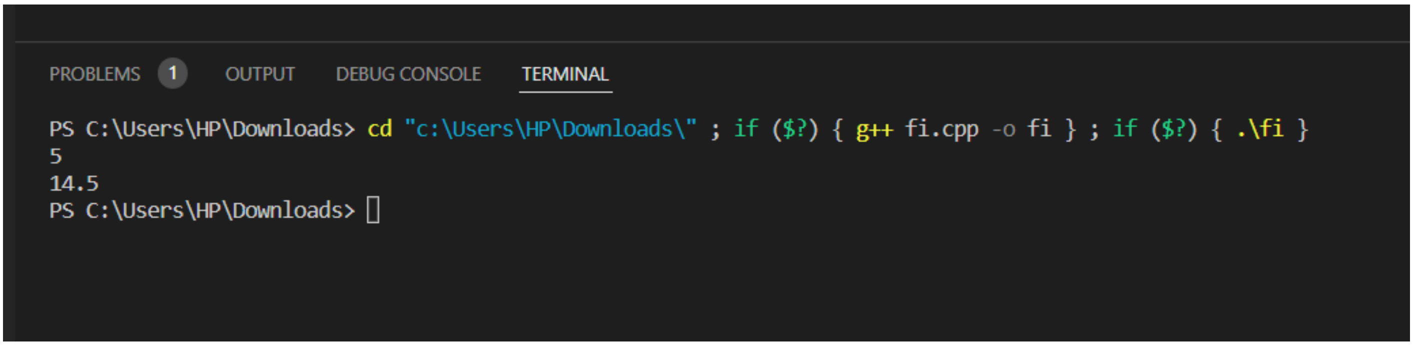
```
#include<iostream>
using namespace std;
template<typename T>T product(T x, T y)
{
    return x*y;
}
int main()
{
    cout<< product(4,5)<<endl;
    cout<< product(4.5,5.5)<<endl;
    return 0;
}
```



```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
20
24.75
PS C:\Users\HP\Downloads> ▯
```

3. Write a Generic Function to find the maximum of two numbers.

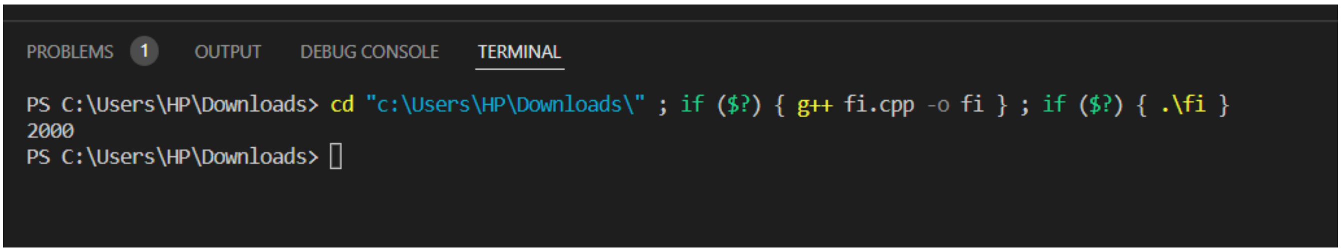
```
#include<iostream>
using namespace std;
template<typename T>T max(T x, T y)
{
    return (x>y)?x:yx;
}
int main()
{
    cout<<max(4,5)<<endl;
    cout<<max(14.5,5.5)<<endl;
    return 0;
}
```



```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
5
14.5
PS C:\Users\HP\Downloads> 
```

4. Write a generic function for simple interest which work for every bank.

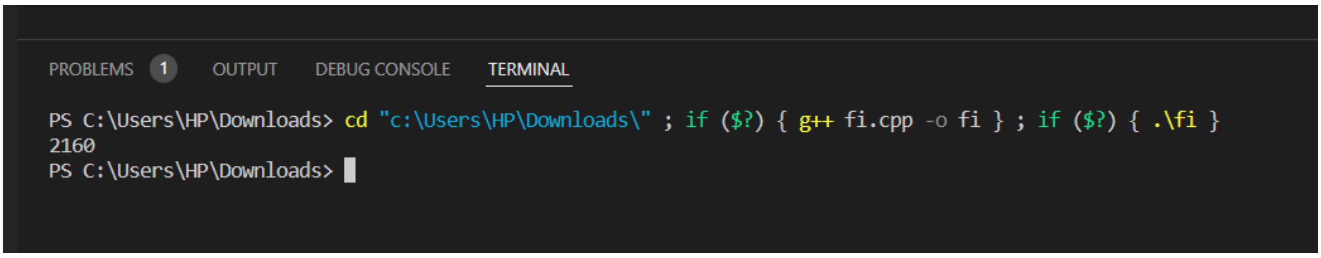
```
#include<iostream>
using namespace std;
template<typename T>T interest(T x, T y, T z)
{
    return (x*y*z)/100;
}
int main()
{
    cout<<interest(4000,10,5)<<endl;
    return 0;
}
```



```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
2000
PS C:\Users\HP\Downloads> 
```

