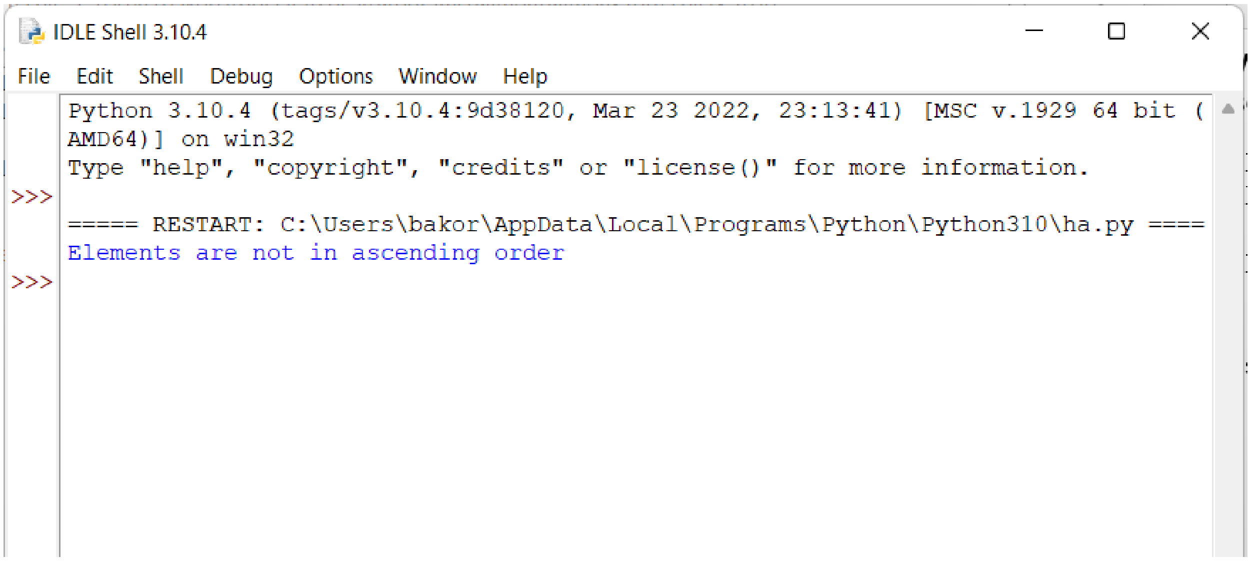
1. Python program to check whether the elements in a list are sorted in ascending order or not.

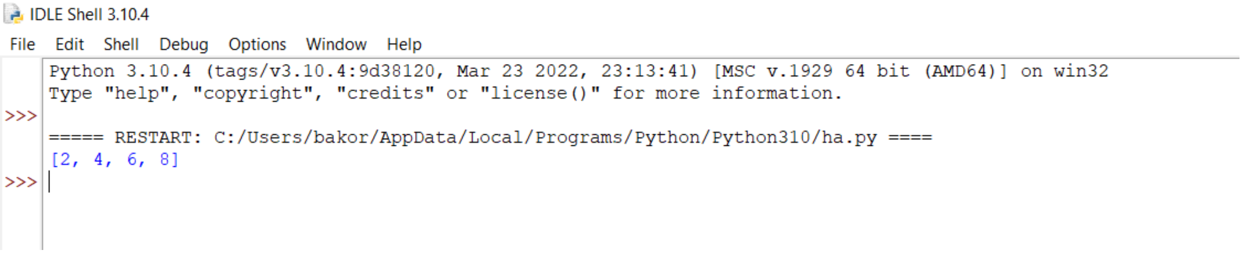
```
mylist1 = [1,4,2,3,6,7]
mylist2 = list(mylist1)

mylist2.sort()
if (mylist1 == mylist2):
    print("Elements are in ascending order")
else:
    print("Elements are not in ascending order")
```



```
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\bakor\AppData\Local\Programs\Python\Python310\ha.py ====
Elements are not in ascending order
>>>
```

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

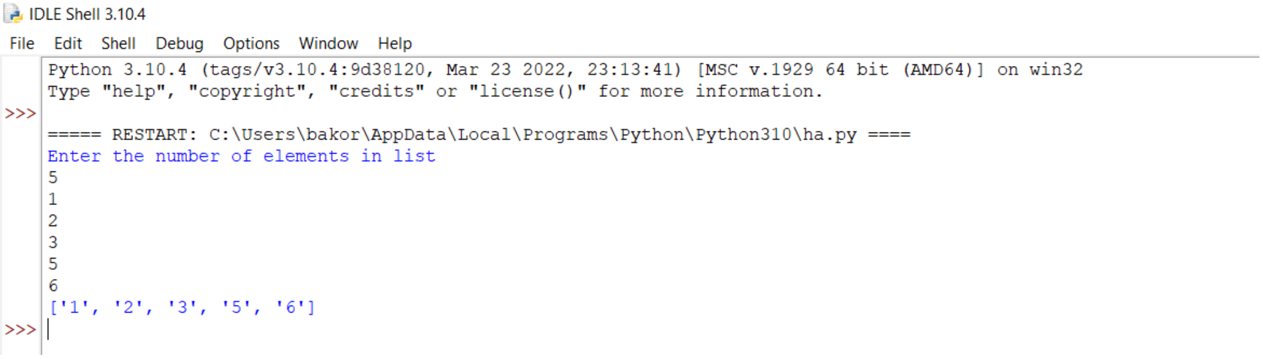
```
mylist = [1,2,3,4,5,6,7,8,9]

evenlist = []
for x in mylist:
    if (x%2 == 0):
        evenlist.append(x)

print(evenlist)
```



```
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/bakor/AppData/Local/Programs/Python/Python310/ha.py ====
[2, 4, 6, 8]
>>>
```

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

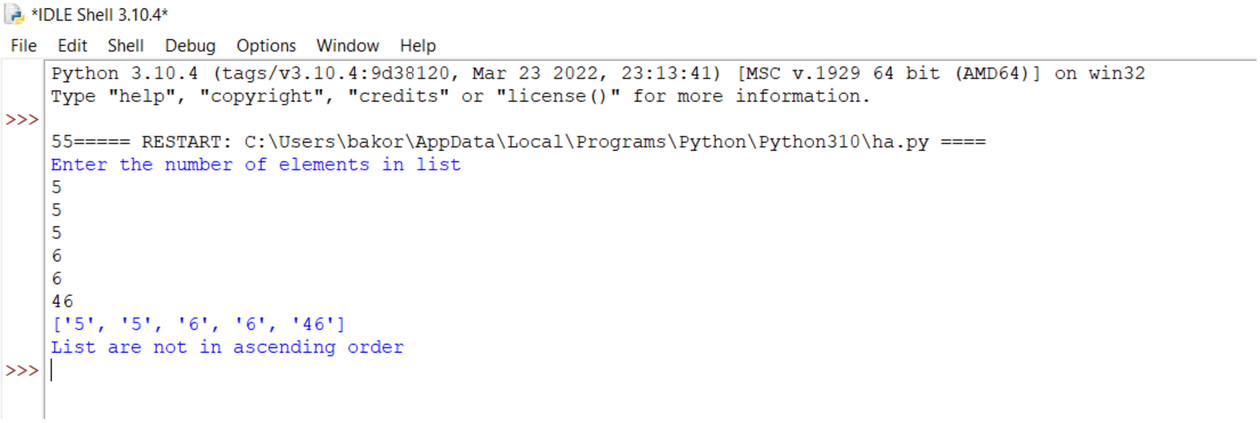
```
mylist = []

n = input("Enter the number of elements in list\n")
n = int(n)

for i in range(n):
    x = input()
    mylist.append(x)

print(mylist)
```



```
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\bakor\AppData\Local\Programs\Python\Python310\ha.py ====
Enter the number of elements in list
5
1
2
3
5
6
['1', '2', '3', '5', '6']
>>>
```

