

Name

Mukund Kukreja

Year

2021-25

Subject

AI Practical File

Class

AIML

Semester

III<sup>rd</sup>

Roll No.

0206AI211059

# INDEX

Sr. No.	Experiment Description	Experiment Date	Submission Date	Remarks / Signature
1.	Breadth First Search			
2.	Depth First Search			
3.	ML Kit - Tiny ML			
4.	Program to Find Factorial			
5.	Arduino : Rain Sensor			
6.	Recommender System			
7.	Arduino : Servo Motor			
8.	Arduino : LED & Buzzer			
9.	Expert system using Prolog			
10.	Artificial Neural Network			
11.	Regular Expression			
12.	Regex : Search string in text			
13.	Regex : Find contact number, email & date in text			

## Experiment - 1

Aim- Implementation of Breadth First Search (BFS) algorithm in Python.

```
import collections
```

```
def BFS (graph, root):
```

```
    visited, queue = set(), collections.deque([root])  
    visited.add(root)
```

```
    while queue:
```

```
# dequeue a vertex from queue
```

```
    vertex = queue.popleft()
```

```
    print(str(vertex) + " ", end="")
```

```
# if not visited, mark it as visited
```

```
# and enqueue it for neighbour in
```

```
for neighbour in graph[vertex]:
```

```
    if neighbour not in visited:
```

```
        visited.add(neighbour)
```

```
        queue.append(neighbour)
```

```
if __name__ == "__main__":
```

```
graph = {0: [1, 2], 1: [2], 2: [3], 3: [1, 2]}
```

```
print("Following is Breadth First Traversal: ")
```

```
BFS(graph, 0)
```

## Experiment-2

Aim- Implementation of Depth First Search (DFS) algorithm in python.

```
def DFS(graph, start, visited=None):
    if visited is None:
        visited = set()
    visited.add(start)
    print(start)
```

```
for next in graph[start] - visited:
    DFS(graph, next, visited)
return visited
```

```
graph = { '0' : set(['1','2']),
          '1' : set(['0','3','4']),
          '2' : set(['0']),
          '3' : set(['1']),
          '4' : set(['2','3'])}
```

DFS(graph, '0')

## Experiment - 3

Aim- Introduction to ML kit Tiny ML (Machine learning kit).

Material- Tiny Machine learning kit

Required

Theory- Tiny ML kit consists of a powerful board equipped with a micro controller and a wide variety of sensors like Arduino Nano 33 BLE. This board can sense moment, acceleration, rotation, temperature sensor, humidity, barometric pressure, sounds, gestures, proximity, color and light intensity.

This kit includes a camera module (OV7675) and custom Arduino shield.

The tiny ML kits includes

- \* 1- Arduino Nano 33 BLE Sense Board
- \* 1- OV 7675 Camera.
- \* 1- Arduino Tiny ML Shield.
- \* 1- USB A to Micro USB Cable.

\* Libraries required to work with ML kit -

There are few important libraries you need to install to start with tiny ML kit -

- \* Arduino APDS9960
- \* Arduino BLE
- \* Arduino LSM9051
- \* Arduino HTS 221
- \* Tensor flowlite
- \* Arduino Mbed OS nano boards

## Experiment - 4

Aim - Write a program to print factorial of a given number.

# python program to print Factorial

```
i = int(input(" Enter Number : " ))
```

```
fact = 1
```

```
while (i > 0):
```

```
    fact = fact * i
```

```
    i -= 1
```

```
print (" Factorial = " , fact)
```

## Experiment - S

Aim- To stimulate the rain sensor using arduino UNO and water sensor.

Theory - The water sensor is used to detect water leakage, rainfall, tank overflow, to measure water level.

The water sensor has 3 pins:-

- ① S (Signal pin) - It is an analog output that will be connected to one of the analog input on Arduino.
- ② + (Vcc) pin - It supplies power for sensor
- ③ - (GND) pin - It is a ground connection.

// code to stimulate rain detection.

```
int rain = A0;      int val = 0;  
int alarm = 4;      int ledGr = 4;
```

```
void setup() { Serial.begin(9600);  
  pinMode(rain, INPUT);  
  pinMode(ledGr, OUTPUT);  
  pinMode(alarm, OUTPUT); }
```

```
void loop() { val = analogRead(rain);  
  Serial.println(val);  
  if (val >= 40) { Serial.println("Rain detected");  
    digitalWrite(alarm, HIGH);  
    digitalWrite(ledGr, HIGH); }  
  else { Serial.println("No rain detected");  
    digitalWrite(ledGr, LOW);  
    digitalWrite(alarm, LOW); } }
```

## Experiment - 6

Page: 6

Aim- To study recommender system

Recommender System is a software system that provides specific suggestions to users according to their preference. These suggestions may provide decision making capabilities to the user. Items refer to any product that the recommender system suggests to its user like movies, music, news, travel packages, e-commerce products etc.