5. Write a generic function for volume of house

```
#include<iostream>
using namespace std;
template<typename T>T volume(T l, T b, T h)
{
    return l*b*h;
}
int main()
{
    cout<<volume(12,15,12)<<endl;
    return 0;
}
```

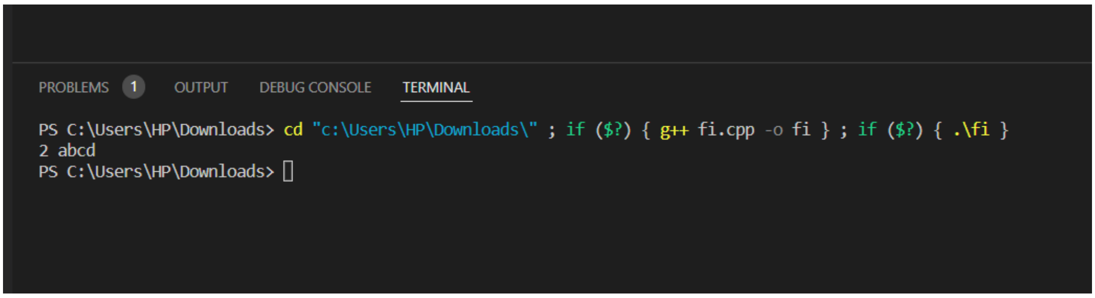


```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
2160
PS C:\Users\HP\Downloads> 
```

Standard Template Library

Pairs in STL

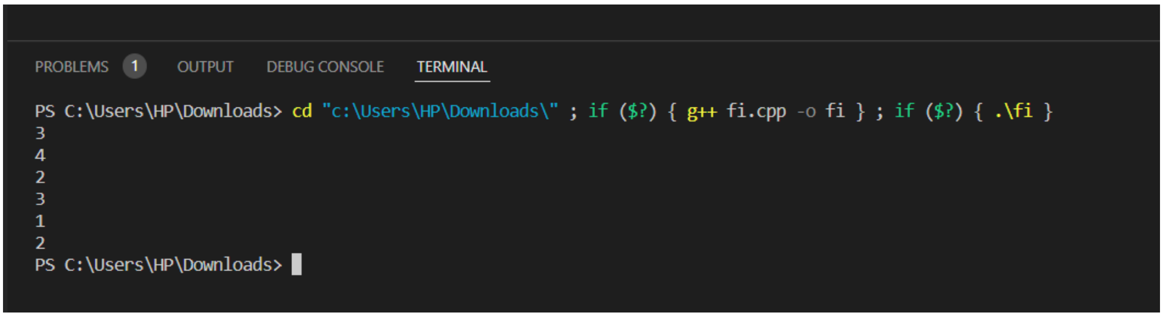
```
#include<iostream>
using namespace std;
int main()
{
   pair<int,string>p;
   p={2,"abcd"};
   cout<<p.first<<" "<<p.second<<endl;
   return 0;
}
```



```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
2 abcd
PS C:\Users\HP\Downloads> 
```

//Pair Array Swap

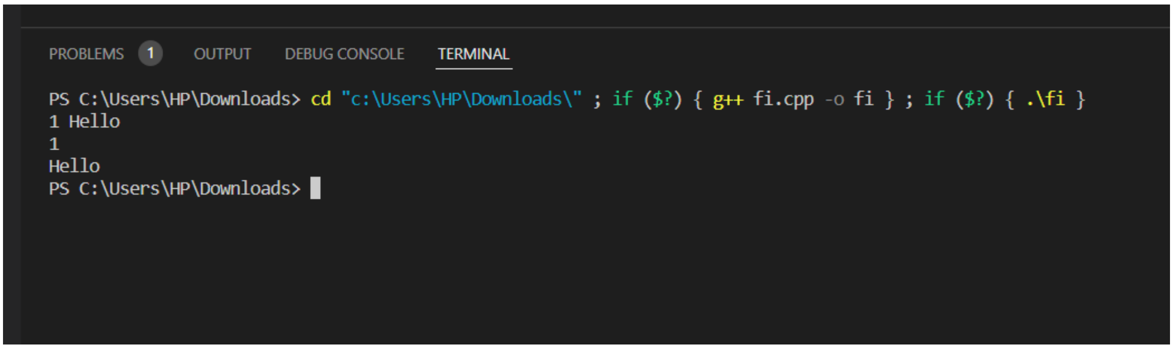
```
#include<iostream>
using namespace std;
int main()
{
   pair<int,int>p[3];
   p[0] = {1,2};
   p[1] = {2,3};
   p[2] = {3,4};
   swap(p[0], p[2]);
   for(int i=0; i<3; ++i)
   {
       cout<<p[i].first<<endl;
       cout<<p[i].second<<endl;
   }
   return 0;
}
```



```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
3
4
2
3
1
2
PS C:\Users\HP\Downloads> 
```

//User Input

```
#include<iostream>
using namespace std;
int main()
{
   pair<int,string>p;

   cin>>p.first;
   cin>>p.second;

   cout<<p.first<<endl;
   cout<<p.second<<endl;
   return 0;
}
```

