PUNE INSTITUTE OF COMPUTER TECHNOLOGY
DHANKAWADI, PUNE – 43.

LAB MANUAL

ACADEMIC YEAR: 2022-23

DEPARTMENT: COMPUTER ENGINEERING

CLASS: T.E **SEMESTER:** II

SUBJECT: Laboratory Practice II

LAB Expt.No.	PROBLEM STATEMENT
Group A	**All assignments are compulsory**
1.	Implement depth first search algorithm and Breadth First Search algorithm, use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.
2.	Implement A Star Algorithm for any game search problem.
3.	Implement Greedy search algorithm for any one of the following application: I Selection Sort II Minimum Spanning Tree III Single-Source Shortest Path Problem IV Job Scheduling Problem V Prim's Minimal Spanning Tree Algorithm VI Kruskal's Minimal Spanning Tree Algorithm VII Dijkstra's Minimal Spanning Tree Algorithm
Group B	
1.	Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem **or** a graph coloring problem
2.	Develop an elementary catboat for any suitable customer interaction application.
Group C	
1.	Implement any one of the following Expert System I Information management II Hospitals and medical facilities III Help desks management IV Employee performance evaluation V Stock market trading

	VI Airline scheduling and cargo schedules
	Part II :Elective II **Cloud Computing** **(All assignments are compulsory)**
1.	Case study on Microsoft azure to learn about Microsoft Azure is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft-managed data centers. OR Case study on Amazon EC2 and learn about Amazon EC2 web services.
2.	Installation and configure Google App Engine. OR Installation and Configuration of virtualization using KVM.
3.	Creating an Application in SalesForce.com using Apex programming Language.
4.	Design and develop custom Application (Mini Project) using Sales force Cloud.
5.	**Mini-Project** Setup your own cloud for Software as a Service (SaaS) over the existing LAN in your laboratory. In this assignment you have to write your own code for cloud controller using open-source technologies to implement **with HDFS**. Implement the basic operations may be like to divide the file in segments/blocks and upload/ download file on/from cloud in encrypted form.
	Information Security **(Any five)**
1.	Write a Java/C/C++/Python program that contains a string (char pointer) with a value \Hello World'. The program should AND or and XOR each character in this string with 127 and display the result.
2.	Write a Java/C/C++/Python program to perform encryption and decryption using the method of Transposition technique.
3.	Write a Java/C/C++/Python program to implement DES algorithm.
4.	Write a Java/C/C++/Python program to implement AES Algorithm.
5.	Write a Java/C/C++/Python program to implement RSA algorithm.
6.	Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob).

7.	Calculate the message digest of a text using the MD5 algorithm in JAVA.

Augmented and Virtual Reality
(Assignments 1,2, 3,7 are mandatory, any 2 from 4, 5 & 6)

1.	Installation of Unity and Visual Studio, setting up Unity for VR development, understanding documentation of the same.
2.	Demonstration of the working of HTC Vive, Google Daydream or Samsung gear VR.
3.	Develop a scene in Unity that includes: i. A cube, plane and sphere, apply transformations on the 3 game objects. ii. Add a video and audio source.
4.	Develop a scene in Unity that includes a cube, plane and sphere. Create a new material and texture separately for three Game objects. Change the color, material and texture of each Game object separately in the scene. Write a C# program in visual studio to change the color and material/texture of the game objects dynamically on button click.
5.	Develop and deploy a simple marker based AR app in which you have to write a C# program to play video on tracking a particular marker.
6.	Develop and deploy an AR app, implement the following using Vuforia Engine developer portal: i. Plane detection ii. Marker based Tracking (Create a database of objects to be tracked in Vuforia) iii. Object Tracking
7.	**Mini-Projects/ Case Study** Create a multiplayer VR game (battlefield game). The game should keep track of score, no. of chances/lives, levels (created using different scenes), involve interaction, animation and immersive environment. **OR** Create a treasure hunt AR application which should have the following features: i. A help button for instruction box to appear. ii. A series of markers which would give hints on being scanned. iii. Involve interaction, sound, and good UI.

Subject Co-ordinator
Mr. Yogesh Handge

HOCD
Dr. G. V. Kale

ASSINGMENT NO.	1
TITLE	Depth first search algorithm and Breadth First Search algorithm
PROBLEM STATEMENT /DEFINITION	Implement depth first search algorithm and Breadth First Search algorithm, Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.
OBJECTIVE	1. To traverse and search the node in DFS and BFS manner
OUTCOME	The student will be able to 1. Learn depth first search algorithm 2. Learn breadth first search algorithm
S/W PACKAGES AND HARDWARE APPARATUS USED	Hardware- 64 bit Windows OS and Linux Software- C/C++/Java/Python
REFERENCES	1. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", Third edition, Pearson, 2003, ISBN :10: 0136042597 2. Deepak Khemani, "A First Course in Artificial Intelligence", McGraw Hill Education(India), 2013, ISBN : 978-1-25-902998-1 3. Elaine Rich, Kevin Knight and Nair, "Artificial Intelligence", TMH, ISBN-978-0-07-008770-5
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