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

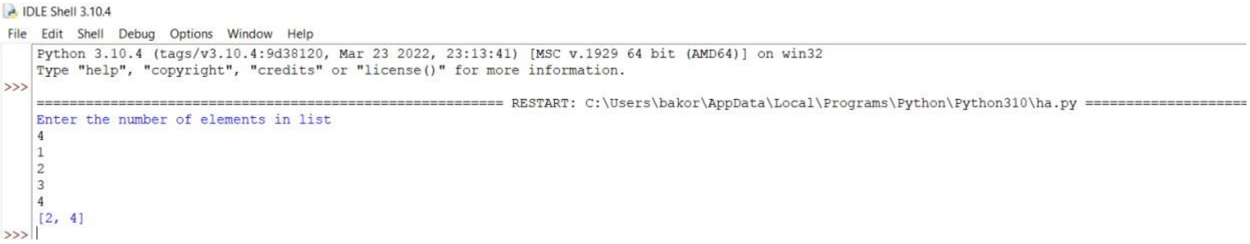
```
mylist = []
n = input("Enter the number of elements in list\n")
n = int(n)

for i in range(n):
    x = input()
    mylist.append(x)

print(mylist)

mylist1 = list(mylist)
mylist1.sort(key = str.lower)

if (mylist == mylist1):
    print("List is in ascending order\n")
else:
    print("List are not in ascending order")
```



```
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.
>>>
55===== RESTART: C:\Users\bakor\AppData\Local\Programs\Python\Python310\ha.py ====
Enter the number of elements in list
5
5
5
6
6
46
['5', '5', '6', '6', '46']
List are not in ascending order
>>>
```

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

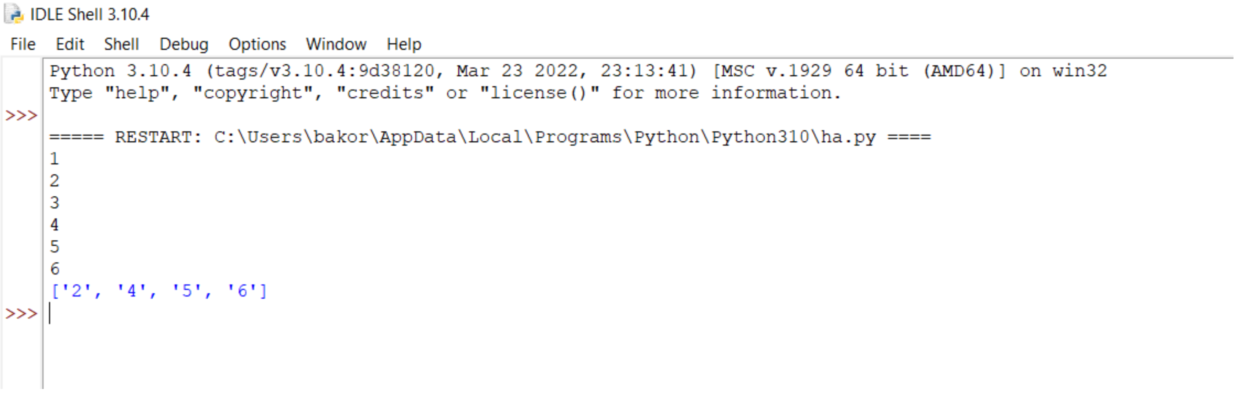
```
mylist = []

n = input("Enter the number of elements in list\n")
n = int(n)

for i in range(n):
    x = input()
    x = int(x)
    mylist.append(x)

evenlist = []
for i in mylist:
    if (i%2 == 0):
        evenlist.append(i)
    print(evenlist)
```



```
    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\bakor\AppData\Local\Programs\Python\Python310\ha.py ====================
    Enter the number of elements in list
    4
    1
    2
    3
    4
    [2, 4]
>>> |
```

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

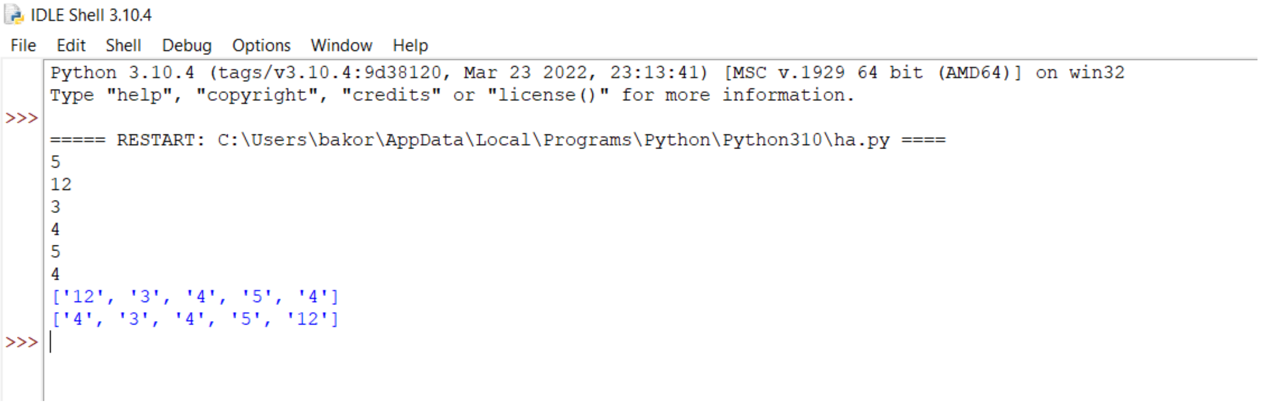
```
    mylist1 = []
    mylist2 = []

    n1 = input()
    n1 = int(n1)
    for i in range(n1):
        x = input()
        mylist1.append(x)

    n2 = input()
    n2 = int(n2)
    for i in range(n2):
        x = input()
        mylist2.append(x)

    mylist1.extend(mylist2)
    print(mylist1)
```



```
    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\bakor\AppData\Local\Programs\Python\Python310\ha.py ====
    1
    2
    3
    4
    5
    6
    ['2', '4', '5', '6']
>>> |
```

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

```
mylist = []

n = input()
n = int(n)
for i in range(n):
    x = input()
    mylist.append(x)

print(mylist)

if(len(mylist) >= 2):
    temp = mylist[0]
    mylist[0] = mylist[len(mylist) - 1]
    mylist[len(mylist) - 1] = temp

print(mylist)
```



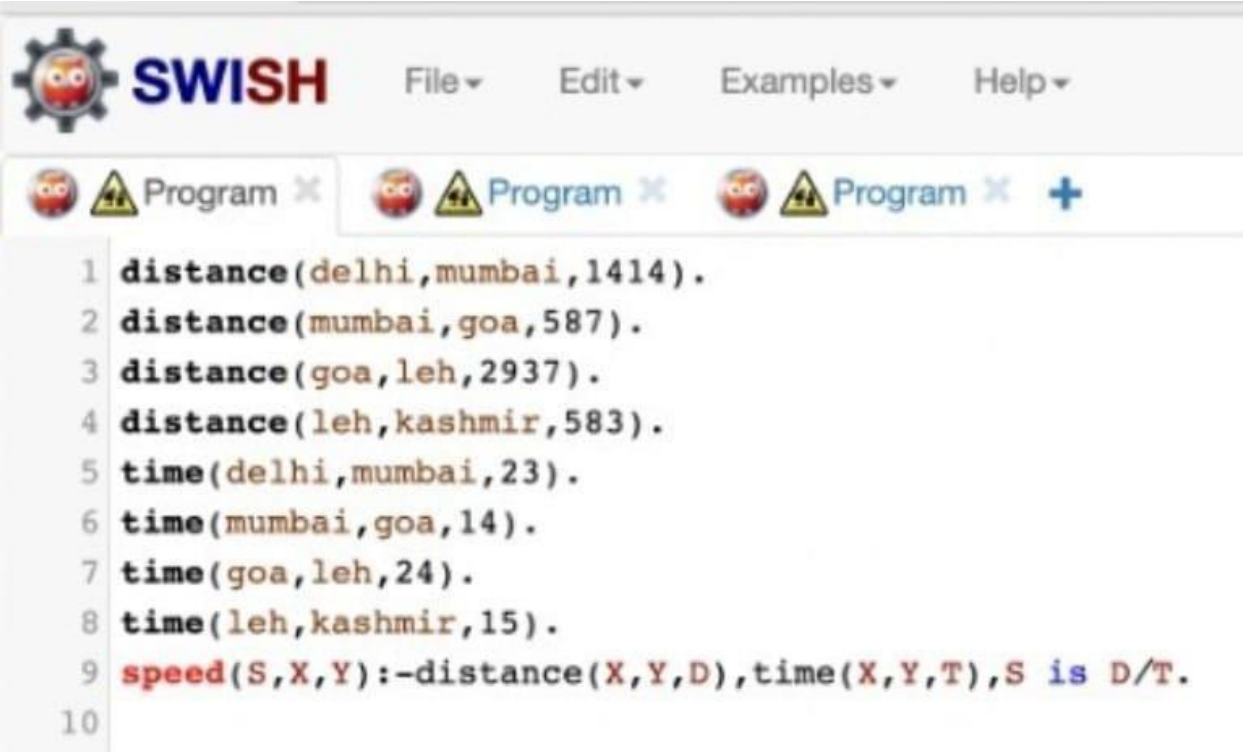
```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\bakor\AppData\Local\Programs\Python\Python310\ha.py ====
5
12
3
4
5
4
['12', '3', '4', '5', '4']
['4', '3', '4', '5', '12']
>>>
```

