A Record On

ARTIFICIAL INTELLIGENCE-LAB

B.TECH(INFORMATIONTECHNOLOGY)

III YEAR II SEMESTER

2021-2022

Department Of Information Technology

Bapatla Engineering College::Bapatla

(Autonomous)

(Affiliated to Acharya Nagarjuna University)

BAPATLA – 522101, A.P

Bapatla Engineering College::Bapatla

(Autonomous)

Department of Information Technology

CERTIFICATE

Regd.No: **Y19AIT413**

Roll No :**37**

Certified and bonafide Record of work done by......**B BALJAI**....... B. Tech Third year…...**IT-A**.................... Section in the subject of **ARTIFICIAL INTELLIGENCE-LAB(Y18ITL62)** during the year 2021–2022.Number of experiments recorded …**12**……

Lecturer-in-charge	**Head of the Department**
Mr. B KRISHNAIAH, Asst.Professor	Prof. Dr N. SIVARAM PRASAD,
Date: ……………………………	Dept. of I.T.

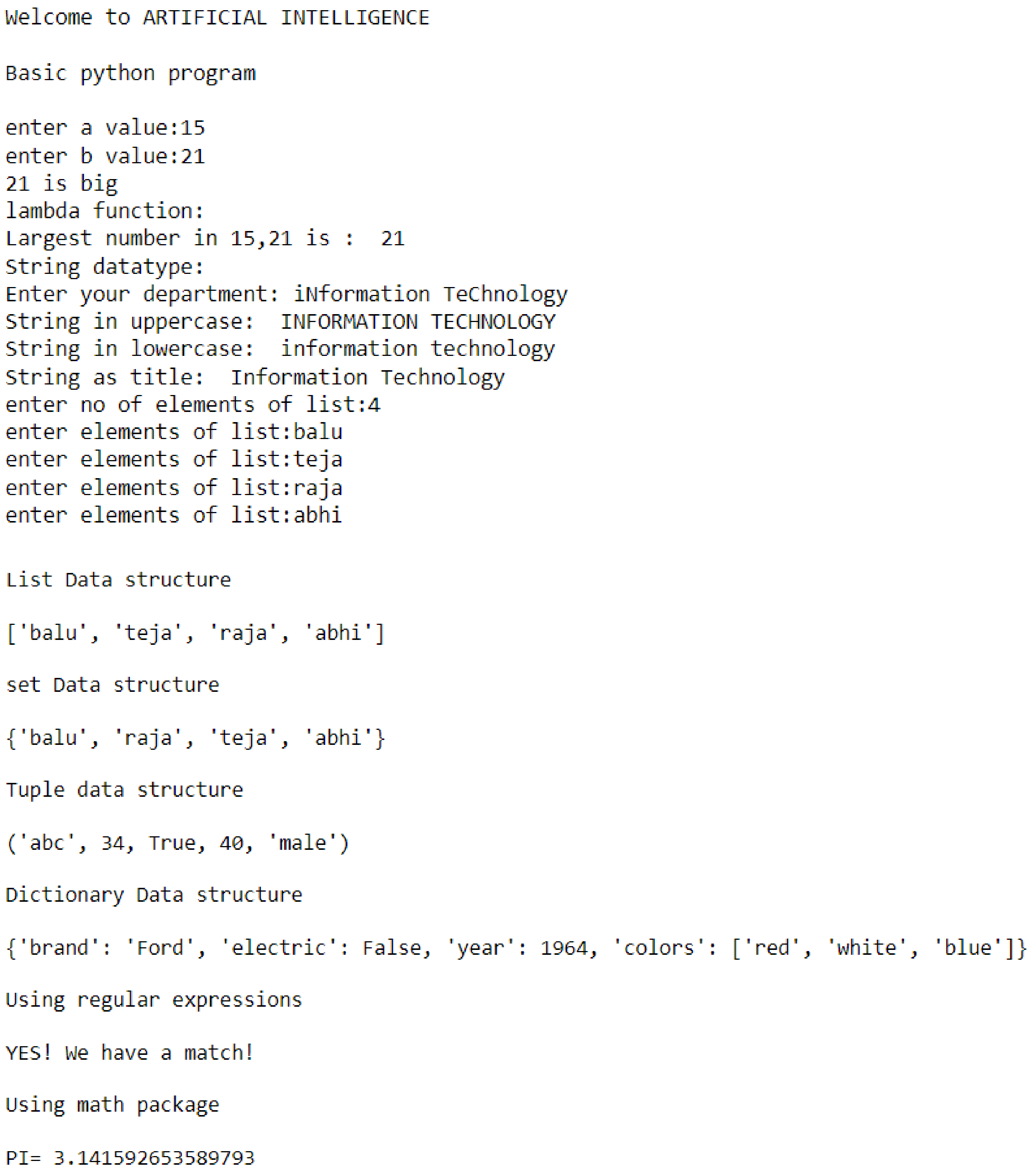
INDEX

Aim: 1. Write a program to demonstrate a simple python program.

Source code:

```
import math
import re
print("\nWelcome to ARTIFICIAL INTELLIGENCE\n")
print("Basic python program\n")
a=int(input("enter a value:"))
b=int(input("enter b value:"))
if(a>b):
    print("{} is big".format(a))
else:
    print("{} is big".format(b))
print("lambda function: ")
x = lambda a, b: a if a>b else b
print(f"Largest number in {a},{b} is : ",x(a, b))
print("String datatype: ")
s=input("Enter your department: ")
print("String in uppercase: ",s.upper())
print("String in lowercase: ",s.lower())
print("String as title: ",s.title())
l=[]
n=int(input("enter no of elements of list:"))
for i in range(0,n):
    k=input("enter elements of list:")
    l.append(k)
print("\nList Data structure\n")
print(l)
print("\nset Data structure\n")
print(set(l))
print("\nTuple data structure\n")
t=tuple(("abc", 34, True, 40, "male"))
print(t)
print("\nDictionary Data structure\n")
d={"brand": "Ford","electric": False, "year": 1964,"colors": ["red", "white", "blue"]}
print(d)
print("\nUsing regular expressions\n ")
txt = "The Bahubali movie is th biggest movie till now in India"
x = re.search("^The.*India$", txt)
if x:
    print("YES! We have a match!")
else:
    print("No match")
print("\nUsing math package\n")
print("PI=",math.pi)
```