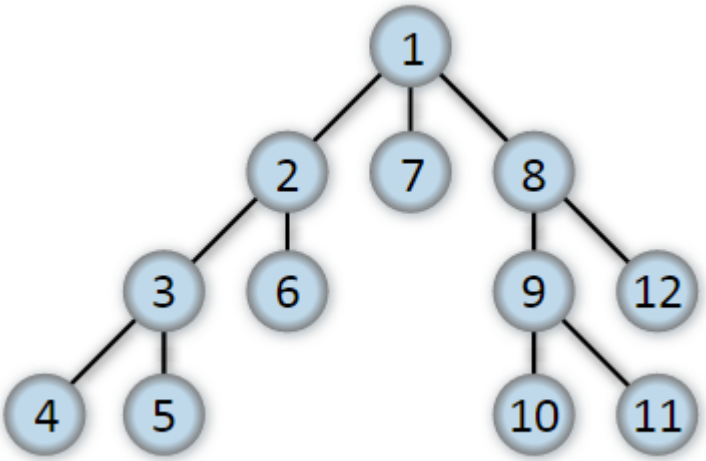
Assignment No:1

Implement depth first search algorithm and Breadth First Search algorithm, Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.

Concepts related Theory:

Depth–first search (DFS) is an algorithm for traversing or searching tree or graph data structures. One starts at the root (selecting some arbitrary node as the root for a graph) and explore as far as possible along each branch before backtracking.

The following graph shows the order in which the nodes are discovered in DFS:



Depth–first search in trees

A tree is an undirected graph in which any two vertices are connected by exactly one path. In other words, any acyclic connected graph is a tree. For a tree, we have the following traversal methods:

Preorder: visit each node before its children.
Postorder: visit each node after its children.
Inorder (for binary trees only): visit left subtree, node, right subtree.
These are already covered in detail in separate posts.

Depth–first search in Graph

A Depth–first search (DFS) is a way of traversing graphs closely related to the preorder traversal of a tree. Following is the recursive implementation of preorder traversal:

```
procedure preorder(treeNode v)
{
   visit(v);
   for each child u of v
      preorder(u);
}
```

To turn this into a graph traversal algorithm, replace “child” with “neighbor”. But to prevent infinite loops, keep track of the vertices that are already discovered and not revisit them.

```
procedure dfs(vertex v)
{
   visit(v);
   for each neighbor u of v
      if u is undiscovered
         call dfs(u);
}
```

Recursive DFS Code in python

```
# A class to represent a graph object
classGraph:
   # Constructor
   def__init__(self,edges,n):
      # A list of lists to represent an adjacency list
      self.adjList=[[]for_inrange(n)]

      # add edges to the undirected graph
      for(src,dest)inedges:
         self.adjList[src].append(dest)
         self.adjList[dest].append(src)


# Function to perform DFS traversal on the graph on a graph
defDFS(graph,v,discovered):

   discovered[v]=True           # mark the current node as discovered
   print(v,end=' ')             # print the current node

   # do for every edge (v, u)
   foruingraph.adjList[v]:
      ifnotdiscovered[u]:       # if `u` is not yet discovered
         DFS(graph,u,discovered)


if__name__=='__main__':
```

```
# List of graph edges as per the above diagram
edges=[
   # Notice that node 0 is unconnected
   (1,2),(1,7),(1,8),(2,3),(2,6),(3,4),
   (3,5),(8,9),(8,12),(9,10),(9,11)
]

# total number of nodes in the graph (labelled from 0 to 12)
n=13

# build a graph from the given edges
graph=Graph(edges,n)

# to keep track of whether a vertex is discovered or not
discovered=[False]*n

# Perform DFS traversal from all undiscovered nodes to
# cover all connected components of a graph
foriinrange(n):
   ifnotdiscovered[i]:
      DFS(graph,i,discovered)
```

Output: 0 1 2 3 4 5 6 7 8 9 10 11 12

BFS algorithm
A standard BFS implementation puts each vertex of the graph into one of two categories:

1. Visited

2. Not Visited

The purpose of the algorithm is to mark each vertex as visited while avoiding cycles.

The algorithm works as follows:

1. Start by putting any one of the graph's vertices at the back of a queue.

2. Take the front item of the queue and add it to the visited list.

3. Create a list of that vertex's adjacent nodes. Add the ones which aren't in the visited list to the back of the queue.

4. Keep repeating steps 2 and 3 until the queue is empty.

The graph might have two different disconnected parts so to make sure that we cover every vertex, we can also run the BFS algorithm on every node

BFS pseudocode

create a queue Q

mark v as visited and put v into Q
while Q is non-empty
remove the head u of Q
mark and enqueue all (unvisited) neighbours of u

```
# BFS algorithm in Python


import collections

# BFS algorithm
defbfs(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 graph[vertex]:
if neighbour notin 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)
```

ASSINGMENT NO.	2
TITLE	A* Algorithm
PROBLEM STATEMENT /DEFINITION	Implement A* algorithm for any game search problem
OBJECTIVE	1. To understand the informed and un informed searching techniques. 2. To analyze A* Algorithm With respect to completeness, optimality, time complexity and space complexity 3. To apply A* algorithm for one of the gaming application.
OUTCOME	The student will be able to 3. Learn how A* algorithm works. 4. Know advantages and disadvantages of Heuristic search based on different parameters.