Benefits of a recommender system:-

① Increasing profit by increasing number of items sold:

One of the major objectives of building commercial recommender systems is to enhance business and increase profit. This could be done by suggesting users new items which may attract the users and they may buy more items as compared to those without recommender systems.

② Enhanced User Satisfaction :

A well designed recommender system enhances the user's overall experience in using that application. They may find recommendations useful and relevant to user needs. So the major purpose of recommender system is to satisfy the users and make them happy.

③ Extraction of useful patterns : The recommender system provides a way of extracting useful pattern of user's needs & preferences that could serve as strategic information for the business.

④ Provides more diverse items to user : Sometimes it is impossible to find the items personally that a recommender system may provide.

## Experiment - 7

Date: \_\_\_\_\_  
Page: 7

Aim- Motor Control using servo Motor.

Requirement- Arduino, Bread board, Connecting wires, Servo motor, Blades, RI-32 cable

Theory- Servo Motor works on PWM (Pulse Width Modulation) principle means its angle of rotation is controlled by the duration of applied pulse to its control, PIN. Basically servo motor is made up of a DC motor which is controlled by a variable resistor (potentiometer) and some gears.

High Speed Force of DC motor is converted into torque by gears since work = force  $\times$  distance, in DC motor force is less and distance (speed) is high and in servo, force is high and distance is less. The potentiometer is connected to the output shaft of the servo to calculate the angle and stop the DC motor at the required angle.

// code

```
#include <Servo.h>
```

```
Servo myServo
```

```
int pos = 0;
```

```
void setup() { myServo.attach(9); }
```

```
void loop() { for (pos = 0; pos <= 180; pos++) {  
    myServo.write(pos);  
    delay(30); } }
```

```
for (pos = 180; pos >= 0; pos--) {
```

```
    myServo.write(pos);  
    delay(10); }
```

## Experiment - 8

Aim- Study and use of LED and buzzer control in AI.

Requirement- Arduino, bread board, connecting wires, led bulb (red), buzzer, alarm, RI-32 cable.

Theory- Arduino is an open-source electronics platform-based easy-to-use hardware and software. Arduino boards are able to read inputs - light on sensor, a finger on a button, on a twitter message and turn into an output - activating a motor, turning on a LED, publishing something online.

// Code

```
int ledR = 9;
int alarm = 11;
```

```
void setup() {
    pinMode(alarm, OUTPUT);
    pinMode(ledR, OUTPUT);
}
```

```
void loop() {
    digitalWrite(alarm, HIGH);
    delay(2000);
    digitalWrite(alarm, LOW);
    delay(200);
```

```
    digitalWrite(ledR, HIGH);
    delay(2000);
    digitalWrite(ledR, LOW);
    delay(200);
```

```
}
```

## Experiment - 9

Aim - To study Expert system using Prolog.

Theory - An expert system are the complex computer application developed to solve the complex problems in particular domain at level of extra-ordinary human intelligence.

- \* Prolog is well suited for implementing expert system due to several reasons:-
  - Prolog itself can be regarded as a simple inference engine that derives conclusion from known rule.
  - Very simple expert system can be implemented by relying on prolog's built-in search & backtracking management.
  - Prolog data structure let us flexibly & conveniently represent rule based system.

### \* Characteristics of Expert System :-

- High Performance
- Reliable
- Understandable
- Highly responsive

\* Expert system use Forward & backward chaining if-else rule base & knowledge base to solve problem.

### \* Expert system has following components :-

- (1) Knowledge base
- (2) Inference engine
- (3) User Interface.