Output:



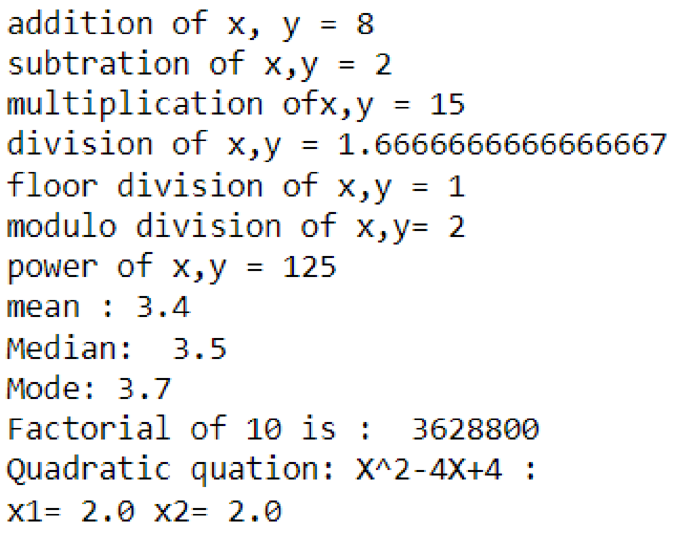
```
Welcome to ARTIFICIAL INTELLIGENCE

Basic python program

enter a value:15
enter b value:21
21 is big
lambda function:
Largest number in 15,21 is :  21
String datatype:
Enter your department: iNformation TeChnology
String in uppercase:  INFORMATION TECHNOLOGY
String in lowercase:  information technology
String as title:  Information Technology
enter no of elements of list:4
enter elements of list:balu
enter elements of list:teja
enter elements of list:raja
enter elements of list:abhi


List Data structure

['balu', 'teja', 'raja', 'abhi']

set Data structure

{'balu', 'raja', 'teja', 'abhi'}

Tuple data structure

('abc', 34, True, 40, 'male')

Dictionary Data structure

{'brand': 'Ford', 'electric': False, 'year': 1964, 'colors': ['red', 'white', 'blue']}

Using regular expressions

YES! We have a match!

Using math package

PI= 3.141592653589793
```

Aim: 2. Write a python program to implement Arithmetic Operations

Source code:

```
import math
x = 5 #int(input("enter x value:"))
y = 3 #int(input("enter y value:"))
print('addition of x, y =',x+y)
print('subtration of x,y =',x-y)
print('multiplication ofx,y =',x*y)
print('division of x,y =',x/y)
print('floor division of x,y =',x//y)
print("modulo division of x,y=",x%y)
print('power of x,y =',x**y)
l=[1,3,5,3,2,4,1,5,3,7]
sum=0
for i in l:
    sum+=i
mean=sum/len(l)
print("mean :",mean)
l.sort()
mid=int(len(l)/2)
if len(l)%2 ==0:
    med=(l[mid]+l[mid+1])/2
else:
    med=l[mid]
print("Median: ",med)
Mode = 3 * med - 2 * mean
print('Mode:',Mode)
fact=1
for i in range(1,11):
    fact*=i
print("Factorial of 10 is : ", fact)
print("Quadratic quation: X^2-4X+4 :")
dis=math.sqrt(4**2 - 4*1*4)
x1,x2=(4-dis)/2*1,(4+dis)/2*1
print("x1=",x1,"x2=",x2)
```

Output:



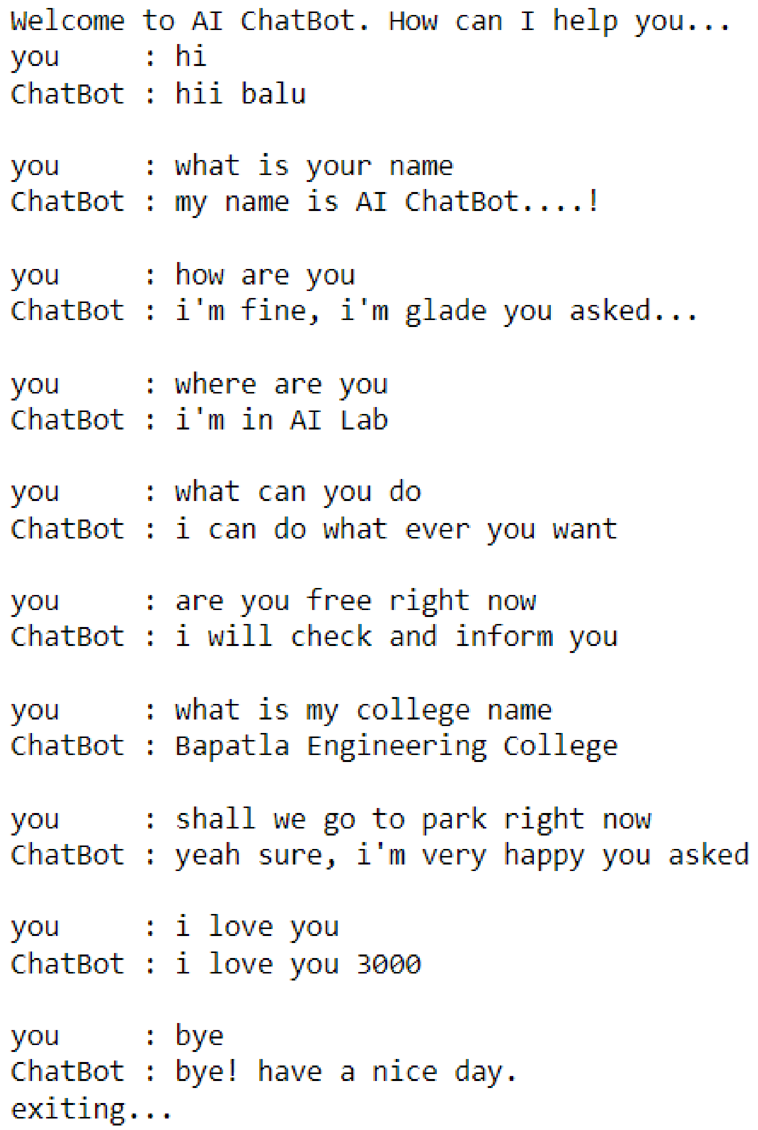
```
addition of x, y = 8
subtration of x,y = 2
multiplication ofx,y = 15
division of x,y = 1.6666666666666667
floor division of x,y = 1
modulo division of x,y= 2
power of x,y = 125
mean : 3.4
Median:  3.5
Mode: 3.7
Factorial of 10 is :  3628800
Quadratic quation: X^2-4X+4 :
x1= 2.0 x2= 2.0
```

Aim: 3. Write a python program to implement chatbot program.