	5. Apply A* algorithm for various AI Problems
S/W PACKAGES AND HARDWARE APPARATUS USED	Hardware- 64 bit Windows OS and Linux Software- C/C++/Java/Python
REFERENCES	1. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", Third edition, Pearson, 2003, ISBN :10: 0136042597 2. Deepak Khemani, "A First Course in Artificial Intelligence", McGraw Hill Education(India), 2013, ISBN : 978-1-25-902998-1 3. Elaine Rich, Kevin Knight and Nair, "Artificial Intelligence", TMH, ISBN-978-0-07-008770-5
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

Prerequisites:

Concepts related Theory:

Informed search strategy: Informed search uses problem-specific knowledge beyond the definition of the problem itself—can find solutions more efficiently than uninformed

strategy.Informed search algorithm uses the idea of heuristic, so it is also called Heuristic search.

A* is a graph traversal and path search algorithm, which is often used in many fields of computer science due to its completeness, optimality, and optimal efficiency

A* is an informed search algorithm, or a best-first search, meaning that it is formulated in terms of weighted graphs: starting from a specific starting node of a graph, it aims to find a path to the given goal node having the smallest cost

The choice of an appropriate heuristic evaluation function, **h(n)**, is still crucial to the behavior of this algorithm.

In general, we want to choose a heuristic evaluation function **h(n)** which is as close as possible to the *actual* cost of getting to a goal state.

If we can choose a function **h(n)** which never *overestimates* the actual cost of getting to the goal state, then we have a very useful property. Such a **h(n)** is said to be *admissible*.

Best-first search, where the agenda is sorted according to the function **f(n) = g(n) + h(n)** and where the function **h(n)** is admissible, can be proven to always find an optimal solution. This is known as *Algorithm A**.

Calculating heuristics

Heuristics are rules of thumb that may find a solution but are not guaranteed to.Heuristic functions have also been defined as evaluation functions that **estimate** the cost from a node to the goal node. The incorporation of domain knowledge into the search process by means of heuristics is meant to speed up the search process.

Heuristic functions are not guaranteed to be completely accurate.Heuristic values are greater than and equal to zero for all nodes. Heuristic values are seen as an approximate cost of finding a solution. A heuristic value of zero indicates that the state is a goal state.

Heuristic functions are the most common form in which additional knowledge of the problem is imparted to the search algorithm.

Heuristics function: Heuristic is a function which is used in Informed Search, and it finds the most promising path. It takes the current state of the agent as its input and produces the estimation of how close agent is from the goal. The heuristic method, however, might not always give the best solution, but it guaranteed to find a good solution in reasonable time. It is represented by h(n)

G is the movement cost (in number of squares for this game) from the start point A to the current position

H is the estimated movement cost (in number of squares for this game) from the current square to the destination point.The closer the estimated movement cost is to the actual cost, the more accurate the final path will be. If the estimate is off, it is possible the path generated will not be the shortest

Path Sore

We'll give a score ***G*** **+** ***H*** where:

G is the movement cost from the start point A to the current square. So for a square adjacent to the start point A, this would be 1, but this will increase as we get farther away from the start point.

H is the estimated movement cost from the current square to the destination point This is often called the heuristic because we don't really know the cost yet – it's just an estimate.

If you allowed diagonal movement, you might make the movement cost a bit bigger for diagonal moves.

The A* Algorithm

- So now that you know how to compute the score of each square (we'll call this ***F***, which again is equal to ***G*** **+** ***H***), let's see how the A* algorithm works.

- The cat will find the shortest path by repeating the following steps:
- Get the square on the open list which has the lowest score. Let's call this square S.
- Remove S from the open list and add S to the closed list.
- For each square T in S's walkable adjacent tiles:
    - ***If T is in the closed list***: Ignore it.
    - ***If T is not in the open list***: Add it and compute its score.
    - ***If T is already in the open list***: Check if the F score is lower when we use the current generated path to get there. If it is, update its score and update its parent as well.
- Don't worry if you're still a bit confused about how this works – we'll walk through an example so you can see it working step by step! :]

Consider the following route finding problem: To find route from Arad to Bucharest