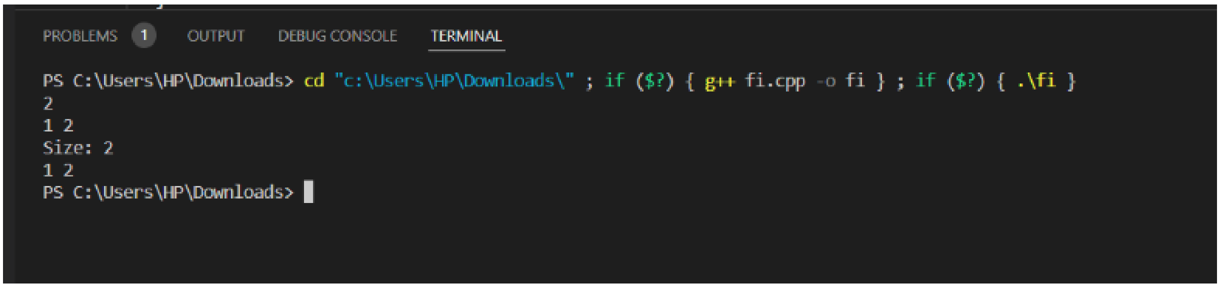


```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
1 Hello
1
Hello
PS C:\Users\HP\Downloads> 
```

Vectors in STL

```
#include<iostream>
#include<vector>
using namespace std;
void printVec(vector<int>v)
{
    cout<<"Size: "<<v.size()<<endl;
    for(int i=0; i<v.size(); ++i)
    {
        cout<<v[i]<<" ";
    }
    cout<<endl;
}
int main()
{
  vector<int>v;
  int n;
  cin>>n;

  for(int i=0; i<n; ++i)
  {
      int x;
```

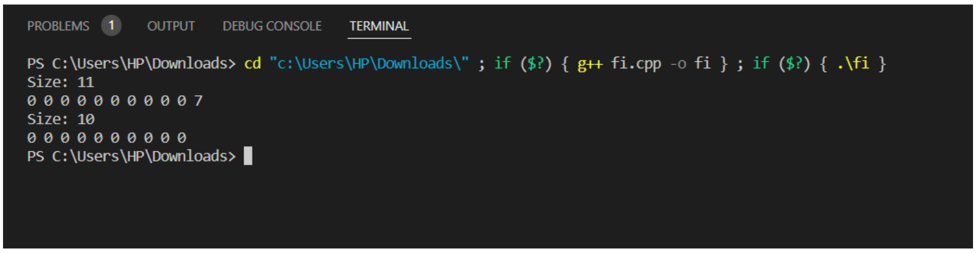
```
        cin>>x;
        v.push_back(x);
    }
    printVec(v);
    return 0;
}
```



```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
2
1 2
Size: 2
1 2
PS C:\Users\HP\Downloads> 
```

//Declare Size of Vector with PushBack and PopBack Operation

```
#include<iostream>
#include<vector>
using namespace std;
void printVec(vector<int>v)
{
    cout<<"Size: "<<v.size()<<endl;
    for(int i=0; i<v.size(); ++i)
    {
        cout<<v[i]<<" ";
    }
    cout<<endl;
}
int main()
{
  vector<int>v(10);
  v.push_back(7);
  printVec(v);
  v.pop_back();
  printVec(v);
  return 0;
}
```

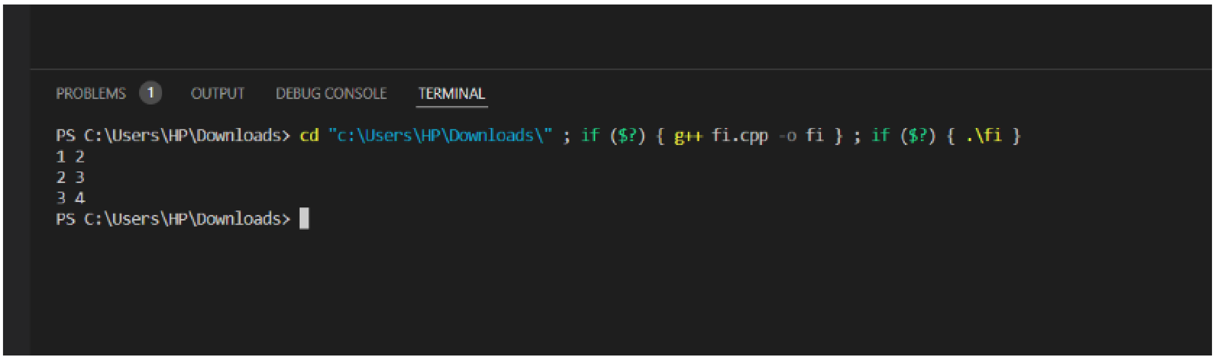


```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
Size: 11
0 0 0 0 0 0 0 0 0 0 7
Size: 10
0 0 0 0 0 0 0 0 0 0
PS C:\Users\HP\Downloads> 
```