Source code:

```
reflections={
    "hi":"hii balu",
    "how are you":"i'm fine, i'm glade you asked...",
    "where are you":"i'm in AI Lab",
    "bye":"bye! have a nice day.",
    "are you free right now":"i will check and inform you",
    "what can you do":"i can do what ever you want",
    "what is my college name":"Bapatla Engineering College",
    "shall we go to park right now":"yeah sure, i'm very happy you asked",
    "what is your name":"my name is AI ChatBot....!",
    "what is your hobbies":"reading novels,palying cricket.",
    "i love you":"i love you 3000",
    "i hate you":"i'm sorry for disturbing you..."
}
print("Welcome to AI ChatBot. How can I help you...")
while True:
    s=input("you     : ")
    if s in reflections.keys():
        print("ChatBot :",reflections[s].capitalize())
        if s=="bye":
            print("exiting...")
            break
    else:
        print("ChatBot : Sorry, I didn't understand...")
    print()
```

Output:



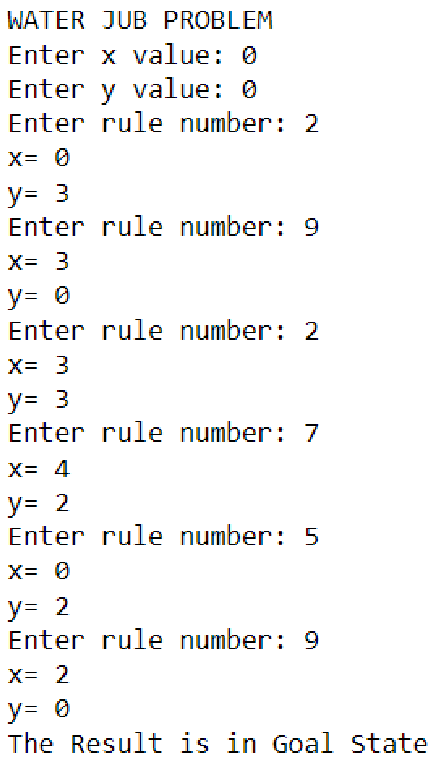
```
Welcome to AI ChatBot. How can I help you...
you     : hi
ChatBot : hii balu

you     : what is your name
ChatBot : my name is AI ChatBot....!

you     : how are you
ChatBot : i'm fine, i'm glade you asked...

you     : where are you
ChatBot : i'm in AI Lab

you     : what can you do
ChatBot : i can do what ever you want

you     : are you free right now
ChatBot : i will check and inform you

you     : what is my college name
ChatBot : Bapatla Engineering College

you     : shall we go to park right now
ChatBot : yeah sure, i'm very happy you asked

you     : i love you
ChatBot : i love you 3000

you     : bye
ChatBot : bye! have a nice day.
exiting...
```

Aim: 4. Write a python program to implement Water jug problem.

Source code:

```
print("WATER JUB PROBLEM")
x=int(input("Enter x value: "))
y=int(input("Enter y value: "))
while True:
    rule=int(input("Enter rule number: ")) #rules order:2, 9, 2, 7, 5, 9
    if rule == 1:
        if x<4:
            x=4
    elif rule == 2:
        if y<3:
            y=3
    elif rule == 5:
        if x>0:
            x=0
    elif rule == 6:
        if y>0:
            y=0
    elif rule == 7:
        if (x+y)>=4 and y>0:
            x,y=4,y-(4-x)
    elif rule == 8:
        if (x+y)>=3 and x>0:
            x,y=x-(3-x),3
    elif rule == 9:
        if (x+y)<=4 and y>0:
            x,y=x+y,0
    elif rule == 10:
        if (x+y)<=3 and x>0:
            x,y=0,y+x

    print("x=",x)
    print("y=",y)

    if x==2:
        print("The Result is in Goal State")
        break
```

Output:



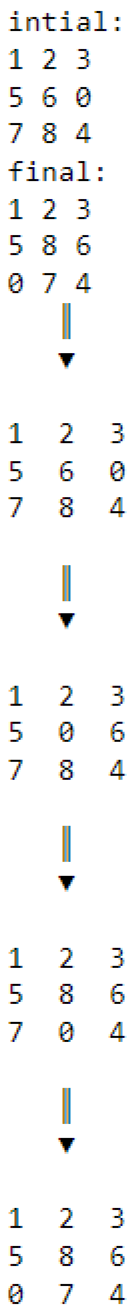
```
WATER JUB PROBLEM
Enter x value: 0
Enter y value: 0
Enter rule number: 2
x= 0
y= 3
Enter rule number: 9
x= 3
y= 0
Enter rule number: 2
x= 3
y= 3
Enter rule number: 7
x= 4
y= 2
Enter rule number: 5
x= 0
y= 2
Enter rule number: 9
x= 2
y= 0
The Result is in Goal State
```

Aim: 5. Write a python program to implement 8-puzzle problem.