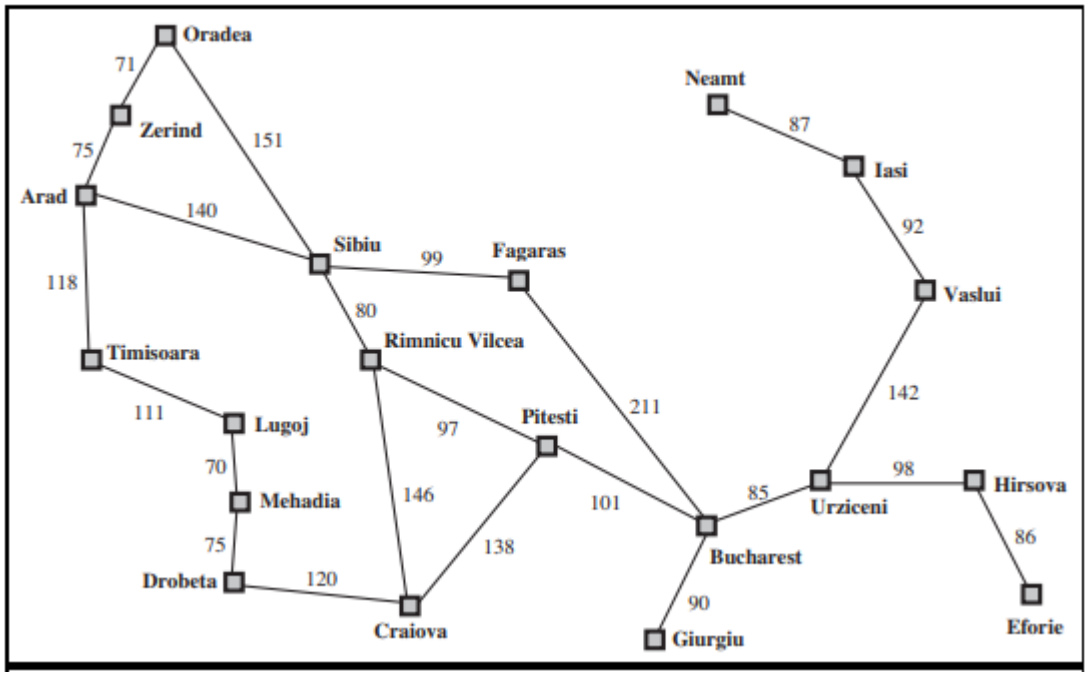


Figure 1: A simplified road map of part of Romania.

The straightline distance is used as heuristic, which can be called h_{SLD} . The goal is Bucharest. Following are straight line distance to Bucharest:

Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

Figure 2: Values of h_{SLD}—straight-line distances to Bucharest.

Example, h_{SLD} (In(Arad)) = 366.
Notice that the values of h_{SLD} cannot be computed from the problem description itself. Moreover, it takes a certain amount of experience to know that h_{SLD} is correlated with actual road distances and is, therefore, a useful heuristic.
*Figure 3*shows the progress of a greedy best-first tree search using hSLD to find a path from Arad to Bucharest.

1. The first node to be expanded from Arad will be Sibiu because it is closer to Bucharest than either Zerind or Timisoara.
2. The next node to be expanded will be Fagaras because it is closest.
3. Fagaras in turn generates Bucharest, which is the goal.