Iterators in STL

//Vector of Pair Iterator

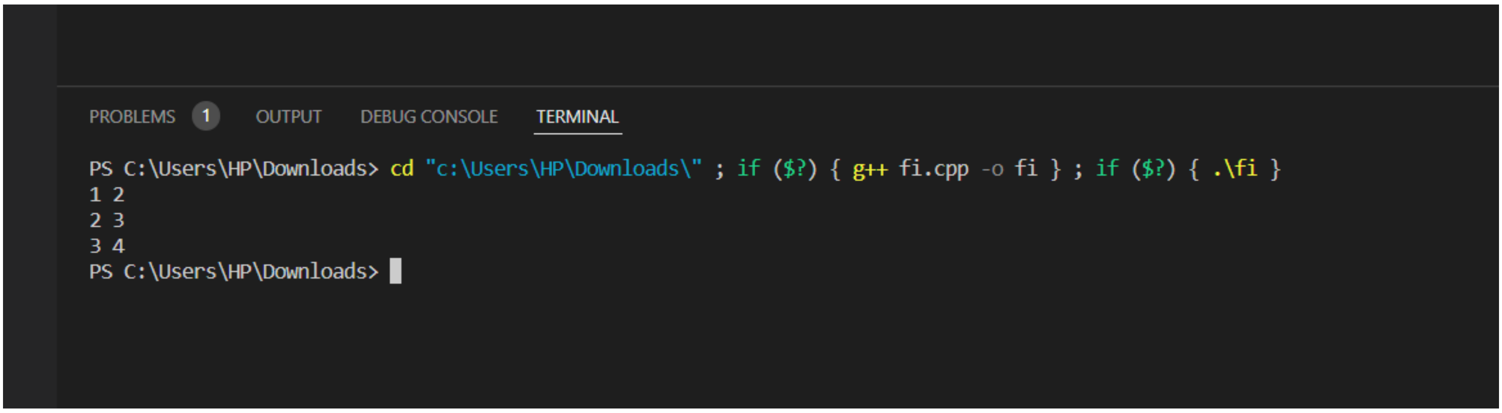
```
#include<bits/stdc++.h>
using namespace std;
int main()
{
  vector<pair<int,int>>v_p{{1,2}, {2,3}, {3,4}};
  vector<pair<int,int>> :: iterator it;

  for(it=v_p.begin(); it!=v_p.end(); ++it)
  {
      cout<<(*it).first<<" "<<(*it).second<<endl;
  }
}
```



```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
1 2
2 3
3 4
PS C:\Users\HP\Downloads> 
```

// Vector of Pair Range Based Loop

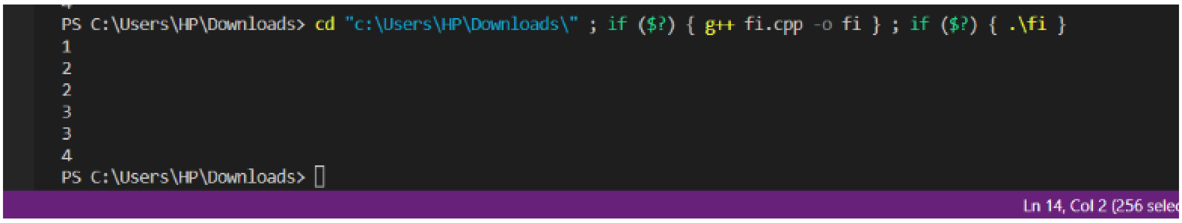
```
#include<bits/stdc++.h>
using namespace std;
int main()
{
  vector<pair<int,int>>v_p{{1,2}, {2,3}, {3,4}};
  for(pair<int,int> &value : v_p)
  {
      cout<<value.first<<" "<<value.second<<endl;
  }
}
```



```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
1 2
2 3
3 4
PS C:\Users\HP\Downloads> 
```

//AutoKeyword

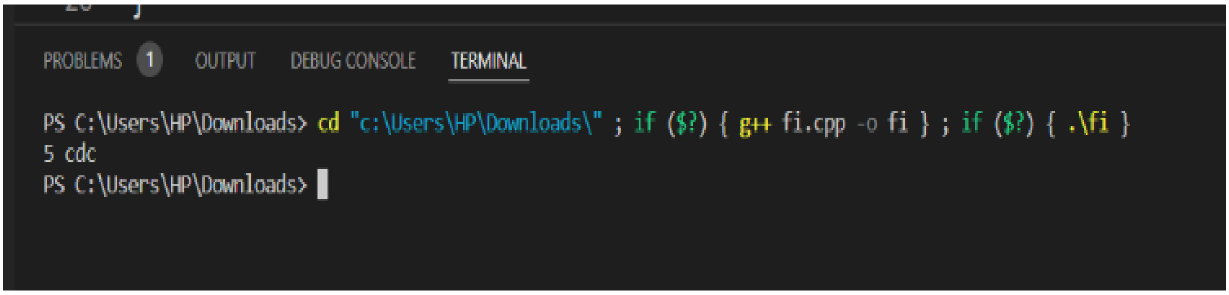
```
#include<bits/stdc++.h>
using namespace std;
int main()
{

  vector<pair<int,int>>vp={{1,2}, {2,3}, {3,4}};

  for(auto it = vp.begin(); it!=vp.end(); ++it)
  {
      cout<<(*it).first<<endl;
      cout<<(*it).second<<endl;
  }
  return 0;
}
```



```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
1
2
2
3
3
4
PS C:\Users\HP\Downloads> 
```