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

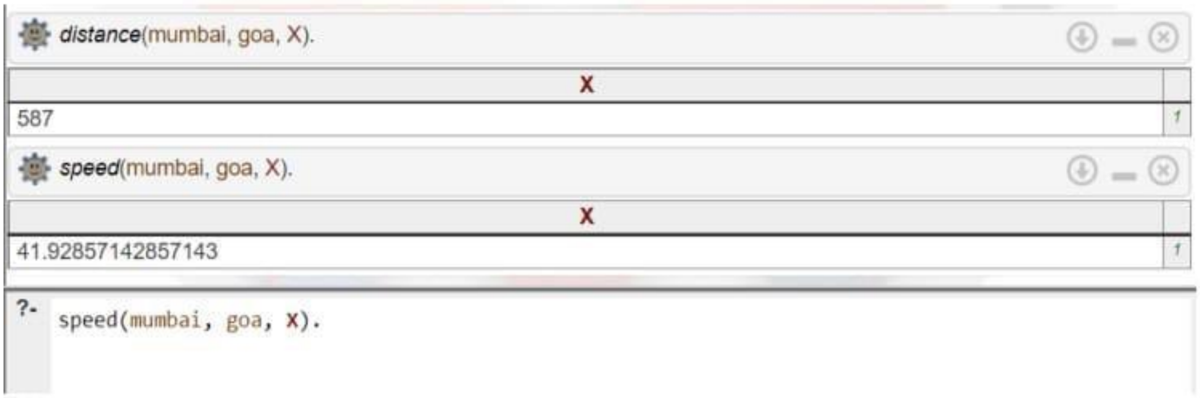
Source	Destination	Distance (KMs)	Time (Hrs)
Delhi	Mumbai	1414	23
Mumbai	Goa	587	14

Goa	Leh	2937	24
Leh	Kashmir	583	15

(i) Write the clauses to create Knowledge base in ProLog.
(ii) Write query to determine the distance between Mumbai and Goa.
(iii) Establish a rule to determine the speed of the biker between given cites.
Write the query for the same.



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

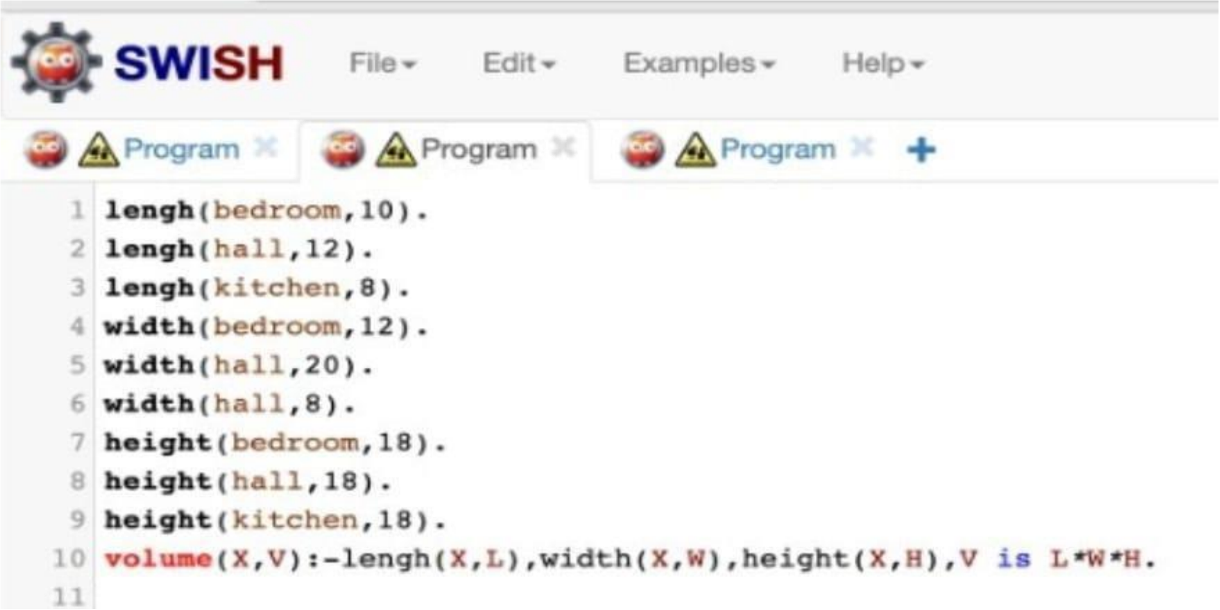


```
distance(mumbai, goa, X).
X
587
speed(mumbai, goa, X).
X
41.92857142857143
?- speed(mumbai, goa, X).
```

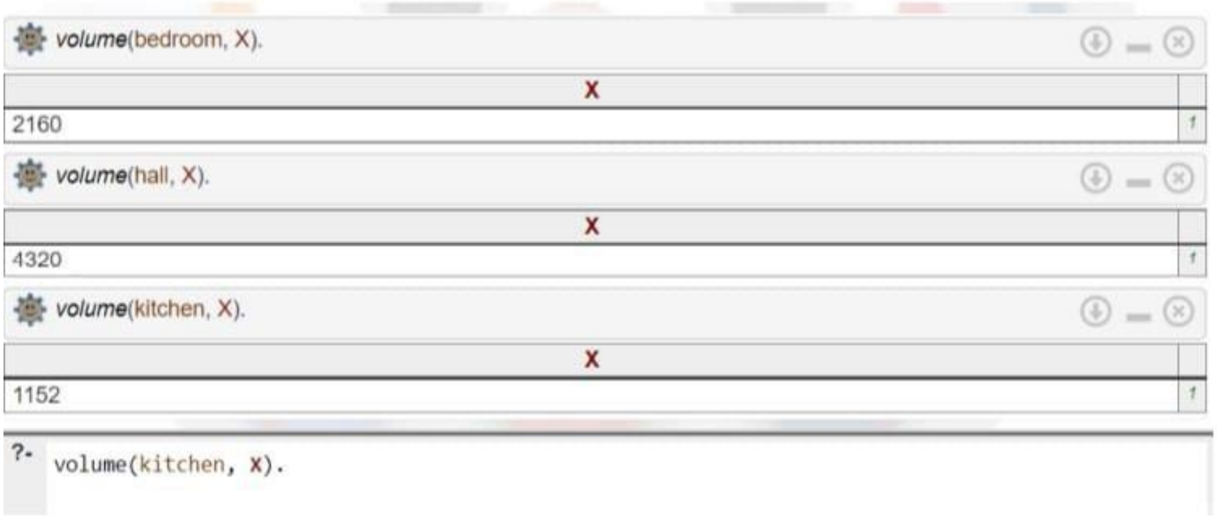
9. The linear measure, height and width of different living spaces are given below:-

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

(i) Write the ProLog program to create knowledge base for the following.
(ii) Also determine the rule to determine the volume of each individual living space.
(iii) Write queries to find out the volume of individual living spaces.



```
lengh(bedroom,10).
lengh(hall,12).
lengh(kitchen,8).
width(bedroom,12).
width(hall,20).
width(hall,8).
height(bedroom,18).
height(hall,18).
height(kitchen,18).
volume(X,V):-lengh(X,L),width(X,W),height(X,H),V is L*W*H.
```



```
volume(bedroom, X).
```

