Assignment -1

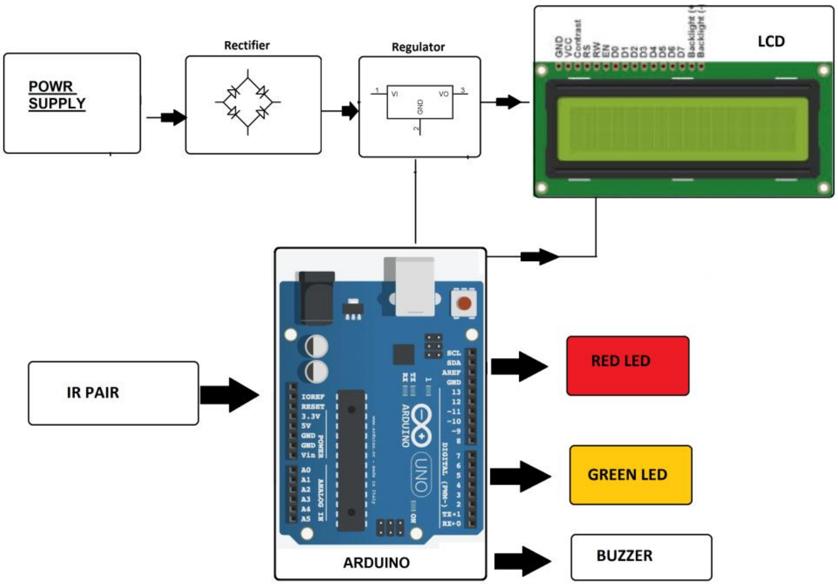
Python Programming

Assignment Date	06 November 2022
Student Name	R.Sowntharya
Student Roll Number	715319106305
Maximum Marks	2 Marks

INTRODUCTION:

Growth in population has led to growth in technology. People use car on large number and number of accidents taking place, is increasing day-by-day. Road accidents are undoubtedly the most frequent happening cases and overall, the cause of the most damage. There are many dangerous roads in the world like mountain roads, narrow curve roads, T roads. Some mountain roads are very narrow and they have many curves. The problems in these curve roads is that the drivers are not able to see the vehicle or obstacles coming from another end of the curve. If the vehicle is in great speed then it is difficult to control and there are chances of falling off a cliff. Hence there is a need of many road safety systems. To avoid these problems in curve roads of mountain areas, we have proposed this vehicle accident prevention system. This accident prevention system using sensors is powered by Arduino board, it consists of IR sensors, LED curve the IR sensor senses the car and LED colour changes to red and raises the buzzer giving signal of danger lights, LCD display and buzzer. When two cars pass from the opposite side of a mountain and then it changes one LED colour into green to allow the one car to pass and then the other LED colour turns green. In this way we can prevent the accidents of curved road.

it is difficult to control and there are chances of falling off a cliff. Hence there is a need of many road safety systems. To avoid these problems in curve roads of mountain areas, we have proposed this vehicle accident prevention system. This accident prevention system using sensors is powered by Arduino board, it consists of IR sensors, LED lights, LCD display and buzzer. When two cars pass from the opposite side of a mountain curve the IR sensor senses the car and LED colour changes to red and raises the buzzer giving signal of danger and then it changes one LED colour into green to allow the one car to pass and then the other LED colour turns green. In this way we