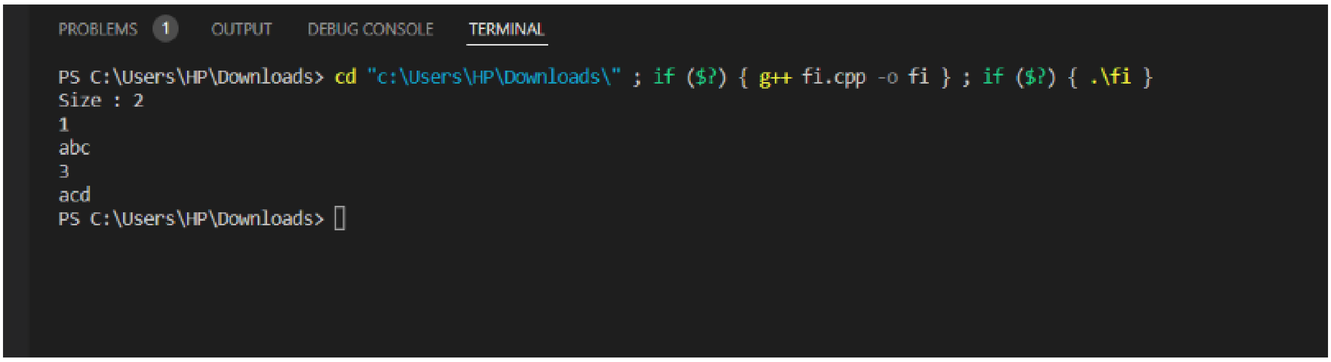
Maps in STL

```
//Find Operation
#include<bits/stdc++.h>
using namespace std;
void print(map<int,string>&m)
{
    cout<<"Size : "<<m.size()<<endl;
    for(auto &pr :m)
    {
        cout<<pr.first<<endl;
        cout<<pr.second<<endl;
    }
}
int main()
{
  map<int,string>m;
  m[1] = "abc";
  m[5] = "cdc";
  m[3] = "acd";

  auto it = m.find(5);
  if(it == m.end())
  cout<<"No Such Value";
  else
  cout<<(*it).first<<" "<<(*it).second<<endl;
  return 0;
}
```



```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
5 cdc
PS C:\Users\HP\Downloads> 
```

```
//Erase Operation
#include<bits/stdc++.h>
using namespace std;
void print(map<int,string>&m)
{
    cout<<"Size : "<<m.size()<<endl;
    for(auto &pr :m)
    {
        cout<<pr.first<<endl;
        cout<<pr.second<<endl;
    }
}
int main()
{
  map<int,string>m;
  m[1] = "abc";
  m[5] = "cdc";
  m[3] = "acd";

  auto it = m.find(5);

  if(it != m.end())
  m.erase(it);
  else
  cout<<"No Such Value";

  print(m);
}
```

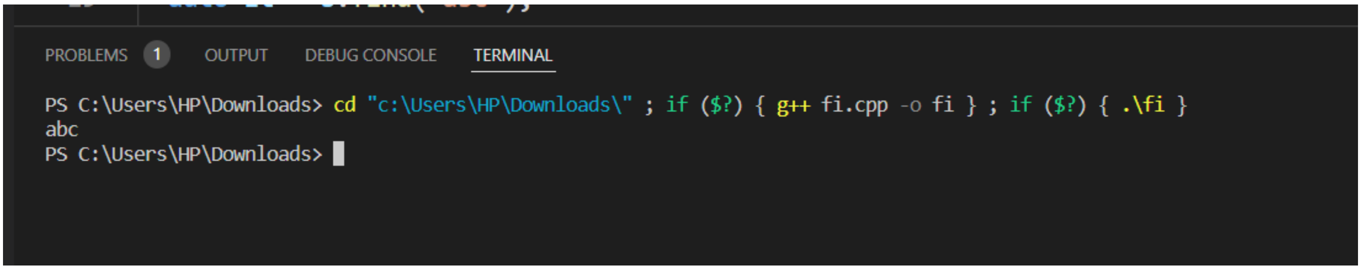


```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
Size : 2
1
abc
3
acd
PS C:\Users\HP\Downloads> 
```