Source code:

```
import copy
from heapq import heappush, heappop
puzzle(n=5)...
n = 3
row = [ 1, 0, -1, 0 ]
col = [ 0, -1, 0, 1 ]
class priorityQueue:
    def __init__(self):
        self.heap = []

    def push(self, k):
        heappush(self.heap, k)

    def pop(self):
        return heappop(self.heap)

    def empty(self):
        if not self.heap:
            return True
        else:
            return False

class node:
    def __init__(self, parent, mat, empty_tile_pos, cost, level):
        self.parent = parent
        self.mat = mat
        self.empty_tile_pos = empty_tile_pos
        self.cost = cost
        self.level = level

    def __lt__(self, nxt):
        return self.cost < nxt.cost

def calculateCost(mat, final) -> int:
    count = 0
    for i in range(n):
        for j in range(n):
            if ((mat[i][j]) and
                (mat[i][j] != final[i][j])):
                count += 1
    return count

def newNode(mat, empty_tile_pos, new_empty_tile_pos, level, parent, final):
    new_mat = copy.deepcopy(mat)
    x1 = empty_tile_pos[0]
    y1 = empty_tile_pos[1]
    x2 = new_empty_tile_pos[0]
    y2 = new_empty_tile_pos[1]
    new_mat[x1][y1], new_mat[x2][y2] = new_mat[x2][y2], new_mat[x1][y1]
    cost = calculateCost(new_mat, final)
    new_node = node(parent, new_mat, new_empty_tile_pos, cost, level)
    return new_node

def printMatrix(mat):
    print('    ‖')
    print('    ▼\n')
    for i in range(n):
        for j in range(n):
            print("%d " % (mat[i][j]), end = " ")

        print()
def isSafe(x, y):
    return x >= 0 and x < n and y >= 0 and y < n

def printPath(root):
    if root == None:
        return
    printPath(root.parent)
    printMatrix(root.mat)
    print()

def solve(initial, empty_tile_pos, final):

    pq = priorityQueue()
    cost = calculateCost(initial, final)
    root = node(None, initial,
                empty_tile_pos, cost, 0)

    pq.push(root)

    while not pq.empty():

        minimum = pq.pop()
        if minimum.cost == 0:
            printPath(minimum)
            return

        for i in range(n):
            new_tile_pos = [
                minimum.empty_tile_pos[0] + row[i],
                minimum.empty_tile_pos[1] + col[i], ]

            if isSafe(new_tile_pos[0], new_tile_pos[1]):

                child = newNode(minimum.mat,
                                minimum.empty_tile_pos,
                                new_tile_pos,
                                minimum.level + 1,
                                minimum, final,)
            pq.push(child)

initial=[]
print("intial:")
for i in range(3):
   a=list(map(int,input().split()))
   initial.append(a)

print("final:")
final=[]
for i in range(3):
   a=list(map(int,input().split()))
   final.append(a)

#Blank tile coordinates in initial configuration
empty_tile_pos = [ 1, 2 ]

# Function call to solve the puzzle
solve(initial1, empty_tile_pos, final1)
```

Output:



```
intial:
1 2 3
5 6 0
7 8 4
final:
1 2 3
5 8 6
0 7 4
    ↓

1   2   3
5   6   0
7   8   4

    ↓

1   2   3
5   0   6
7   8   4

    ↓

1   2   3
5   8   6
7   0   4

    ↓

1   2   3
5   8   6
0   7   4
```

Aim: 6. Write a python program to implement BFS and DFS.

Source code of BFS:

```
from collections import defaultdict

class Graph:

    def __init__(self):

        self.graph = defaultdict(list)

    def addEdge(self,u,v):
        self.graph[u].append(v)

    def BFS(self, s):
        visited = [False] * (max(self.graph) + 1)

        queue = []

        queue.append(s)
        visited[s] = True

        while queue:

            s = queue.pop(0)
            print (s, end = " ")

            for i in self.graph[s]:
                if visited[i] == False:
                    queue.append(i)
                    visited[i] = True

g = Graph()
g.addEdge(0, 1)
g.addEdge(0, 2)
g.addEdge(1, 2)
g.addEdge(2, 0)
g.addEdge(2, 3)
g.addEdge(3, 3)

print ("Breadth First Traversal"
              " (starting from vertex 2)")
g.BFS(2)
```

Output:

```
Breadth First Traversal (starting from vertex 2)
2 0 3 1
```

Source code for DFS:

```
from collections import defaultdict


class Graph:
    def __init__(self):
        self.graph = defaultdict(list)

    def addEdge(self, u, v):
        self.graph[u].append(v)
    def DFSUtil(self, v, visited):

        visited.add(v)
        print(v, end=' ')

        for neighbour in self.graph[v]:
            if neighbour not in visited:
                self.DFSUtil(neighbour, visited)

    def DFS(self, v):
        visited = set()

        self.DFSUtil(v, visited)

g = Graph()
g.addEdge(0, 1)
g.addEdge(0, 2)
g.addEdge(1, 2)
g.addEdge(2, 0)
g.addEdge(2, 3)
g.addEdge(3, 3)

print("DFS from (starting from vertex 2)")
g.DFS(2)
```

Output:

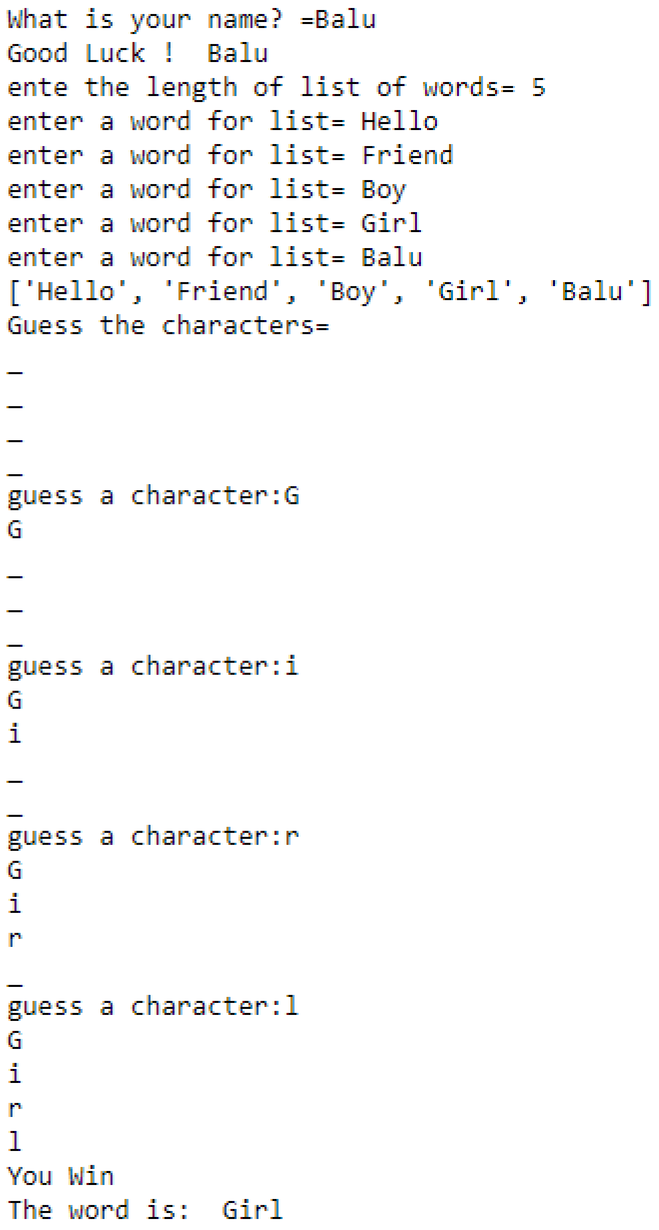
```
DFS from (starting from vertex 2)
2 0 1 3
```

Aim:7. Write a python program to implement Hangman game

Source code:

```
import random
name = input("What is your name? =")
print("Good Luck ! ", name)
length=int(input("ente the length of list of words= "))
words=[]
for i in range(0,length):
    word=str(input("enter a word for list= "))
    words.append(word)
print(words)
word = random.choice(words)
print("Guess the characters= ")
guesses = ''
turns = 12
while turns > 0:
    failed = 0
    for char in word:
        if char in guesses:
            print(char)
        else:
            print("_")
            failed += 1
    if failed == 0:
        print("You Win")
        print("The word is: ", word)
        break
    guess = input("guess a character:")
    guesses += guess
    if guess not in word:
        turns -= 1
        print("Wrong")
        print("You have", + turns, 'more guesses')
        if turns == 0:
            print("You Loose")
```

Output:



```
What is your name? =Balu
Good Luck !  Balu
ente the length of list of words= 5
enter a word for list= Hello
enter a word for list= Friend
enter a word for list= Boy
enter a word for list= Girl
enter a word for list= Balu
['Hello', 'Friend', 'Boy', 'Girl', 'Balu']
Guess the characters=
_
_
_
_
guess a character:G
G
_
_
_
guess a character:i
G
i
_
_
guess a character:r
G
i
r
_
guess a character:l
G
i
r
l
You Win
The word is:  Girl
```

Aim: 8. Write a python program to remove stop words for a given passage from a text file using NLTK.

Source code:

```
import nltk
from nltk.corpus import stopwords
f1=open("file1.txt","r")
f2=open("file2.txt","w")
stop=stopwords.words('english')
print(stop)
for line in f1:
    w=line.split(" ")
    print(line)
    for word in w:
        if word not in stop:
            f2.write(word)
            print(word)
            f2.write("")
f1.close()
f2.close()
```

Output:

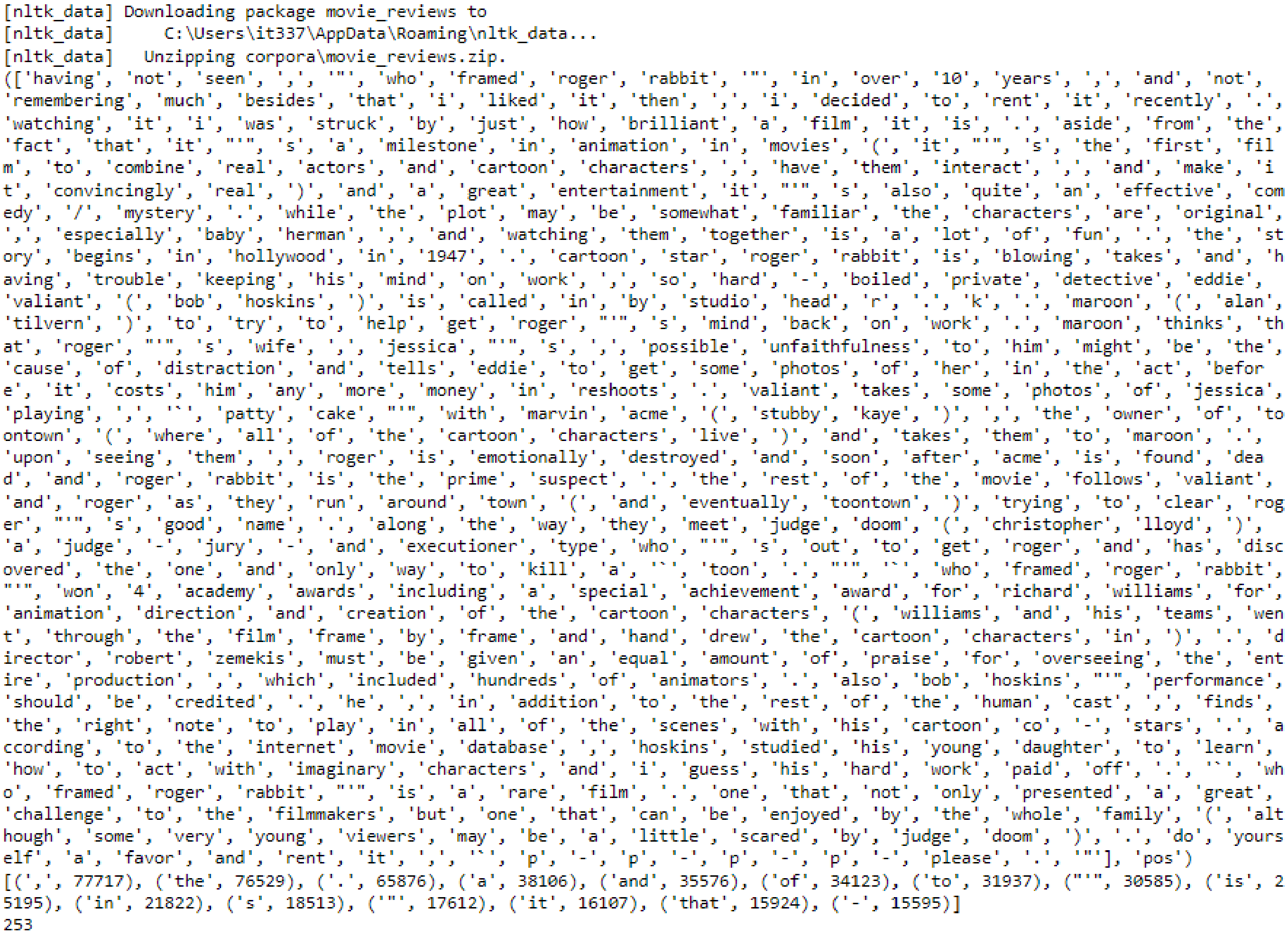
```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you
've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'h
is', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'its
elf', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'who
m', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were',
'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing'
, 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'o
f', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'd
uring', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out
', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there
', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'mos
t', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than
', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "shoul
d've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'could
n', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "ha
sn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mu
stn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn
't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]

Rohit Gurunath Sharma (born 30 April 1987) is an Indian international cricketer who
is the current captain of the Indian national team.

Rohit
Gurunath
Sharma
(born
30
April
1987)
Indian
international
cricketer
current
captain
Indian
national
team.
```