X
2160

```
volume(hall, X).
```

X
4320

```
volume(kitchen, X).
```

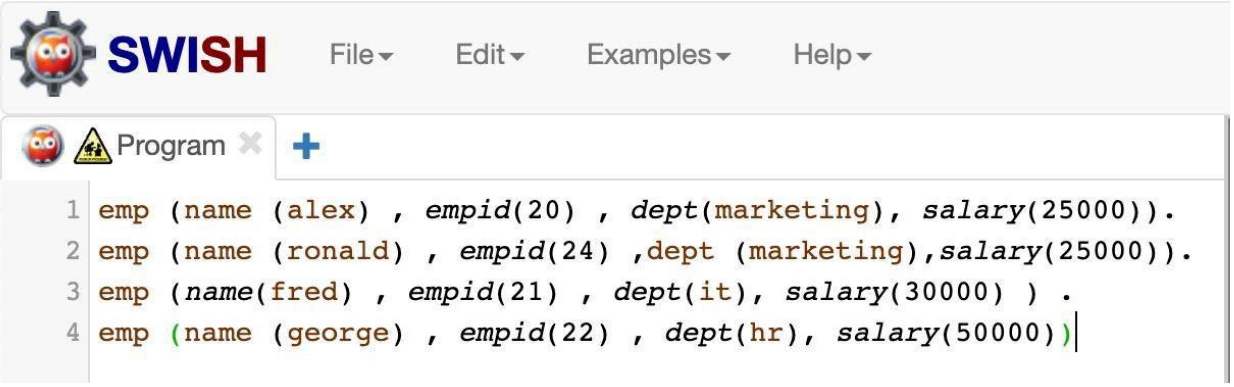
X
1152

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

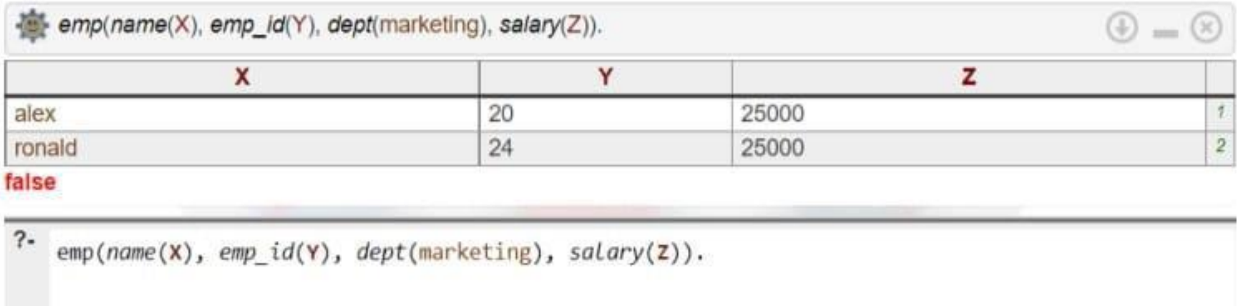
10. Following given are the details of Employees of an Organisation :-

Name	EmpID	Department	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 in Prolog.
(ii) Write query to display employees of Marketing department.



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



emp(name(X), emp_id(Y), dept(marketing), salary(Z)).

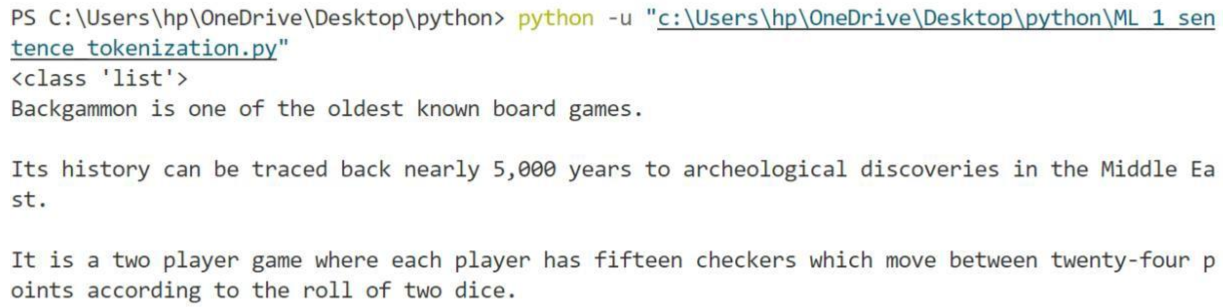
X	Y	Z	
alex	20	25000	1
ronald	24	25000	2

false

```
?- emp(name(X), emp_id(Y), dept(marketing), salary(Z)).
```

11. Sentence Tokenization

```
import nltk

text = "Backgammon is one of the oldest known board games. Its history can be
traced back nearly 5,000 years to archeological discoveries in the Middle East.
It is a two player game where each player has fifteen checkers which move between
twenty-four points according to the roll of two dice."

sentences = nltk.sent_tokenize(text)
print(type(sentences))

for sentence in sentences:
    print(sentence)
    print()
```

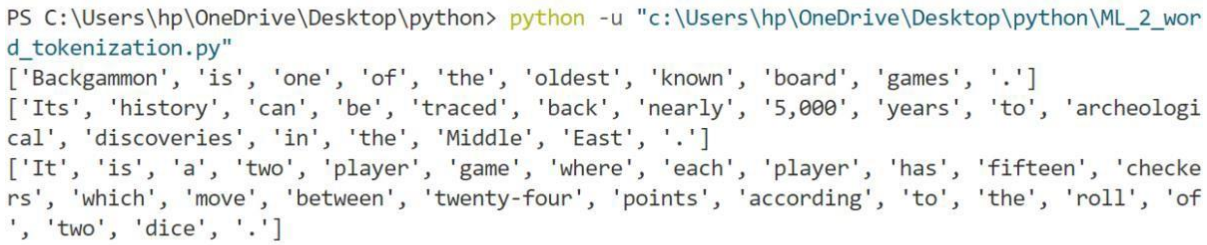


```
PS C:\Users\hp\OneDrive\Desktop\python> python -u "c:\Users\hp\OneDrive\Desktop\python\ML_1_sen
tence_tokenization.py"
<class 'list'>
Backgammon is one of the oldest known board games.

Its history can be traced back nearly 5,000 years to archeological discoveries in the Middle Ea
st.

It is a two player game where each player has fifteen checkers which move between twenty-four p
oints according to the roll of two dice.
```

12. Word Tokenization

```
import nltk

text = "Backgammon is one of the oldest known board games. Its history can be
traced back nearly 5,000 years to archeological discoveries in the Middle East.
It is a two player game where each player has fifteen checkers which move between
twenty-four points according to the roll of two dice."
```

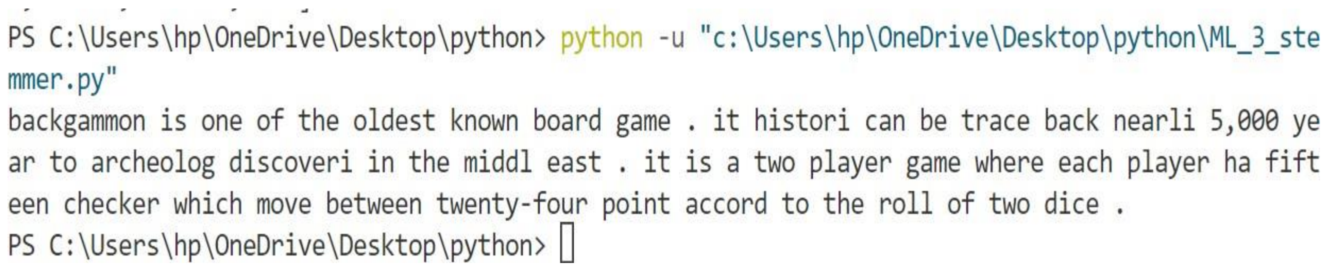
```
sentences = nltk.sent_tokenize(text)
for sentence in sentences:
        words = nltk.word_tokenize(sentence)
        print(words)
print()

for i in range(len(sentences)):
    words = nltk.word_tokenize(sentences[i])
    print(words)
```



```
PS C:\Users\hp\OneDrive\Desktop\python> python -u "c:\Users\hp\OneDrive\Desktop\python\ML_2_wor
d_tokenization.py"
['Backgammon', 'is', 'one', 'of', 'the', 'oldest', 'known', 'board', 'games', '.']
['Its', 'history', 'can', 'be', 'traced', 'back', 'nearly', '5,000', 'years', 'to', 'archeologi
cal', 'discoveries', 'in', 'the', 'Middle', 'East', '.']
['It', 'is', 'a', 'two', 'player', 'game', 'where', 'each', 'player', 'has', 'fifteen', 'checke
rs', 'which', 'move', 'between', 'twenty-four', 'points', 'according', 'to', 'the', 'roll', 'of
', 'two', 'dice', '.']
```