- **Hardware Specifications**

- Arduino

- Crystal Oscillator

- Resistors

- Capacitors

- Transistors

- Cables and Connectors

- Diodes

- PCB and Breadboards
- LED
- Transformer/Adapter
- Push Buttons
- Switch
- IC

- **Designing a Dynamic Assessment Core**
- Recent thrust into machine learning and its different variants
- (supervised, unsupervised, and reinforcement-based) has
- made addressing high processing demands in solving various
- problems possible. The operational requirements highlighted
- in Section 3.4 above position the target assessment core as
- a typical candidate for machine learning considerations. In
- what follows, we showcase an example modeling to the road
- network that can facilitate a robust and dynamic processing.
- Hidden Markov Modelling (HMM) is a powerful statistical
- tool for modelling time-series systems that can be
- characterized to represent probability distributions over a
- sequence of observations. The tool thus lends itself easily to
- the nature of data gathering found in IoT and smart cities
- applications. It further stands as a potential base model for
- several machine learning approaches, including Bayesian or
- Mixture Density Network inferences.
- Our interest herein is in a novel application HMMsto our
- problem. Specifically, a first-order, time-homogeneous, and
- discrete HMM is employed to identify the degree to which
- traversing a certain road link is safe, thereby realizing a safety
- metric.
- The HMM at hand can be formally defined by the fivetuple
- $\Phi = (S, M, P, \delta, \pi)$, where S is an array of links' states;
- M is the emission symbols that characterizes observations
- per each state; P is the states' transition probabilities; δ is the
- emission (or output) probability matrix; and π is the initial
- state distribution array.We assume that the road link state in
- terms of safety, denoted S_t, is hidden fromthe observer. We
- also assume that the current hidden state of a certain road
- link depends only on the preceding state of the same road,

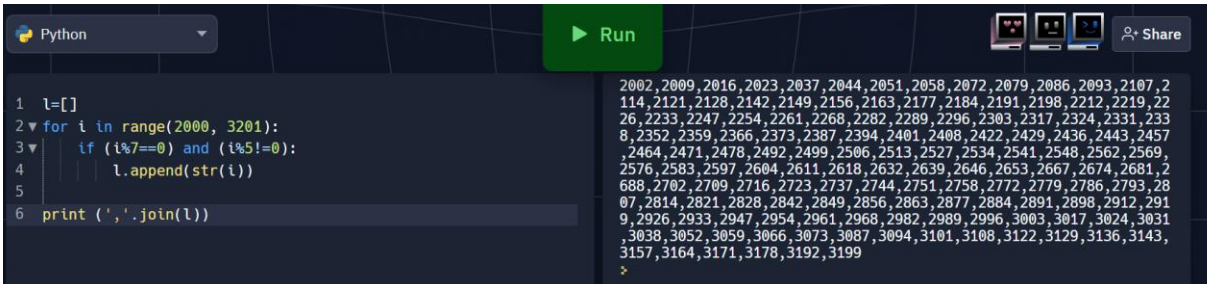
- i.e., that the Markov property is satisfied.
- In what follows we elaborate on the tuple elements.
- **States, *S***: the hidden states of the road link status, which
- describe the safety of the road link. For example, and without
- loss of generality, two-states can be utilized, whether safe or
- not. Further in-between states can be added.
- **Emission Symbols, $\boldsymbol{M}$**: the observations from which
- the hidden states can be deduced. Examples of possible
- observations are road link congestion rate; road condition
- metric; road infrastructure type, e.g., number of road link
- lanes, type side or highway link; or road infrastructure
- characteristic; e.g., road visibility, etc. A metric utilizing a
- combination of two or more of these and other observations
- can also be synthesized.
- For illustration, M can be made to capture road link
- congestion, with $M_i(t)$ = 0, 1, 2, . . . ,$m - 1$, representing m
- levels of congestion. Ifm =4, the congestion can be digitized
- into 4 levels at 0, 25%, 50%, 75%, all relative to the link's
- maximum capacity. If we define the random variable $q_n(t)$ as
- the congestion of road segment, n at time, t, then,$M_n(t)$, as a
- function of road segment congestion, can be defined to be as
- follows.
- $M_n\,(t)$ =
- {{{{{{{{{{{{{{
- 0, 0.25 > $q_n\,(t) \geq 0$
- 1, 0.5 > $q_n\,(t) \geq 0.25$
- 2, 0.75 > $q_n\,(t) \geq 0.5$
- 3, 1 ≥ $q_n\,(t) \geq 0.75$
- (1)
- It is essential to note, however, that the limits of these
- levels would need to be normalized to usefully reflect actual
- links congestion levels.
- **Transition Probabilities, $\boldsymbol{P}$**, is the probability of transition
- among the two states in the HMM.
- **Probability of Emission, $\boldsymbol{\delta}$** (or output probabilities), is
- equal to $P(M_i/S_1)$ and $P(M_i/S_2)$ given the current state is
- S_1 (safe) or S_2 (unsafe) road link, respectively. Extending the
- illustrative example above, δ_i can be calculated if we know
- the probability distribution of $q_n(t)$, an empirical distribution
- interpolated from the sensory data. The empirical distribution
- can be approximated into the best standard distribution,
- if possible, taking into consideration the trade-off between
- accuracy and complexity.
- **Initial State Distribution, $\boldsymbol{\pi}$**, specifies the initial probability
- distribution of the states. While typically initialized
- using a uniform distribution assignment, there are generally
- no assumptions needed regarding prior distribution.
- Three traditional algorithms can be employed to efficiently
- compute an HMM, namely, the Viterbi algorithm,
- the Forward-Backward algorithm, and the Expectation-
- Maximization algorithm [42]. This is particularly advantageous
- in the computation of link safety for our purposes.
- Furthermore, once the HMM is determined, the state of
- the links (as well the state of regions) can be identified and
- utilized using various services, including route planning, as
- will be discussed in the next **Question 1:**