Aim: 9. Write a python program to form text classification to the given sentence using NLTK.

Source code:

```
import nltk
import random
nltk.download("movie_reviews")
from nltk.corpus import movie_reviews
documents=[(list(movie_reviews.words(fileid)),category)
    for category in movie_reviews.categories()
    for fileid in movie_reviews.fileids(category)]
random.shuffle(documents)
print(documents[1])
all_words=[]
for w in movie_reviews.words():
    all_words.append(w.lower())
all_words=nltk.FreqDist(all_words)
print(all_words.most_common(15))
print(all_words["stupid"])
```

Output:



```
[nltk_data] Downloading package movie_reviews to
[nltk_data]     C:\Users\it337\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\movie_reviews.zip.
(['having', 'not', 'seen', ',', '"', 'who', 'framed', 'roger', 'rabbit', '"', 'in', 'over', '10', 'years', ',', 'and', 'not',
'remembering', 'much', 'besides', 'that', 'i', 'liked', 'it', 'then', ',', 'i', 'decided', 'to', 'rent', 'it', 'recently', '.',
'watching', 'it', 'i', 'was', 'struck', 'by', 'just', 'how', 'brilliant', 'a', 'film', 'it', 'is', '.', 'aside', 'from', 'the',
'fact', 'that', 'it', "'", 's', 'a', 'milestone', 'in', 'animation', 'in', 'movies', '(', 'it', "'", 's', 'the', 'first', 'fil
m', 'to', 'combine', 'real', 'actors', 'and', 'cartoon', 'characters', ',', 'have', 'them', 'interact', ',', 'and', 'make', 'i
t', 'convincingly', 'real', ')', 'and', 'a', 'great', 'entertainment', 'it', "'", 's', 'also', 'quite', 'an', 'effective', 'com
edy', '/', 'mystery', '.', 'while', 'the', 'plot', 'may', 'be', 'somewhat', 'familiar', 'the', 'characters', 'are', 'original',
',', 'especially', 'baby', 'herman', ',', 'and', 'watching', 'them', 'together', 'is', 'a', 'lot', 'of', 'fun', '.', 'the', 'st
ory', 'begins', 'in', 'hollywood', 'in', '1947', '.', 'cartoon', 'star', 'roger', 'rabbit', 'is', 'blowing', 'takes', 'and', 'h
aving', 'trouble', 'keeping', 'his', 'mind', 'on', 'work', ',', 'so', 'hard', '-', 'boiled', 'private', 'detective', 'eddie',
'valiant', '(', 'bob', 'hoskins', ')', 'is', 'called', 'in', 'by', 'studio', 'head', 'r', '.', 'k', '.', 'maroon', '(', 'alan',
'tilvern', ')', 'to', 'try', 'to', 'help', 'get', 'roger', "'", 's', 'mind', 'back', 'on', 'work', '.', 'maroon', 'thinks', 'th
at', 'roger', "'", 's', 'wife', ',', 'jessica', "'", 's', ',', 'possible', 'unfaithfulness', 'to', 'him', 'might', 'be', 'the',
'cause', 'of', 'distraction', 'and', 'tells', 'eddie', 'to', 'get', 'some', 'photos', 'of', 'her', 'in', 'the', 'act', 'befor
e', 'it', 'costs', 'him', 'any', 'more', 'money', 'in', 'reshoots', '.', 'valiant', 'takes', 'some', 'photos', 'of', 'jessica',
'playing', ',', '`', 'patty', 'cake', "'", 'with', 'marvin', 'acme', '(', 'stubby', 'kaye', ')', ',', 'the', 'owner', 'of', 'to
ontown', '(', 'where', 'all', 'of', 'the', 'cartoon', 'characters', 'live', ')', 'and', 'takes', 'them', 'to', 'maroon', '.',
'upon', 'seeing', 'them', ',', 'roger', 'is', 'emotionally', 'destroyed', 'and', 'soon', 'after', 'acme', 'is', 'found', 'dea
d', 'and', 'roger', 'rabbit', 'is', 'the', 'prime', 'suspect', '.', 'the', 'rest', 'of', 'the', 'movie', 'follows', 'valiant',
'and', 'roger', 'as', 'they', 'run', 'around', 'town', '(', 'and', 'eventually', 'toontown', ')', 'trying', 'to', 'clear', 'rog
er', "'", 's', 'good', 'name', '.', 'along', 'the', 'way', 'they', 'meet', 'judge', 'doom', '(', 'christopher', 'lloyd', ')',
'a', 'judge', '-', 'jury', '-', 'and', 'executioner', 'type', 'who', "'", 's', 'out', 'to', 'get', 'roger', 'and', 'has', 'disc
overed', 'the', 'one', 'and', 'only', 'way', 'to', 'kill', 'a', '`', 'toon', '.', "'", '`', 'who', 'framed', 'roger', 'rabbit',
"'", 'won', '4', 'academy', 'awards', 'including', 'a', 'special', 'achievement', 'award', 'for', 'richard', 'williams', 'for',
'animation', 'direction', 'and', 'creation', 'of', 'the', 'cartoon', 'characters', '(', 'williams', 'and', 'his', 'teams', 'wen
t', 'through', 'the', 'film', 'frame', 'by', 'frame', 'and', 'hand', 'drew', 'the', 'cartoon', 'characters', 'in', ')', '.', 'd
irector', 'robert', 'zemekis', 'must', 'be', 'given', 'an', 'equal', 'amount', 'of', 'praise', 'for', 'overseeing', 'the', 'ent
ire', 'production', ',', 'which', 'included', 'hundreds', 'of', 'animators', '.', 'also', 'bob', 'hoskins', "'", 'performance',
'should', 'be', 'credited', '.', 'he', ',', 'in', 'addition', 'to', 'the', 'rest', 'of', 'the', 'human', 'cast', ',', 'finds',
'the', 'right', 'note', 'to', 'play', 'in', 'all', 'of', 'the', 'scenes', 'with', 'his', 'cartoon', 'co', '-', 'stars', '.', 'a
ccording', 'to', 'the', 'internet', 'movie', 'database', ',', 'hoskins', 'studied', 'his', 'young', 'daughter', 'to', 'learn',
'how', 'to', 'act', 'with', 'imaginary', 'characters', 'and', 'i', 'guess', 'his', 'hard', 'work', 'paid', 'off', '.', '`', 'wh
o', 'framed', 'roger', 'rabbit', "'", 'is', 'a', 'rare', 'film', '.', 'one', 'that', 'not', 'only', 'presented', 'a', 'great',
'challenge', 'to', 'the', 'filmmakers', 'but', 'one', 'that', 'can', 'be', 'enjoyed', 'by', 'the', 'whole', 'family', '(', 'alt
hough', 'some', 'very', 'young', 'viewers', 'may', 'be', 'a', 'little', 'scared', 'by', 'judge', 'doom', ')', '.', 'do', 'yours
elf', 'a', 'favor', 'and', 'rent', 'it', ',', '`', 'p', '-', 'p', '-', 'p', '-', 'p', '-', 'please', '.', '"'], 'pos')
[(',', 77717), ('the', 76529), ('.', 65876), ('a', 38106), ('and', 35576), ('of', 34123), ('to', 31937), ("'", 30585), ('is', 2
5195), ('in', 21822), ('s', 18513), ('"', 17612), ('it', 16107), ('that', 15924), ('-', 15595)]
253
```