13. Stemming

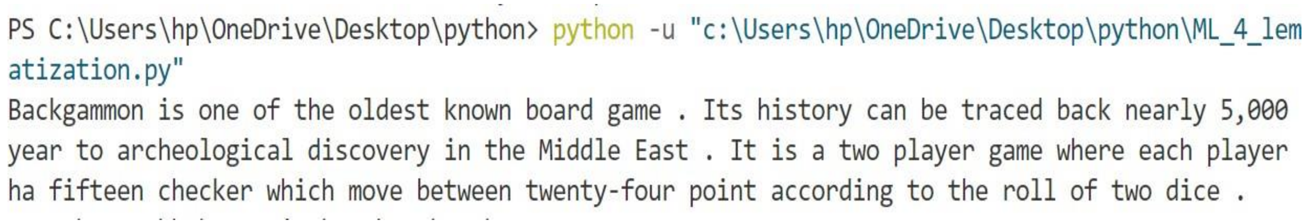
```
import nltk
from nltk.stem import PorterStemmer
stemmer = PorterStemmer()

text = "Backgammon is one of the oldest known board games. Its history can be
traced back nearly 5,000 years to archeological discoveries in the Middle East.
It is a two player game where each player has fifteen checkers which move between
twenty-four points according to the roll of two dice."

sentences = nltk.sent_tokenize(text)

for sentence in sentences:
    words = nltk.word_tokenize(sentence)
    for word in words:
        print(stemmer.stem(word),end=" ")
```



```
PS C:\Users\hp\OneDrive\Desktop\python> python -u "c:\Users\hp\OneDrive\Desktop\python\ML_3_ste
mmer.py"
backgammon is one of the oldest known board game . it histori can be trace back nearli 5,000 ye
ar to archeolog discoveri in the middl east . it is a two player game where each player ha fift
een checker which move between twenty-four point accord to the roll of two dice .
PS C:\Users\hp\OneDrive\Desktop\python> 
```

14. Lemmatization

```
import nltk
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()

text = "Backgammon is one of the oldest known board games. Its history can be
traced back nearly 5,000 years to archeological discoveries in the Middle East.
It is a two player game where each player has fifteen checkers which move between
twenty-four points according to the roll of two dice."

sentences = nltk.sent_tokenize(text)
for sentence in sentences:
    words = nltk.word_tokenize(sentence)
    for word in words:
        print(lemmatizer.lemmatize(word), end=" ")
```

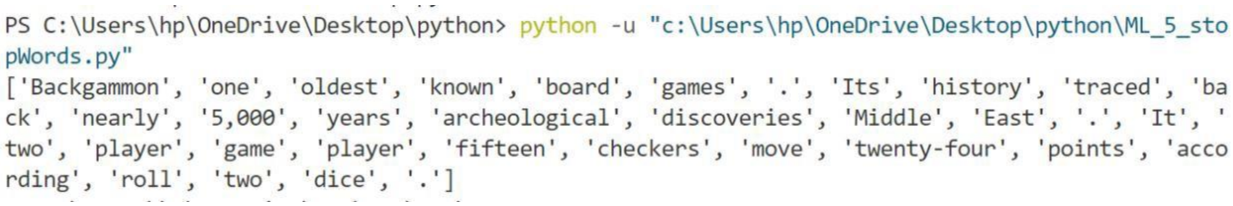


```
PS C:\Users\hp\OneDrive\Desktop\python> python -u "c:\Users\hp\OneDrive\Desktop\python\ML_4_lem
atization.py"
Backgammon is one of the oldest known board game . Its history can be traced back nearly 5,000
year to archeological discovery in the Middle East . It is a two player game where each player
ha fifteen checker which move between twenty-four point according to the roll of two dice .
```

15. Stop words removal

```
import nltk
from nltk.corpus  import  stopwords

# print(stopwords.words("english"))

text = "Backgammon is one of the oldest known board games. Its history can be
traced back nearly 5,000 years to archeological discoveries in the Middle East.
```

```
It is a two player game where each player has fifteen checkers which move between
twenty-four points according to the roll of two dice."

stop_words = set(stopwords.words("english"))
words = nltk.word_tokenize(text)

without_stop_words = []

for word in words:
    if word not in stop_words:
        without_stop_words.append(word)

print(without_stop_words)

"""
```

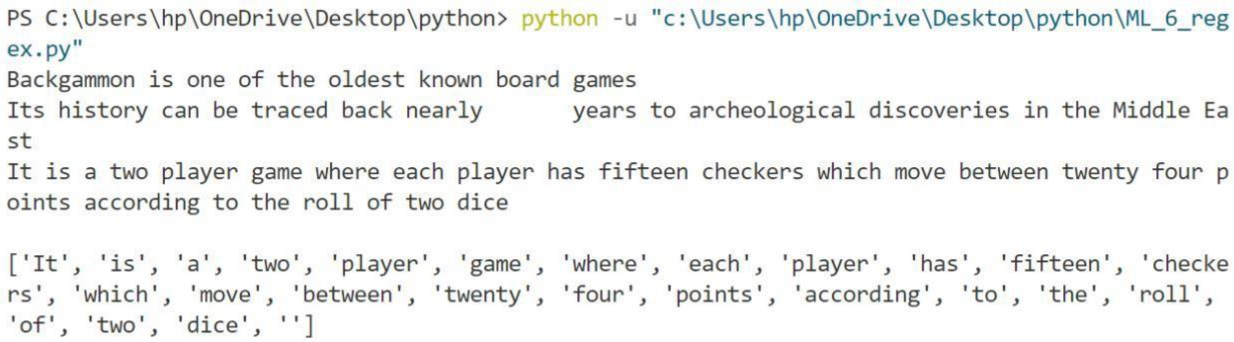


```
PS C:\Users\hp\OneDrive\Desktop\python> python -u "c:\Users\hp\OneDrive\Desktop\python\ML_5_sto
pWords.py"
['Backgammon', 'one', 'oldest', 'known', 'board', 'games', '.', 'Its', 'history', 'traced', 'ba
ck', 'nearly', '5,000', 'years', 'archeological', 'discoveries', 'Middle', 'East', '.', 'It', '
two', 'player', 'game', 'player', 'fifteen', 'checkers', 'move', 'twenty-four', 'points', 'acco
rding', 'roll', 'two', 'dice', '.']
```

16. Regex expression

```
import re
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords

text = "Backgammon is one of the oldest known board games. Its history can be
traced back nearly 5,000 years to archeological discoveries in the Middle East.
It is a two player game where each player has fifteen checkers which move between
twenty-four points according to the roll of two dice."

lemmatizer = WordNetLemmatizer()
sentences = nltk.sent_tokenize(text)

# sub - substitute
for sentence in sentences:
    review = re.sub("[^a-zA-Z]", " ", sentence)
    print(review)
print()
# print(review)
# print(review.lower())
```