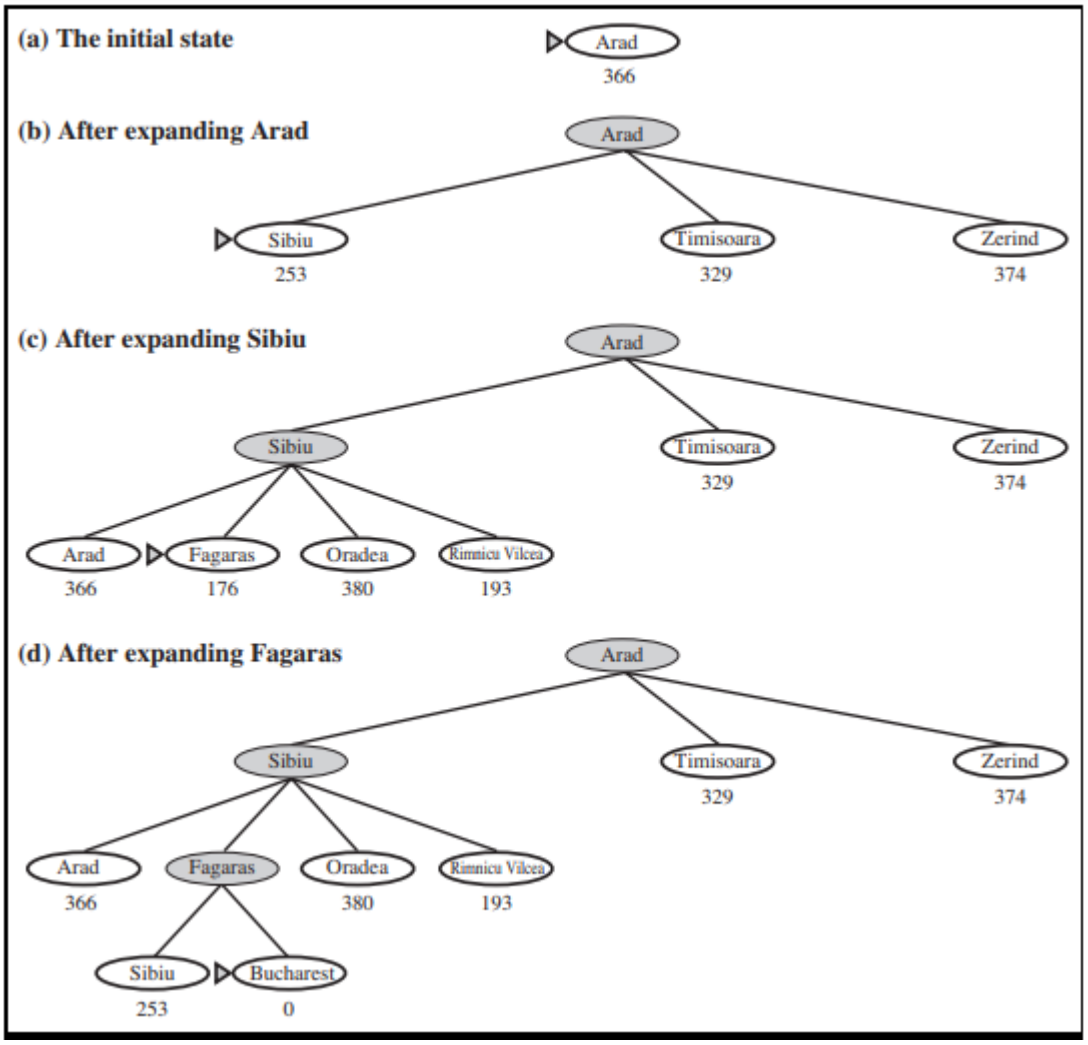


Figure 3: Stages in a greedy best-first tree search for Bucharest with the straight-line distance heuristic hSLD . Nodes are labeled with their h-values.

Analysis

A search algorithm's performance can be analyzed in four ways:

• Completeness: Is the algorithm guaranteed to find a solution when there is one?

• Optimality: Does the strategy find the optimal solution

• Time complexity: How long does it take to find a solution?

• Space complexity: How much memory is needed to perform the search?

Optimality:. Yes

Completeness: Yes (unless there are infinitely many nodes with f ≤ *f(G)*)

Time and Space complexity: Exponential time growth and Keeps all nodes in memory

Conclusion:

Thus A* algorithm was studied and implemented.

Review Questions:

What are informed search techniques?

What are uninformed search techniques?

What is heuristic function?

What is space and time complexity of A* algorithm?

Comment on optimality of of A* algorithm
Is A* complete? Explain.
What do you mean by Admissible heuristics?

ASSINGMENT NO.	3
TITLE	Greedy Search Algorithm
PROBLEM STATEMENT /DEFINITION	Implement Greedy search algorithm for any one of the following application: I Selection Sort II Minimum Spanning Tree

	III Single-Source Shortest Path Problem IV Job Scheduling Problem V Prim's Minimal Spanning Tree Algorithm VI Kruskal's Minimal Spanning Tree Algorithm VII Dijkstra's Minimal Spanning Tree Algorithm
OBJECTIVE	1. To understand the informed search technique- Greedy best first search algorithm 2. To analyze Greedy Search Algorithm according to completeness, optimality, time complexity and space complexity 3. To apply Greedy Best Search algorithm for one of the above mentioned application.
OUTCOME	The student will be able to 1. Learn how search proceeds during a greedy search. 2. Know advantages and disadvantages of greedy search according to analysis parameters. 3. Apply greedy search to solve AI problems.
S/W PACKAGES AND HARDWARE APPARATUS USED	Hardware- 64 bit Windows OS and Linux Software- C/C++/Java/Python
REFERENCES	1. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", Third edition, Pearson, 2003, ISBN :10: 0136042597 2. Deepak Khemani, "A First Course in Artificial Intelligence", McGraw Hill Education(India), 2013, ISBN : 978-1-25-902998-1 3. Elaine Rich, Kevin Knight and Nair, "Artificial Intelligence", TMH, ISBN-978-0-07-008770-5
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

Prerequisites:

Concepts related Theory:

Informed search strategy: Informed search uses problem-specific knowledge beyond the definition of the problem itself—can find solutions more efficiently than uninformed strategy.Informed search algorithm uses the idea of heuristic, so it is also called Heuristic search.

Greedy best-first search

Greedy best-first search algorithm always selects the path which appears best at that moment.. Best-first search is an instance of the general TREE-SEARCH or GRAPH-SEARCH algorithm in which a node is selected for expansion based on an

evaluation function, f(n). The evaluation function is constructed as a cost estimate, so the node with the lowest evaluation is expanded first. The choice of f determines the search strategy. Most best-first algorithms include as a component of f a heuristic function, denoted by h(n):

h(n) = estimated cost of the cheapest path from the state at node n to a goal state.

Heuristic functions are the most common form in which additional knowledge of the problem is imparted to the search algorithm.