Aim: 10. Write a python program to implement A* algorithm.

Source code:

```
def aStarAlgo(start_node, stop_node):
        open_set = set(start_node)
        closed_set = set()
        g = {}
        parents = {}
        g[start_node] = 0
        parents[start_node] = start_node
        while len(open_set) > 0:
            n = None
            for v in open_set:
                if n == None or g[v] + heuristic(v) < g[n] + heuristic(n):
                    n = v

            if n == stop_node or Graph_nodes[n] == None:
                pass
            else:
                for (m, weight) in get_neighbors(n):
                    if m not in open_set and m not in closed_set:
                        open_set.add(m)
                        parents[m] = n
                        g[m] = g[n] + weight

                    else:
                        if g[m] > g[n] + weight:
                            g[m] = g[n] + weight
                            parents[m] = n
                            if m in closed_set:
                                closed_set.remove(m)
                                open_set.add(m)

            if n == None:
                print('Path does not exist!')
                return None

            if n == stop_node:
                path = []
                while parents[n] != n:
                    path.append(n)
                    n = parents[n]
                path.append(start_node)
                path.reverse()
                print('Path found: {}'.format(path))
                return path

            open_set.remove(n)
            closed_set.add(n)

        print('Path does not exist!')
        return None
```

```
def get_neighbors(v):
    if v in Graph_nodes:
        return Graph_nodes[v]
    else:
        return None
def heuristic(n):
        H_dist = {
            'A': 11,
            'B': 6,
            'C': 99,
            'D': 1,
            'E': 7,
            'G': 0,

        }
        return H_dist[n]

#Describe your graph here
Graph_nodes = {
    'A': [('B', 2), ('E', 3)],
    'B': [('C', 1),('G', 9)],
    'C': None,
    'E': [('D', 6)],
    'D': [('G', 1)],

}
aStarAlgo('A', 'G')
```

Output:

```
Path found: ['A', 'E', 'D', 'G']
```

Aim: 11. Write a python program to implement Tic Tac Toe.

Source code:

```
import numpy as np
import random
from time import sleep
def create_board():
    return(np.array([[0, 0, 0], [0, 0, 0], [0, 0, 0]]))
def possibilities(board):
    l = []
    for i in range(len(board)):
        for j in range(len(board)):
            if board[i][j] == 0:
                l.append((i, j))
    return(l)

def random_place(board, player):
    selection = possibilities(board)
    current_loc = random.choice(selection)
    board[current_loc] = player
    return(board)

def row_win(board, player):
    for x in range(len(board)):
        win = True

        for y in range(len(board)):
            if board[x, y] != player:
                win = False
                continue

        if win == True:
            return(win)
    return(win)

def col_win(board, player):
    for x in range(len(board)):
        win = True

        for y in range(len(board)):
            if board[y][x] != player:
                win = False
                continue

        if win == True:
            return(win)
    return(win)

def diag_win(board, player):
    win = True
    y = 0
    for x in range(len(board)):
        if board[x, x] != player:
            win = False
    if win:
        return win
    win = True
    if win:
        for x in range(len(board)):
            y = len(board) - 1 - x
            if board[x, y] != player:
                win = False
    return win

def evaluate(board):
    winner = 0

    for player in [1, 2]:
        if (row_win(board, player) or
            col_win(board,player) or
            diag_win(board,player)):

            winner = player

    if np.all(board != 0) and winner == 0:
        winner = -1
    return winner

def play_game():
    board, winner, counter = create_board(), 0, 1
    print(board)
    sleep(2)

    while winner == 0:
        for player in [1, 2]:
            board = random_place(board, player)
            print("Board after " + str(counter) + " move")
            print(board)
            sleep(2)
            counter += 1
            winner = evaluate(board)
            if winner != 0:
                break
    return(winner)

print("Winner is: " + str(play_game()))
```