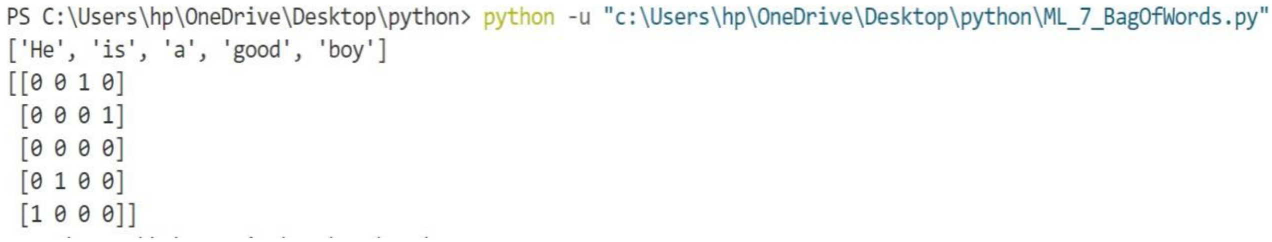
```
review = re.split("[^a-zA-Z]", review)
print(review)
```



```
PS C:\Users\hp\OneDrive\Desktop\python> python -u "c:\Users\hp\OneDrive\Desktop\python\ML_6_reg
ex.py"
Backgammon is one of the oldest known board games
Its history can be traced back nearly       years to archeological discoveries in the Middle Ea
st
It is a two player game where each player has fifteen checkers which move between twenty four p
oints according to the roll of two dice

['It', 'is', 'a', 'two', 'player', 'game', 'where', 'each', 'player', 'has', 'fifteen', 'checke
rs', 'which', 'move', 'between', 'twenty', 'four', 'points', 'according', 'to', 'the', 'roll',
'of', 'two', 'dice', '']
```

17. Bag of Words

```
text = "He is a good boy"

import nltk
from sklearn.feature_extraction.text import CountVectorizer
from nltk.corpus import stopwords
count_vectorizer = CountVectorizer()

stop_words = set(stopwords.words("english"))
words = nltk.word_tokenize(text)

print(words)
bag_of_words = count_vectorizer.fit_transform(words).toarray()
print(bag_of_words)
```

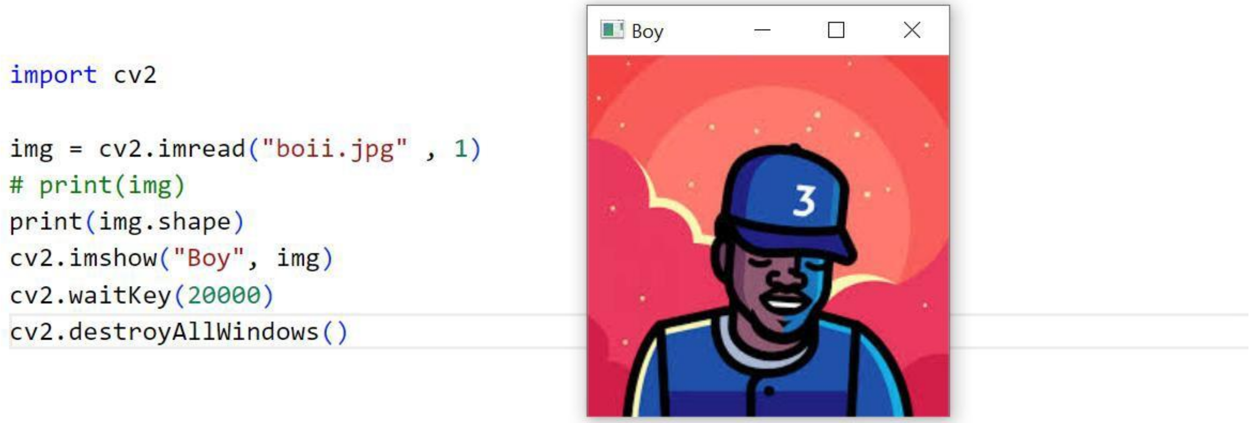


```
PS C:\Users\hp\OneDrive\Desktop\python> python -u "c:\Users\hp\OneDrive\Desktop\python\ML_7_BagOfWords.py"
['He', 'is', 'a', 'good', 'boy']
[[0 0 1 0]
 [0 0 0 1]
 [0 0 0 0]
 [0 1 0 0]
 [1 0 0 0]]
```

18. reading image

```
import cv2
```

```
img = cv2.imread("boii.jpg" , 1)
# print(img)
print(img.shape)
cv2.imshow("Boy", img)
cv2.waitKey(20000)
cv2.destroyAllWinows()
```



```
import cv2

img = cv2.imread("boii.jpg" , 1)
# print(img)
print(img.shape)
cv2.imshow("Boy", img)
cv2.waitKey(20000)
cv2.destroyAllWindows()
```

19. video Capture

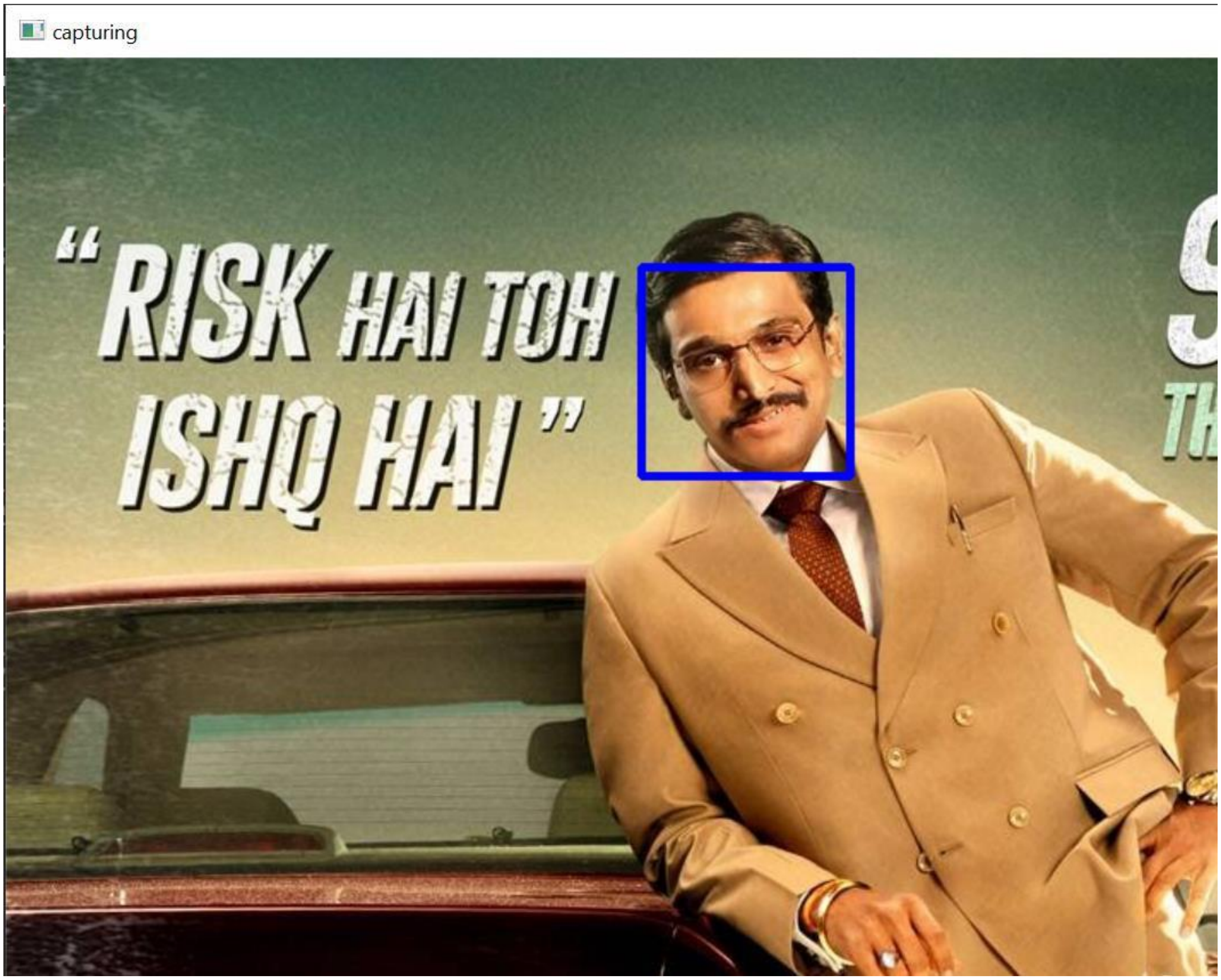
```
import cv2,time

video = cv2.VideoCapture(0,cv2.CAP_DSHOW
time.sleep(5)
check,frame = video.read()

video.release()
cv2.imshow("captured",frame)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

20. full video reading