SET in STL

//Find Operation

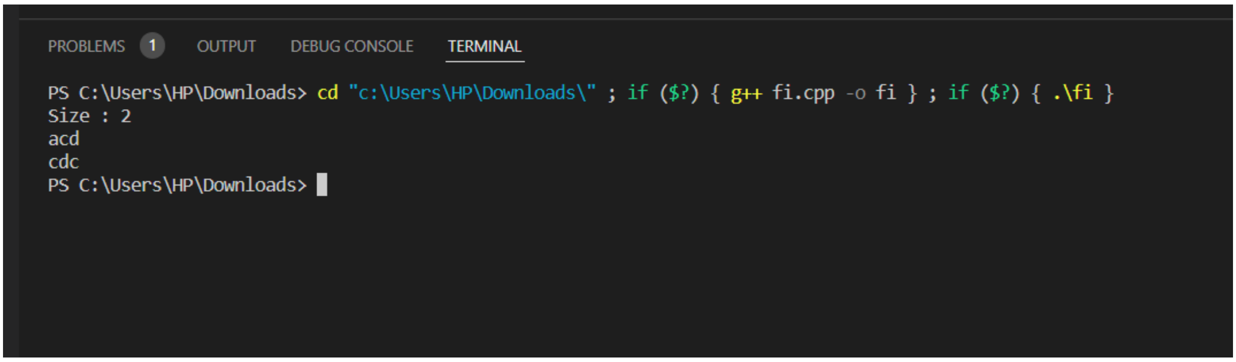
```
#include<bits/stdc++.h>
using namespace std;
void print(set<string> &s)
{
    cout<<"Size : "<<s.size()<<endl;
    for(string value : s)
    {
        cout<<value<<endl;
    }
}

int main()
{
  set<string>s;
  s.insert("abc");
  s.insert("cdc");
  s.insert("acd");

  auto it = s.find("abc");
  if(it != s.end())
  cout<<(*it);
  else
  cout<<"No Such Value";


}
```



```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
abc
PS C:\Users\HP\Downloads> 
```

//Erase Operation

```
#include<bits/stdc++.h>
using namespace std;
void print(set<string> &s)
{
    cout<<"Size : "<<s.size()<<endl;
    for(string value : s)
    {
        cout<<value<<endl;
    }
}

int main()
{
  set<string>s;
  s.insert("abc");
  s.insert("cdc");
  s.insert("acd");

  auto it = s.find("abc");

  if(it != s.end())
  s.erase(it);
  else
  cout<<"No Such Value";

  print(s);

}
```



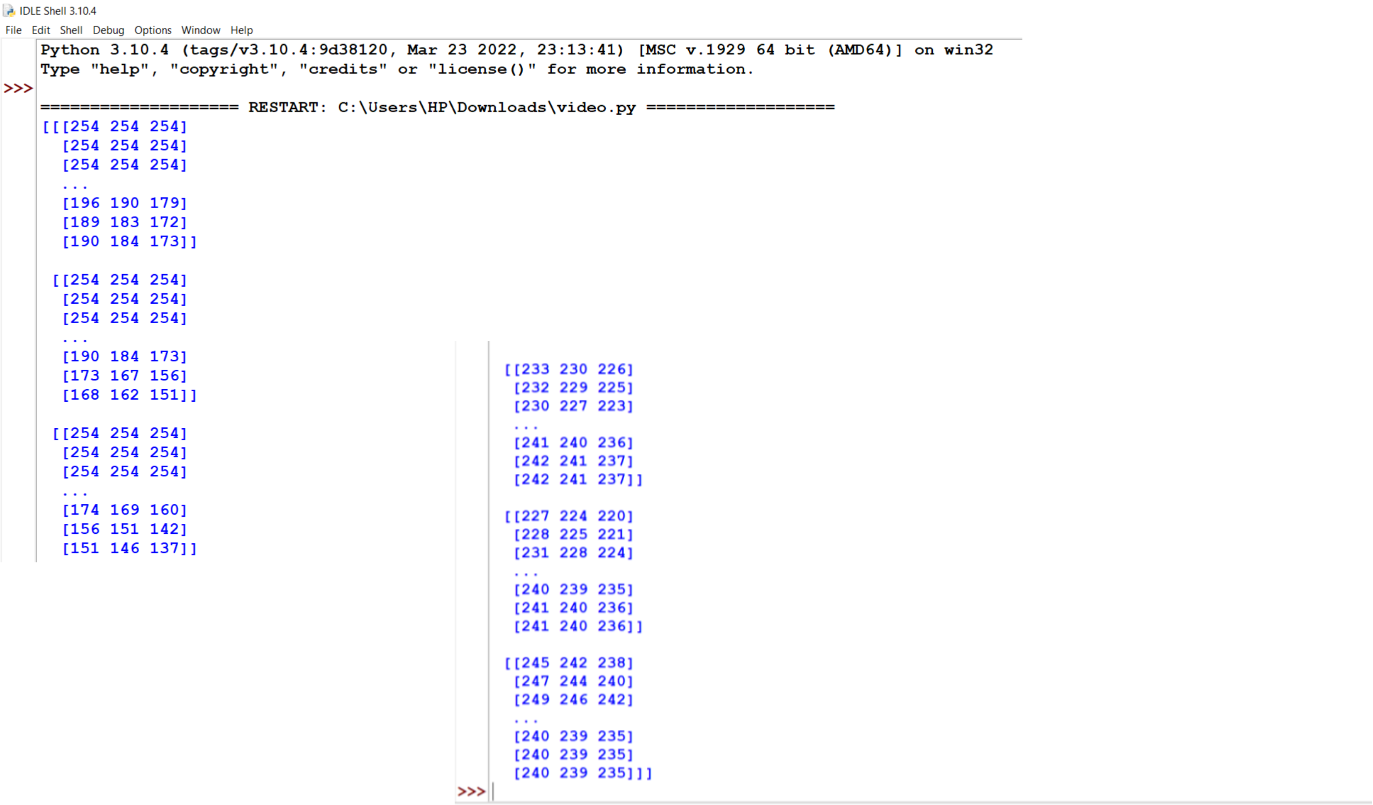
```
PS C:\Users\HP\Downloads> cd "c:\Users\HP\Downloads\" ; if ($?) { g++ fi.cpp -o fi } ; if ($?) { .\fi }
Size : 2
acd
cdc
PS C:\Users\HP\Downloads> 
```