Output:

```
[[0 0 0]
 [0 0 0]
 [0 0 0]]
Board after 1 move
[[0 0 1]
 [0 0 0]
 [0 0 0]]
Board after 2 move
[[0 2 1]
 [0 0 0]
 [0 0 0]]
Board after 3 move
[[0 2 1]
 [0 0 1]
 [0 0 0]]
Board after 4 move
[[0 2 1]
 [0 0 1]
 [2 0 0]]
Board after 5 move
[[0 2 1]
 [0 0 1]
 [2 1 0]]
Board after 6 move
[[2 2 1]
 [0 0 1]
 [2 1 0]]
Board after 7 move
[[2 2 1]
 [0 0 1]
 [2 1 1]]
Winner is: 1
```

Aim: 12. Write a Program to implement a simple Robot.

.

File(a): build_a_voice_controllered_robot

Source code:

```
#include <SoftwareSerial.h>
SoftwareSerial BT(2, 3); //TX, RX respetively
String readvoice;

#define MLa 8                  //left motor 1st pin
#define MLb 9                 //left motor 2nd pin
#define MRa 10              //right motor 1st pin
#define MRb 11              //right motor 2nd pin

void setup() {
 BT.begin(9600);
 Serial.begin(9600);
  pinMode(MLa, OUTPUT);// declaring Motors pin as output pin
  pinMode(MLb, OUTPUT);
  pinMode(MRa, OUTPUT);
  pinMode(MRb, OUTPUT);

}

void loop() {
  while (BT.available())
  {  //Check if there is an available byte to read
  delay(10); //Delay added to make thing stable
  char c = BT.read(); //Conduct a serial read
  readvoice += c; //build the string- "forward", "reverse", "left" and "right"
  }
  if (readvoice.length() > 0)
  {
    Serial.println(readvoice);
    if(readvoice == "forward")
    {
      //forward
      digitalWrite(MLa, HIGH);
      digitalWrite(MLb, LOW);
      digitalWrite(MRa, HIGH);
      digitalWrite(MRb, LOW);
      delay(100);
    }
    else if(readvoice == "back")
    {
      //backward
      digitalWrite(MLa, LOW);
      digitalWrite(MLb, HIGH);
      digitalWrite(MRa, LOW);
      digitalWrite(MRb, HIGH);
      delay(100);
    }
```

```
   else if (readvoice == "turn right" )
   {
     //Rightturn
     digitalWrite(MLa, HIGH);
     digitalWrite(MLb, LOW);
     digitalWrite(MRa, LOW);
     digitalWrite(MRb, LOW);
     delay (100);

   }
   else if ( readvoice == "turn left")
  {
     //LeftTurn
     digitalWrite(MLa, LOW);
     digitalWrite(MLb, LOW);
     digitalWrite(MRa, HIGH);
     digitalWrite(MRb, LOW);
    delay (100);
  }
  else if (readvoice == "stop")
  {
     //Stop
     digitalWrite(MLa, LOW);
     digitalWrite(MLb, LOW);
     digitalWrite(MRa, LOW);
     digitalWrite(MRb, LOW);
     delay (100);
  }

 readvoice="";}//Reset the variable

}
```

File(b): interfacing_Buetooth_With_Ardino
Source Code:

```
#define ledPin 13  // we can use builtin led
int led_last_state =0;
void setup()
{
  pinMode(ledPin,OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  if(Serial.available())
  {
    char data=Serial.read();

      if(data=='1')
      {
        digitalWrite(ledPin,HIGH);
      Serial.println("LED is ON");l
      }

       else if (data=='0')
      {
        digitalWrite(ledPin,LOW);
        Serial.println("led is OFF");
      }
  }
}
```

File : build_a_Motion_Robot
Source Code:

```
#define MLa 8     //left motor 1st pin
#define MLb 9     //left motor 2nd pin
#define MRa 10    //right motor 1st pin
#define MRb 11    //right motor 2nd pin
void setup()
{
  pinMode(MLa, OUTPUT);
  pinMode(MLb, OUTPUT);
  pinMode(MRa, OUTPUT);
  pinMode(MRb, OUTPUT);
}

void loop()
{
  //stop both the motors//
    digitalWrite(MLa, LOW);
    digitalWrite(MLb, LOW);
    digitalWrite(MRa, LOW);
    digitalWrite(MRb, LOW);
    delay(1000);    //wait for 1 second//

  //Rotate both Motors in Forward Direction//
    digitalWrite(MLa, HIGH);
    digitalWrite(MLb, LOW);
    digitalWrite(MRa, HIGH);
    digitalWrite(MRb, LOW);
    delay(2000);            //wait for 2 second//

    //Rotate both Motors in Backward Direction//
    digitalWrite(MLa, LOW);
    digitalWrite(MLb, HIGH);
    digitalWrite(MRa, LOW);
    digitalWrite(MRb, HIGH);
    delay(2000);           //wait for 2 second//

   //Rotate robot in right direction
```

```
    digitalWrite(MLa, HIGH);    //Rotate left motor in fwd direction
    digitalWrite(MLb, LOW);
    digitalWrite(MRa, LOW);     //Rotate right motor in back direction
    digitalWrite(MRb, HIGH);
    delay(2000);         //wait for 2 second//

     //Rotate robot in left direction
    digitalWrite(MLa, LOW);      //Rotate left motor in fwd direction
    digitalWrite(MLb, HIGH);
    digitalWrite(MRa, HIGH);     //Rotate right motor in back direction
    digitalWrite(MRb, LOW);
    delay(2000);         //wait for 2 second//
}
```