### \* Capabilities of expert system :-

- Advising
- Diagnosing
- Demonstrating
- Explaining
- Deriving a solution
- Interpreting result
- Predicting result
- Justifying conclusion

### \* They are incapable of :-

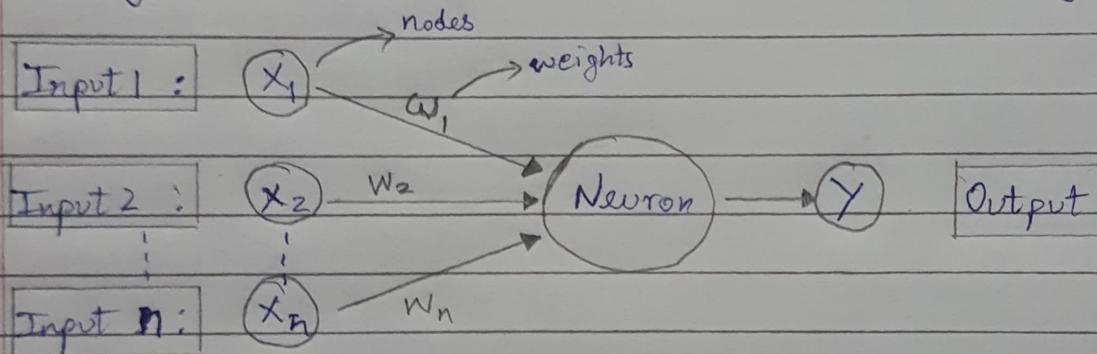
- Substituting human decision
- Refining their own knowledge.

## Experiment - 10

Aim- To study Artificial Neural Network

The term "Artificial Neural Network" is derived from biological neural network that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.

The typical artificial neural network looks something like given figure:



Dendrites from biological neural network represent inputs in artificial neural networks, cell nucleus represents nodes, synapse represents weights and axon represents outputs

## Experiment - 11

**Aim** - Discuss regular expression in detail.

A regular expression (sometimes called a relational expression) is a sequence of characters that define a search pattern, mainly for use in pattern matching with strings, or string matching i.e find-and-replace like operations.

eg → Regular expression for an email address

$$\begin{aligned} & ^{[a-z A-Z 0-9 _ \cdot]} + | @ ([a-z A-Z 0-9 _ \cdot]^{+}) \cdot \\ & ([a-z A-Z] \{2,5\}) \$ \end{aligned}$$

\* Repeaters \* + {}

These symbols act as repeaters & tell computer that the preceding character is to be used for more than just one time.

→ The asterisk symbol \*

Match preceding character (or set of characters) for 0 or more times (upto infinity)

eg → The regular expression ab\*c will give ac, abc, abbc, --- so on

→ The plus symbol +

Repeat preceding character (or set of characters) for atleast 1 or more times (upto infinite)

→ The curly braces {}

Repeat preceding character (or set of characters) for as many times as value inside bracket.

\* Wild card .

The dot symbol can take place of any other symbol that's why it's called wild card character.

### \* Optional Character ?

Preceding character may or may not be present in study to be matched.

### \* The Caret symbol ^

Match must start at begining of string or line

### \* The dollar symbol \$

Match occur at the end of string or before \n at the end of the line or string.

## Experiment -12

Aim - Write program to search a string from given text using re library.

A regular expression ~~def~~ is a sequence of characters that defines a search pattern.

```
# code :- (using re.findall())
```

```
import re
```

```
text = "hello 12 hi 89 . Howdy 34"
```

```
pattern = "\d +"
```

```
result = re.findall(pattern, text)
```

```
print(result) # [12, 89, 34]
```

```
# using re.search()
```

```
import re
```

```
text = "Python is fun"
```

```
# To check if 'Python' is at beginning.
```

```
match = re.search('IA Python', text)
```

```
if match:
```

```
    print("Pattern Found inside the string") # output
```

```
else:
```

```
    print("Pattern not found")
```

## Experiment - 13

Aim - Write program to find contact number, email and date from given text using re.

```
import re
```

```
from datetime import datetime
```

text = """ My name is Mukund Kukreja.

I am currently pursuing GE AIML from Oryan Branga Institute of Technology and Sciences since 8-11-2021.

You can contact me at +91 98765466321 or 1237896540

My email is: abc.def@gmail.com

My Birthday is 20-11-2003 "" "" "

phonePattern = r"[+0-9]\* \d{10}";

phoneNumbers = re.findall(phonePattern, text)

emailPattern = r"[a-zA-Z0-9+-\_.]+@[a-zA-Z0-9.-]+\.\w{1}

[a-zA-Z]\w{2,4}\w{3}"

emails = re.findall(emailPattern, text)

datePattern = r"\d{1,2} - \d{1,2} - \d{2,4}"

dateStrings = re.findall(datePattern, text)

print(phoneNumbers) # ['+91 98765466321', '1237896540']

print(emails) # ['abc.def@gmail.com']

print(dateStrings) # ['8-11-2021', '20-11-2003']