**PROGRAM 19**

INPUT:

#tictactoe

import random

a=['1','2','3','4','5','6','7','8','9']

def printboard():

print('.......')

print(a[0]+'|'+a[1]+'|'+a[2]+'|')

print('......')

print(a[3]+'|'+a[4]+'|'+a[5]+'|')

print('......')

print(a[6]+'|'+a[7]+'|'+a[8]+'|')

print('.......')

printboard()

p1 = True

while(True):

printboard()

if p1:

p=input("player1, you have to choose your place : ")

if p in a:

a[int(p)-1]='x'

p1= not p1

else:

p=input("player 2, you have to choose your place : ")

if p in a:

a[int(p)-1]='0'

p1=not p1

for i in(0,3,6):

if(a[i]==a[i+1] and a[i]==a[i+2]):

print("game over")

printboard()

exit()

for i in range(3):

if(a[i]==a[i+3] and a[i]==a[i+6]):

print("game ends")

printboard()

exit()

if(a[0]==a[4] and a[0]==a[8]):

print("game ends")

printboard()

exit()

if(a[2]==a[4] and a[2]==a[6]):

print("game ends")

printboard()

exit()

else:

print("invalid position")

continue

OUTPUT:

.......

1|2|3|

......

4|5|6|

......

7|8|9|

.......

.......

1|2|3|

......

4|5|6|

......

7|8|9|

.......

player1, choose your place : 1

invalid position

.......

x|2|3|

......

4|5|6|

......

7|8|9|

.......

player 2, choose your place : 3

invalid position

.......

x|2|0|

......

4|5|6|

......

7|8|9|

.......

player1, choose your place :

invalid position

.......

x|2|0|

......

x|5|6|

......

7|8|9|

.......

player 2, choose your place : 6

invalid position

.......

x|2|0|

......

x|5|0|

......

7|8|9|

.......

player1, choose your place : 2

invalid position

.......

x|x|0|

......

x|5|0|

......

7|8|9|

.......

player 2, choose your place : 9

game ends

.......

x|x|0|

......

x|5|0|

......

7|8|0|

.......

**PROGRAM 20**

INPUT:

#pic.py

import os

def takepic():

os.system("fswebcam -F 4 --fps 20 -r 800\*600 /home/cl215/sharin/temp.jpg")

print("pic taken ")

return

takepic()

**PROGRAM 21**

INPUT:

#speech

import speech\_recognition as sr

#Record audio

r = sr.Recognizer()

with sr.Microphone() as source:

r.adjust\_for\_ambient\_noise(source,duration=5)

r.dynamic\_energy\_threshold = True

print("speak something!")

audio = r.listen(source)

try:

print("you said: " + r.recognize\_google(audio))

except sr.UnKnownValveError:

print("Google Speech Recognition could not understand audio")

except sr.RequestError as e:

print("could not request results from Google Speech Recognition service; {0}".format(e))

OUTPUT:

speak something!

you said: hello how are you keeping

**PROGRAM 22**

INPUT:

#Libraries

import RPi.GPIO as GPIO

import time

#GPIO Mode (BOARD / BCM)

GPIO.setmode(GPIO.BCM)

#set GPIO Pins

GPIO\_TRIGGER = 18

GPIO\_ECHO = 24

#set GPIO direction (IN / OUT)

GPIO.setup(GPIO\_TRIGGER, GPIO.OUT)

GPIO.setup(GPIO\_ECHO, GPIO.IN)

def distance():

# set Trigger to HIGH

GPIO.output(GPIO\_TRIGGER, True)

# set Trigger after 0.01ms to LOW

time.sleep(0.00001)

GPIO.output(GPIO\_TRIGGER, False)

StartTime = time.time()

StopTime = time.time()

# save StartTime

while GPIO.input(GPIO\_ECHO) == 0:

StartTime = time.time()

# save time of arrival

while GPIO.input(GPIO\_ECHO) == 1:

StopTime = time.time()

# time difference between start and arrival

TimeElapsed = StopTime - StartTime

# multiply with the sonic speed (34300 cm/s)

# and divide by 2, because there and back

distance = (TimeElapsed \* 34300) / 2

return distance

if \_\_name\_\_ == '\_\_main\_\_':

try:

while True:

dist = distance()

print ("Measured Distance = %.1f cm" % dist)

time.sleep(1)

if (dist>100):

print("distance is above 100")

break

# Reset by pressing CTRL + C

except KeyboardInterrupt:

print("Measurement of distance has been stopped by User")

GPIO.cleanup()

OUTPUT:

Measured distance = 4.8 cm

Measured distance = 5.4 cm

Measured distance = 5.8 cm

Measured distance = 4.0 cm

Measured distance = 5.6 cm

Measured distance = 6.3 cm

Measured distance = 9.5 cm

Measured distance = 5.2 cm

Measured distance = 1585 cm

distance is above 100.