Greedy best-first search tries to expand the node that is closest to the goal, on the grounds that this is likely to lead to a solution quickly. Thus, it evaluates nodes by using just the heuristic function; that is, f(n) = h(n).

Heuristics function: Heuristic is a function which is used in Informed Search, and it finds the most promising path. It takes the current state of the agent as its input and produces the estimation of how close agent is from the goal. The heuristic method, however, might not always give the best solution, but it guaranteed to find a good solution in reasonable time. It is represented by h(n)

Consider the following route finding problem: To find route from Arad to Bucharest

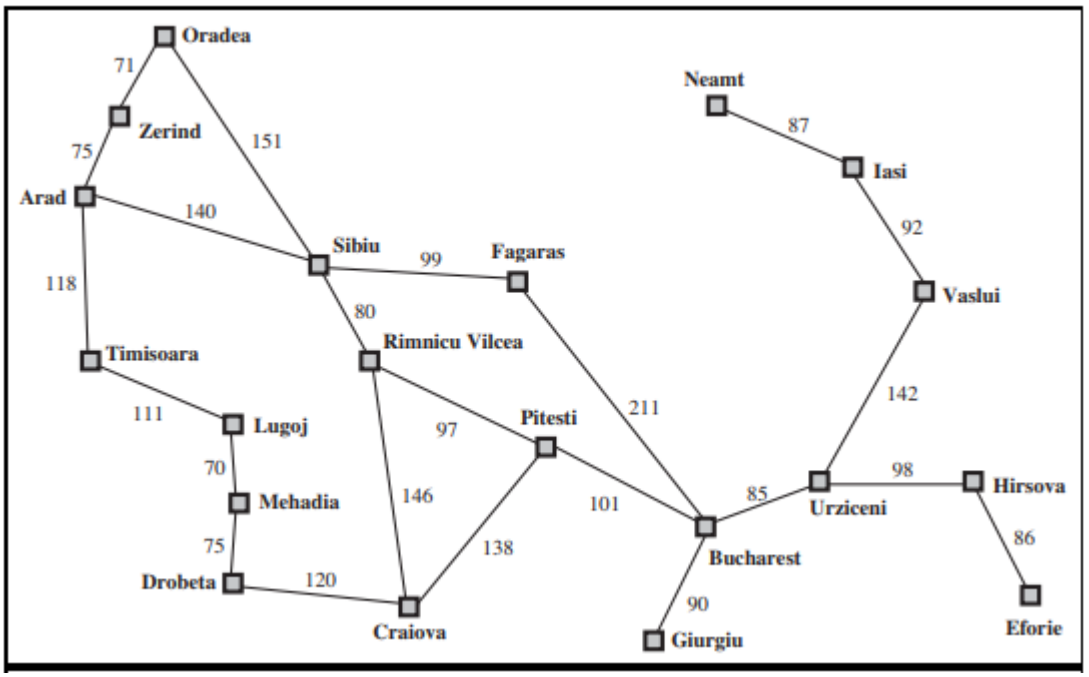


Figure 1: A simplified road map of part of Romania.

The straightline distance is used as heuristic, which can be called h_{SLD} . The goal is Bucharest. Following are straight line distance to Bucharest:

Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

Figure 2: Values of h_{SLD}—straight-line distances to Bucharest.

Example, h_{SLD} (In(Arad)) = 366.
Notice that the values of h_{SLD} cannot be computed from the problem description itself. Moreover, it takes a certain amount of experience to know that h_{SLD} is correlated with actual road distances and is, therefore, a useful heuristic.
*Figure 3*shows the progress of a greedy best-first tree search using hSLD to find a path from Arad to Bucharest.

4. The first node to be expanded from Arad will be Sibiu because it is closer to Bucharest than either Zerind or Timisoara.
5. The next node to be expanded will be Fagaras because it is closest.
6. Fagaras in turn generates Bucharest, which is the goal.