```
import cv2

video = cv2.VideoCapture(0)
while True:
    check,frame = video.read()

    cv2.imshow("video",frame)
    key = cv2.waitKey(1)
    if(key == ord('q')):
        break

video.release()
cv2.destroyAllWindows()
```

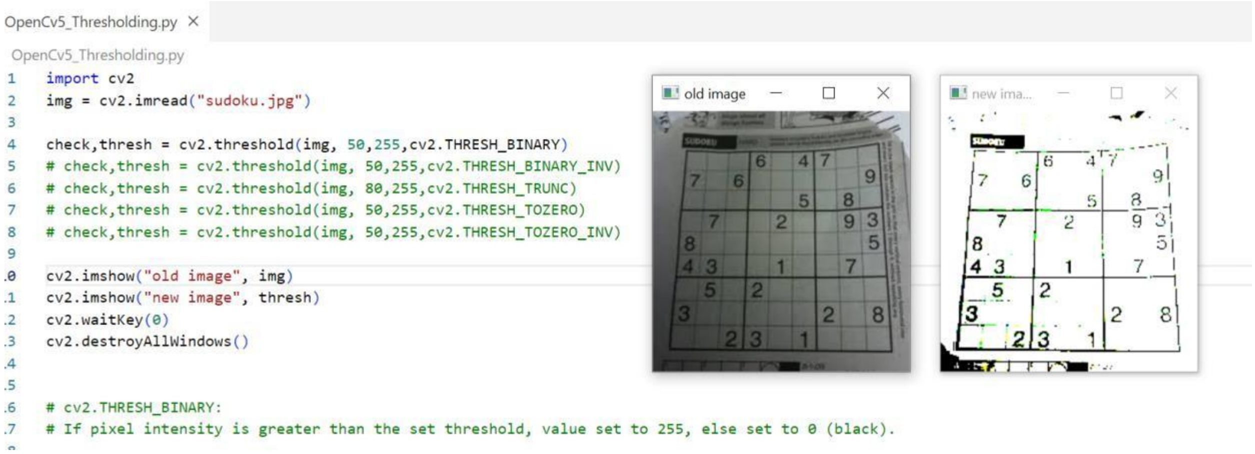
21. face detection

```
import cv2
face_cascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
img = cv2.imread("harshad.jpg",1)
face = face_cascade.detectMultiScale(img,1.5,3)

for x,y,w,h in face:
    img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),3)

cv2.imshow("capturing",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



22. Thresholding

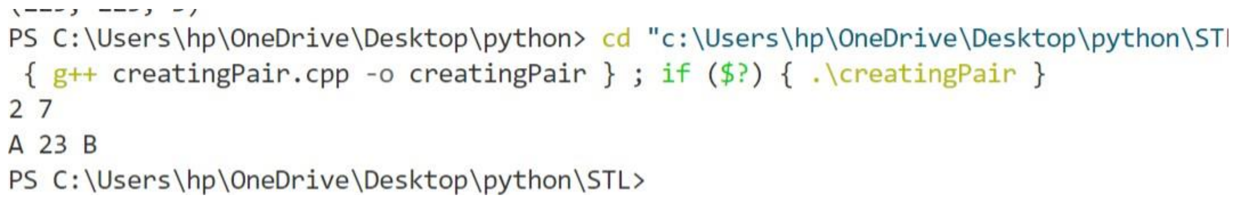
```
import cv2
img = cv2.imread("sudoku.jpg")

check,thresh = cv2.threshold(img, 50,255,cv2.THRESH_BINARY)

cv2.imshow("old image", img)
cv2.imshow("new image", thresh)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



```
import cv2
img = cv2.imread("sudoku.jpg")

check,thresh = cv2.threshold(img, 50,255,cv2.THRESH_BINARY)
# check,thresh = cv2.threshold(img, 50,255,cv2.THRESH_BINARY_INV)
# check,thresh = cv2.threshold(img, 80,255,cv2.THRESH_TRUNC)
# check,thresh = cv2.threshold(img, 50,255,cv2.THRESH_TOZERO)
# check,thresh = cv2.threshold(img, 50,255,cv2.THRESH_TOZERO_INV)

cv2.imshow("old image", img)
cv2.imshow("new image", thresh)
cv2.waitKey(0)
cv2.destroyAllWindows()


# cv2.THRESH_BINARY:
# If pixel intensity is greater than the set threshold, value set to 255, else set to 0 (black).
```

23. creating Pair

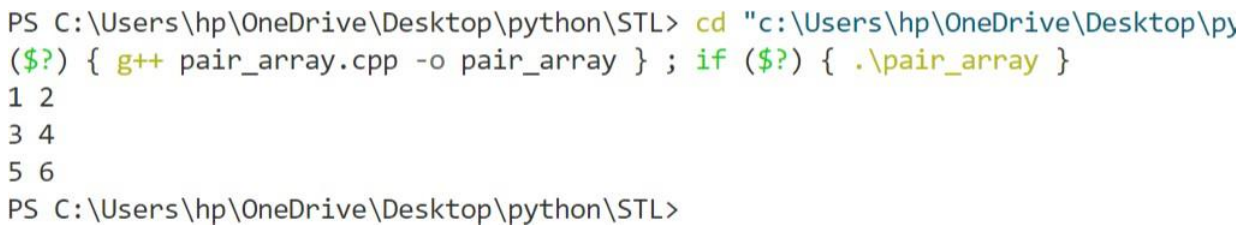
```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    pair<int, int> p;
    p.first = 2;
    p.second = 7;
    cout << p.first << " " << p.second << endl;

    pair<char, pair<int, char>> p1('A', {23, 'B'});
    cout << p1.first << " " << p1.second.first << " " << p1.second.second <<
endl;
    return 0;
}
```



```
PS C:\Users\hp\OneDrive\Desktop\python> cd "c:\Users\hp\OneDrive\Desktop\python\ST
 { g++ creatingPair.cpp -o creatingPair } ; if ($?) { .\creatingPair }
2 7
A 23 B
PS C:\Users\hp\OneDrive\Desktop\python\STL>
```

24. Pair Array

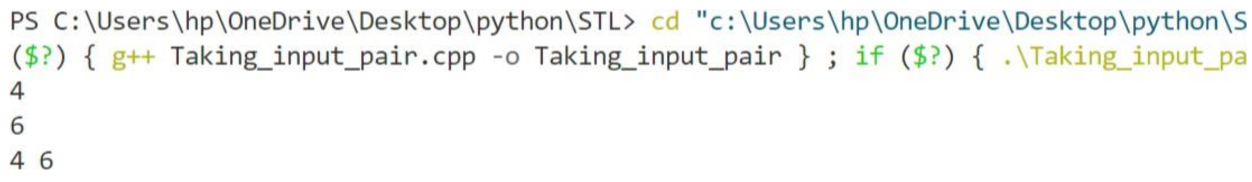
```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    pair<int, int> p[3];
    p[0] = {1, 2};
    p[1] = {3, 4};
    p[2] = {5, 6};

    for (auto i : p)
        cout << i.first << " " << i.second << endl;
    return 0;
}
```



```
PS C:\Users\hp\OneDrive\Desktop\python\STL> cd "c:\Users\hp\OneDrive\Desktop\py
($?) { g++ pair_array.cpp -o pair_array } ; if ($?) { .\pair_array }
1 2
3 4
5 6
PS C:\Users\hp\OneDrive\Desktop\python\STL>
```

25. Taking Input pair

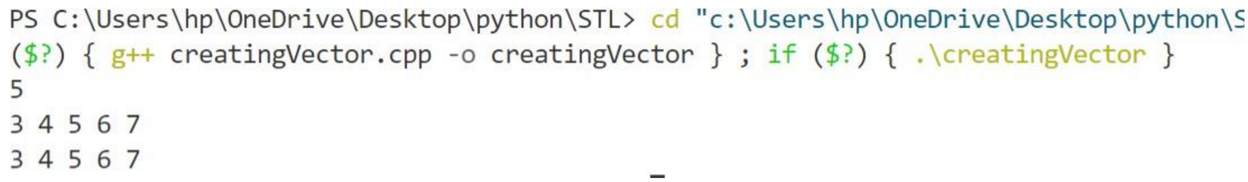
```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    pair<int, int> p;
    cin >> p.first;
    cin >> p.second;