Lab Assignment-5

OPEN CV

1. Reading an image and printing it.

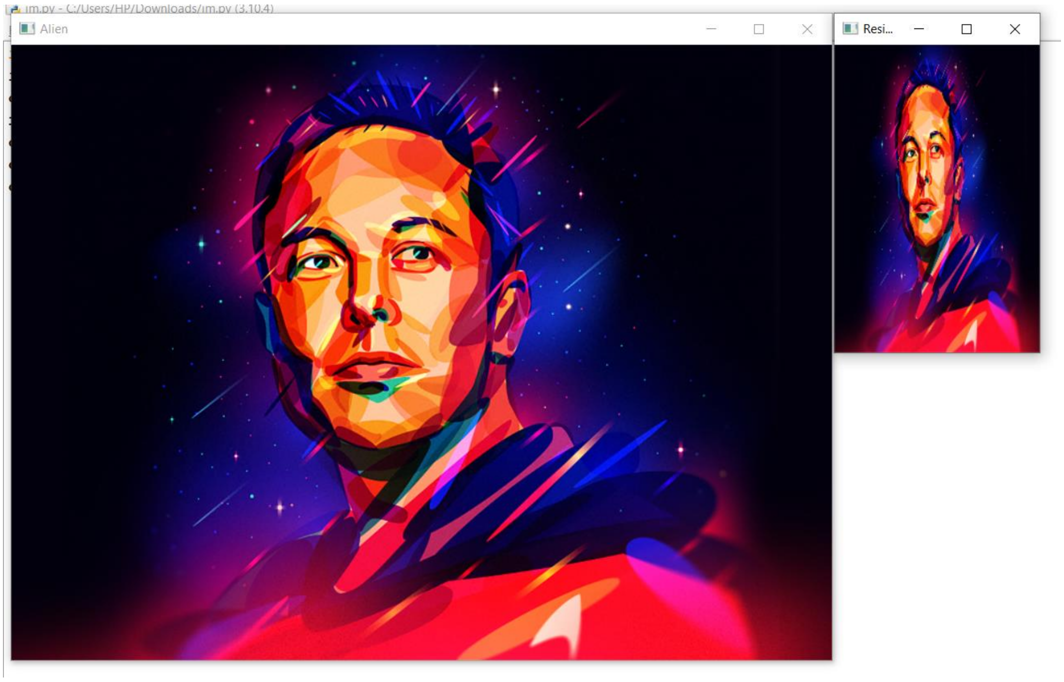
```
import cv2
img = cv2.imread("E:\PHOTOS\shinchain.jpg", 1)
print(img)
```



```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==================== RESTART: C:\Users\HP\Downloads\video.py ===================
[[[254 254 254]
  [254 254 254]
  [254 254 254]
  ...
  [196 190 179]
  [189 183 172]
  [190 184 173]]

 [[254 254 254]
  [254 254 254]
  [254 254 254]
  ...
  [190 184 173]
  [173 167 156]
  [168 162 151]]

 [[254 254 254]
  [254 254 254]
  [254 254 254]
  ...
  [174 169 160]
  [156 151 142]
  [151 146 137]]

 [[233 230 226]
  [232 229 225]
  [230 227 223]
  ...
  [241 240 236]
  [242 241 237]
  [242 241 237]]

 [[227 224 220]
  [228 225 221]
  [231 228 224]
  ...
  [240 239 235]
  [241 240 236]
  [241 240 236]]

 [[245 242 238]
  [247 244 240]
  [249 246 242]
  ...
  [240 239 235]
  [240 239 235]
  [240 239 235]]]
>>>
```