1.Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included). The numbers obtained should be printed in a comma-separated sequence on a single line.

Solution:

```
l=[]
for i in range(2000, 3201):
  if (i%7==0) and (i%5!=0):
l.append(str(i))


print (','.join(l))
#---------------------------------------#
#---------------------------------------#
```



2.With a given integral number n, write a program to generate a dictionary that contains (i, i*i) such that is an integral number between 1 and n (both included). and then the program should print the dictionary.

Suppose the following input is supplied to the program:
8

Then, the output should be:
`{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}`

Solution:

```
n=int(input())
d=dict()
for i in range(1,n+1):
  d[i]=i*i


print d
```

#---------------------------------------#
#---------------------------------------#

Safety-Based Route Planning

Route planning has become widely used in both personal andreliability. Various applications employ efficient algorithmsfor route planning [43]. Trip time and cost, e.g., for tolls,have beenthe typical metrics for route planning applications,but other metrics, however, have been utilized, e.g., for fuel emission/consumption or energy requirements of electric vehicles.Using the dynamic safety assessment proposed above, itis now possible to route vehicles across cities based on asafety. In this manner, drivers can be directed through routesthat minimizes their overall risk in traversing the road network.Meanwhile, enforcement can distribute vehicles acrossdifferent paths to distribute risk of the network and avoidhaving critically unsafe links or routes within the network.It is furthermore possible to target auxiliary mechanismsfor safety-control across.

The network by controlling and redirecting traffic based on user driving behavior or inresponseto incidental changes in the road network.8 Wireless Communications and Mobile ComputingAn advantage of the assessment core proposed aboveis that a routing algorithm can be operated directly onits generated values. In what follows we describe a directapplication for routes assumed to be traversed shortly aftethe route have been computed, and that require a traversaltime sufficiently less than transition time in the HMM. Theseassumptions are without loss in generality and can be relaxedwith easy modifications to the route planning formulation presented below.

Consider a graph (V, E), with V comprising $n + 1$ nodes(vertices), and E comprising the edges in the graph. Nodesrepresent starting, ending, and midway stops for the vehicle,and our interest is in routing a vehicle from a source node (s) to a destination node (d). Vertices are further identifiedby numbers, with vertex 0 identifying the source node and$0 < i < n$ identifying possible target destinations. Nonsource nodes make the set $V* = V-$ {0.The set of edges $E = \{(i, j) : i, j \in V; i < j\}$ represents

The set of $n \cdot (n + 1)/2$ links between the $n + 1$ nodes. Eachedge has an associated traversal cost $cij > 0$, which maybe either symmetric ($cij = cji$) or asymmetric ($cij \neq cij$).