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



```
PS C:\Users\hp\OneDrive\Desktop\python\STL> cd "c:\Users\hp\OneDrive\Desktop\python\S
($?) { g++ Taking_input_pair.cpp -o Taking_input_pair } ; if ($?) { .\Taking_input_pa
4
6
4 6
```

26. Creating vector

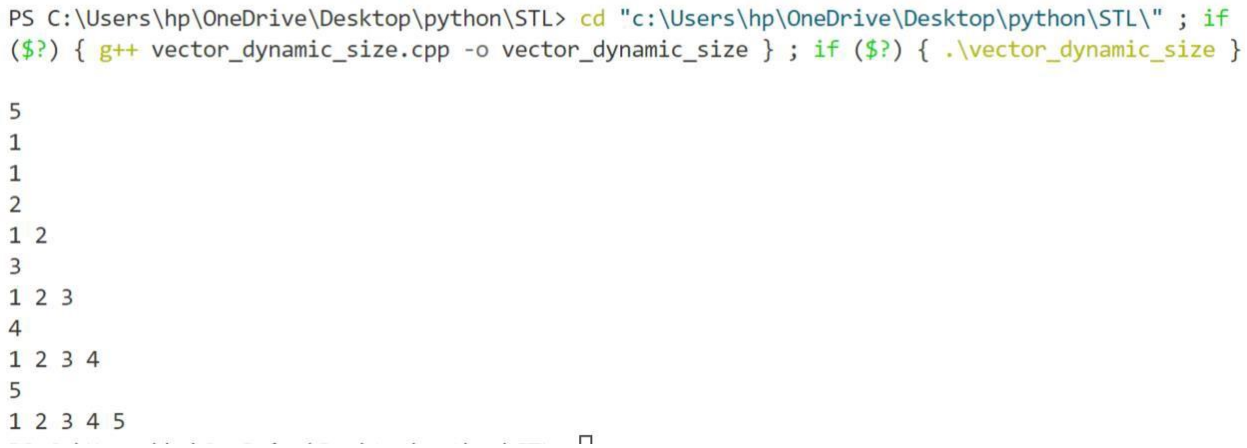
```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

void printVec(vector<int> v)
{
    for (auto i : v)
        cout << i << " ";
}

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    vector<int> arr;
    int n;
    cin >> n;
    int x;

    for (int i = 0; i < n; i++)
    {
        cin >> x;
        arr.push_back(x);
    }

    printVec(arr);

    return 0;
}
```



```
PS C:\Users\hp\OneDrive\Desktop\python\STL> cd "c:\Users\hp\OneDrive\Desktop\python\S
($?) { g++ creatingVector.cpp -o creatingVector } ; if ($?) { .\creatingVector }
5
3 4 5 6 7
3 4 5 6 7
```

27. Dynamic vector

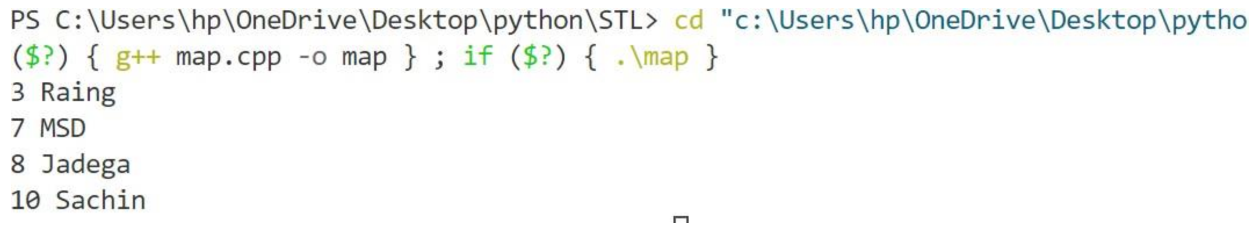
```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

void printVec(vector<int> v)
{
    for (auto i : v)
        cout << i << " ";
    cout << endl;
}

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    vector<int> v;
    // vector<int> v(3); vector of size 3
    // vector<int> v(5,8); vector of size 5 with all elements 8

    int n;
    cin >> n;
    int x;

    for (int i = 0; i < n; i++)
    {
        cin >> x;
        v.push_back(x);
        printVec(v);
    }

    return 0;
}
```



```
PS C:\Users\hp\OneDrive\Desktop\python\STL> cd "c:\Users\hp\OneDrive\Desktop\python\STL\" ; if
($?) { g++ vector_dynamic_size.cpp -o vector_dynamic_size } ; if ($?) { .\vector_dynamic_size }

5
1
1
2
1 2
3
1 2 3
4
1 2 3 4
5
1 2 3 4 5
```

28. Map

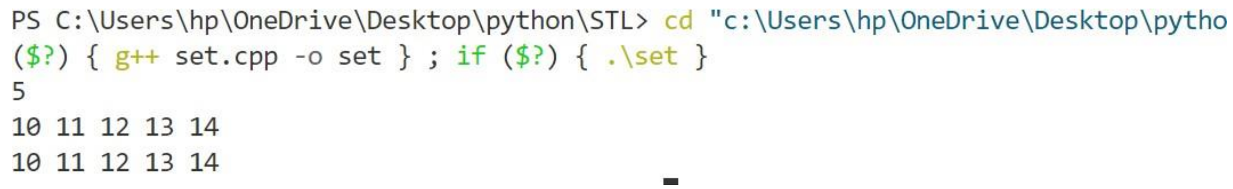
```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    map<int, string> m;
    m[7] = "MSD";
    m[8] = "Jadega";
    m[3] = "Raing";

    m.insert({10, "Sachin"});

    // map<int,string> :: iterator it = m.begin();
    for (auto it : m)
        cout << it.first << " " << it.second << endl;

    auto it = m.find(3);
    if (it != m.end())
        m.erase(it);
    else
        cout << "Key value pair not found" << endl;

    return 0;
}
```



```
PS C:\Users\hp\OneDrive\Desktop\python\STL> cd "c:\Users\hp\OneDrive\Desktop\pytho
($?) { g++ map.cpp -o map } ; if ($?) { .\map }
3 Raing
7 MSD
8 Jadega
10 Sachin
```

29. Set

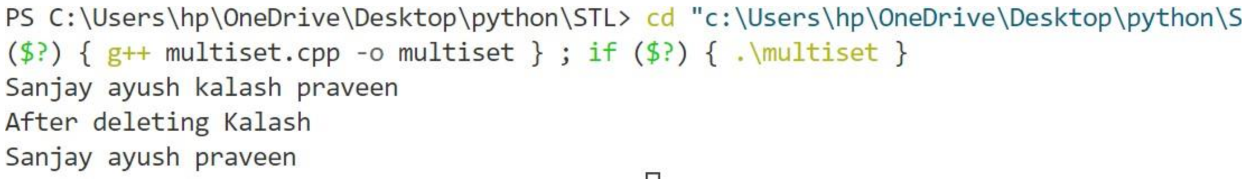
```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    int n;
    cin >> n;
    set<string> st;
    string s;

    for (int i = 0; i < n; i++)
    {
        cin >> s;
        st.insert(s);
    }

    for (auto it : st)
        cout << it << " ";

    return 0;
}
```



```
PS C:\Users\hp\OneDrive\Desktop\python\STL> cd "c:\Users\hp\OneDrive\Desktop\pytho
($?) { g++ set.cpp -o set } ; if ($?) { .\set }
5
10 11 12 13 14
10 11 12 13 14
```

30. Multiset

```cpp
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    multiset<string> m;
    m.insert("ayush");
    m.insert("kalash");
    m.insert("praveen");
    m.insert("Sanjay");

    for (string s : m)
        cout << s << " ";
    cout << endl;

    auto it = m.find("kalash");
    if (it != m.end())
        m.erase(it);
    else
        cout << "No such value exist";

    cout << "After deleting Kalash" << endl;
    for (string s : m)
        cout << s << " ";

    return 0;
}
```



```
PS C:\Users\hp\OneDrive\Desktop\python\STL> cd "c:\Users\hp\OneDrive\Desktop\python\S
($?) { g++ multiset.cpp -o multiset } ; if ($?) { .\multiset }
Sanjay ayush kalash praveen
After deleting Kalash
Sanjay ayush praveen
```