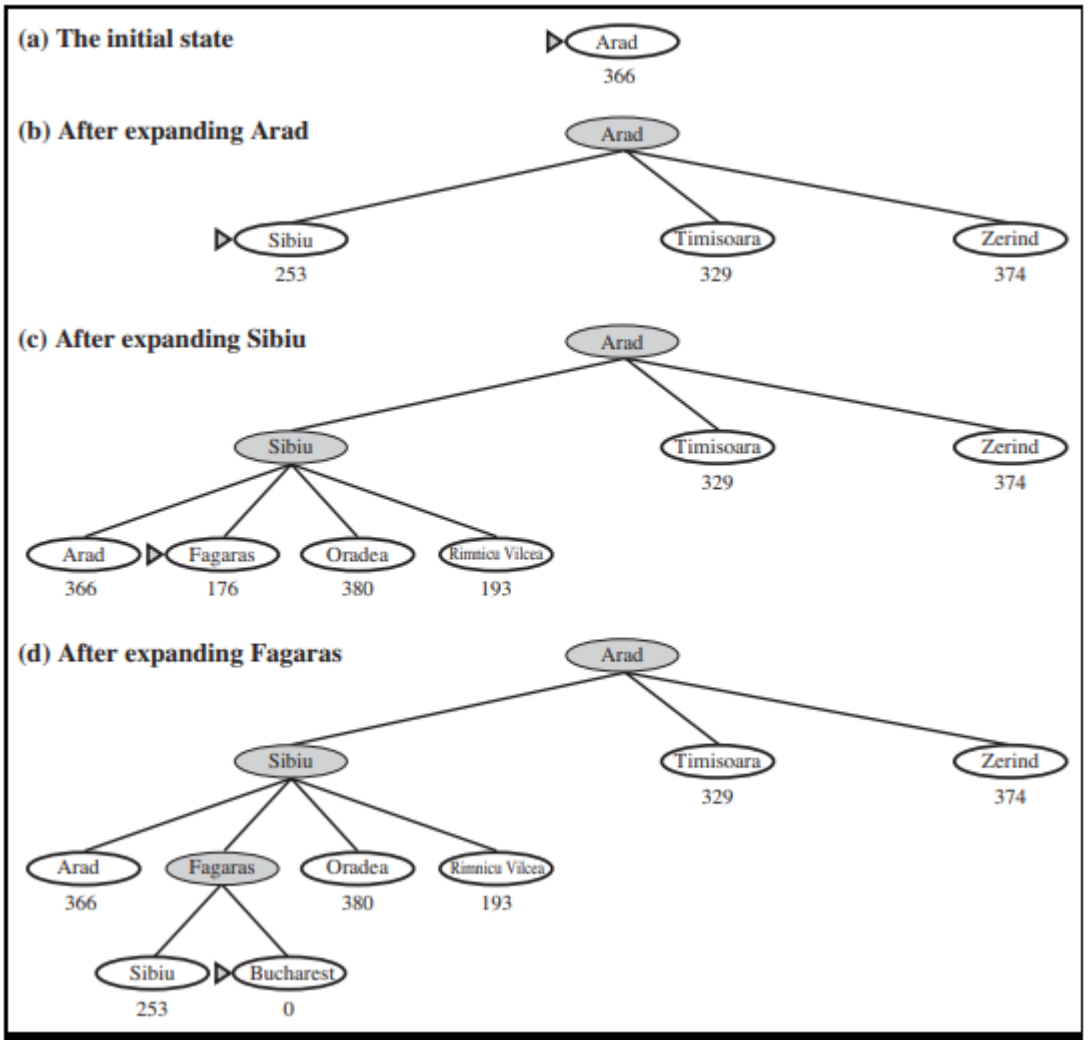


Figure 3: Stages in a greedy best-first tree search for Bucharest with the straight-line distance heuristic hSLD . Nodes are labeled with their h-values.

Analysis

A search algorithm's performance can be analyzed in four ways:

• Completeness: Is the algorithm guaranteed to find a solution when there is one?
• Optimality: Does the strategy find the optimal solution
• Time complexity: How long does it take to find a solution?
• Space complexity: How much memory is needed to perform the search?

Optimality: For this particular problem, greedy best-first search using hSLD finds a solution without ever expanding a node that is not on the solution path; hence, its search cost is minimal. It is not optimal., however: the path via Sibiu and Fagaras to Bucharest is 32 kilometers longer than the path through Rimnicu Vilcea and Pitesti. This shows why the algorithm is called "greedy"—at each step it tries to get as close to the goal as it can. Thus greedy search does not guarantee optimal solution.

Completeness: The greedy search algorithm is complete in finite state space but not in infinite one.

Time and Space complexity: The time and space complexity of greedy search depends on the accuracy of heuristic function.The worst-case time and space complexity is $O(b^m)$, where m is the maximum depth of the search space and b is branching factor.

Algorithm:

Step 1: Place the starting node into the FRONTIER list.

Step 2: If the FRONTIER list is empty, Stop and return failure.

Step 3: Remove the node n, from the FRONTIER list which has the lowest value of h(n), and places it in the EXPLORED list.

Step 4: Expand the node n, and generate the successors of node n.

Step 5: Check each successor of node n, and find whether any node is a goal node or not. If any successor node is goal node, then return success and terminate the search, else proceed to Step 6.

Step 6: For each successor node, algorithm checks for evaluation function h(n), and then check if the node has been in either FRONTIER or EXPLORED list. If the node has not been in both list, then add it to the FRONTIER list.

Step 7: Return to Step 2.

Conclusion:

Thus greedy search algorithm was studied and implemented.

Review Questions:

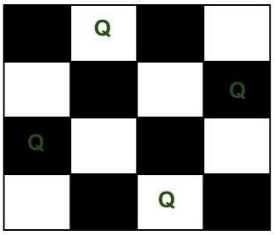
What are informed search techniques?
What are uninformed search techniques?
What is heuristic function?
What is space and time complexity of greedy search?
Comment on optimality of of greedy search.
Is greedy search complete? Explain.