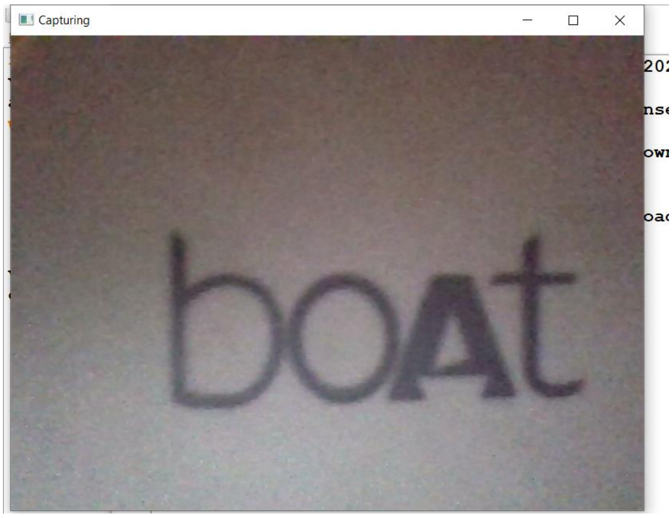
2. Resizing an image

```
import cv2
img = cv2.imread("E:\PHOTOS\elon.png", 1)
cv2.imshow("Alien", img)
res_img = cv2.resize(img, (200,300))
cv2.imshow("Resized", res_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



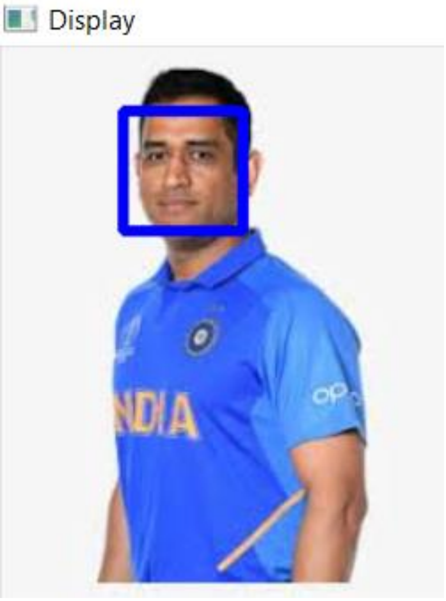
3. Capturing a video

```
import cv2,time
video = cv2.VideoCapture(0) #for webcam we can use 1,2
also for external webcams
a=1
wqhile True:
    check, frame = video.read()
    a = a + 1
    cv2.imshow("Capturing", frame)
    key = cv2.waitKey(1)
    if(key == ord('q')):
        break
video.release()
cv2.destroyAllWindows()
```



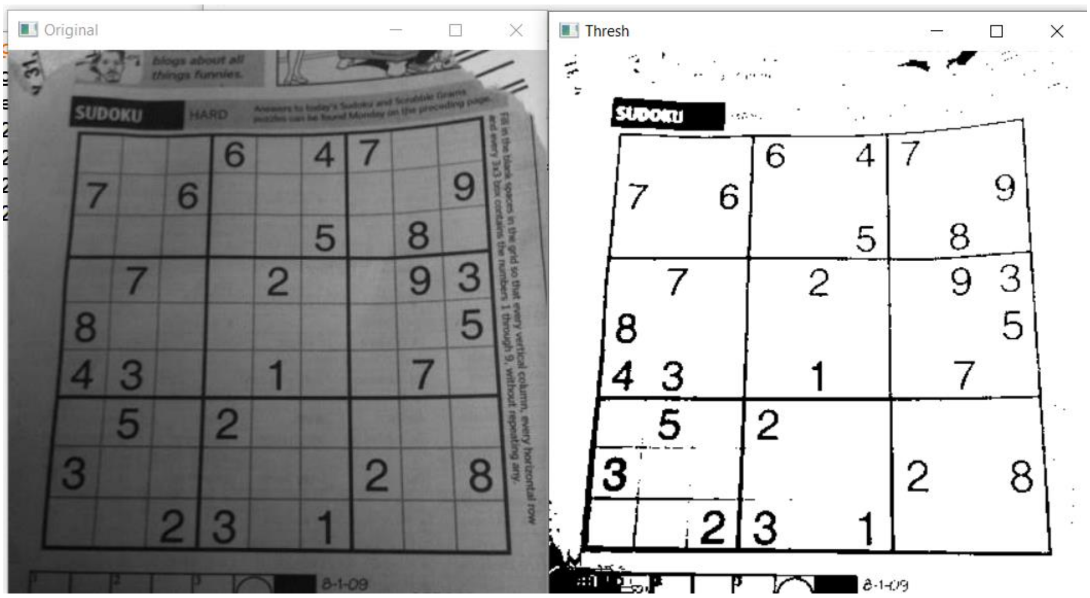
4. Face Detection

```
import cv2
face_cascade = cv2.CascadeClassifier("D:\haarcascade_frontalface_default.xml")
img = cv2.imread("E:\PHOTOS\mahi.jpg") #reading the colored image
faces = face_cascade.detectMultiScale(img, 1.5, 3) #searching for the face coordinates
for x,y,w,h in faces:
    img = cv2.rectangle(img, (x,y), (x+w, y+h), (255,0,0),3) #drawing rectangle
cv2.imshow("Display", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



5. ThreshHolding

```
import cv2
img = cv2.imread("E:\PHOTOS\sudoku.jpg",0)
check,thresh = cv2.threshold(img, 50, 255, cv2.THRESH_BINARY)
cv2.imshow("Original", img)
cv2.imshow("Thresh", thresh)
cv2.waitKey(0)
cv2.destroyAllWindows()
```


Original
Thresh
SUDOKU
HARD
8-1-09