ASSINGMENT NO.	Group B 1
TITLE	Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem.
PROBLEM STATEMENT /DEFINITION	Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem.
OBJECTIVE	To understand Problem Solving using various peculiar search strategies for AI

	To Place N chess queens on an N×N chessboard so that no two queens attack each other.
OUTCOME	To understand Problem Solving using various peculiar search strategies for AI To Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning
S/W PACKAGES AND HARDWARE APPARATUS USED	Operating System recommended: - 64-bit Windows OS and Linux C++/Java/Python
REFERENCES	https://www.geeksforgeeks.org/n-queen-problem-backtracking-3/?ref=gcse http://see.stanford.edu/materials/icspacs106b/H19-RecBacktrackExamples.pdf http://en.literateprograms.org/Eight_queens_puzzle_%28C%29 http://en.wikipedia.org/wiki/Eight_queens_puzzle
STEPS	1) Start in the leftmost column 2) If all queens are placedreturn true 3) Try all rows in the current column. Do following for every tried row. a) If the queen can be placed safely in this row,then mark this [row, column] as part of the solution and recursively check if placing queen here leads to a solution. b) If placing the queen in [row, column] leads toa solution then returns true. c) If placing queen doesn't lead to a solution thenunmark this [row, column] (Backtrack) and go tostep (a) to try other rows. 4) If all rows have been tried and nothing worked, return false to trigger backtracking.
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

Prerequisites: To know about Problem solving skills & Data Structures.
Concepts related Theory:

Let us discuss N Queen as another example problem that can be solved using Backtracking. The N Queen is the problem of placing N chess queens on an N×N chessboard so that no two queens attack each other. For example, following is a solution for 4 Queen problem.



The expected output is a binary matrix which has 1s for the blocks where queens are placed. For example, following is the output matrix for above 4 queen solution.

{0, 1, 0, 0}

{ 0, 0, 0, 1}

{1, 0, 0, 0}

{ 0, 0, 1, 0}

Algorithm:

1) Start in the leftmost column

2) If all queens are placedreturn true

3) Try all rows in the current column.

Do following for every tried row.

a) If the queen can be placed safely in this row, then mark this [row, column] as part of the solution and recursively check if placing queen here leads to a solution.

b) If placing the queen in [row, column] leads toa solution then returns true.

c) If placing queen doesn't lead to a solution thenunmark this [row, column] (Backtrack) and go tostep (a) to try other rows.

4) If all rows have been tried and nothing worked, return false to trigger backtracking.

Conclusion: In this way we have studied the Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem

Review Questions:

1. In how many directions do queens attack each other?
2. Where is the n-queens problem implemented?
3. In n-queen problem, how many values of n does not provide an optimal solution?
4. Which of the following methods can be used to solve n-queen's problem?

5. How many possible solutions exist for an 8-queen problem?

ASSINGMENT NO.	Group B 2
TITLE	Develop an elementary catboat
PROBLEM STATEMENT /DEFINITION	Develop an elementary catboat for any suitable customer interaction application.
OBJECTIVE	These are the computer program you can talk to through messaging apps, chat windows, or voice calling apps. These intelligent digital assistants resolve customer queries in a cost-effective, quick, and consistent manner.
OUTCOME	Facilitate Seamless Live Communication. It Save Time and Money and also Reduce People-to-People Interactions with Customers.
S/W PACKAGES AND HARDWARE APPARATUS USED	Programming Language: python

REFERENCES	What is Artificial Intelligence (AI) and How Does it Work? - Definition from TechTarget Artificial Intelligence (AI) – What it is and why it matters | SAS 1. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", Third edition, Pearson, 2003, ISBN :10: 0136042597 2. Deepak Khemani, "A First Course in Artificial Intelligence", McGraw Hill Education(India), 2013, ISBN : 978-1-25-902998-1
STEPS	To create your own chatbot: 1 Identify your business goals and customer needs. 2 Choose a chatbot builder that you can use on your desired channels. 3 Design your bot conversation flow by using the right nodes. 4 Test your chatbot and collect messages to get more insights. 5 Use data and feedback from customers to train your bot.
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

Prerequisites:
Concepts related Theory:

What is a Chatbot?

For a deeper understanding of Chatbot, we can define it as a computer program that impersonates human conversations in its natural format, which may include text (since the advent of bots) or spoken language using artificial intelligence (AI) techniques such as Natural Language Processing (NLP) and audio analysis. One of the primary aspects of an AI-based bot is that it is dynamic.

AI-based bots learn from the previous interactions and in retrospect, become more intelligent to handle conversations that are more complex.

How do the Chatbots function?

The main technology that lies behind chatbots is NLP and Machine Learning.

When a question is presented to a chatbot, a series or complex algorithms process the received input, understand what the user is asking, and based on that, determines the answer suitable to the question.

Chatbots have to rely on the ability of the algorithms to detect the complexity of both text and spoken words. Some chatbots perform very well to the point it becomes difficult to differentiate whether the user is a machine or a human.

However, handling complex conversations is a huge challenge; where there is a usage of various figures of speech, it may be difficult for machines to understand.

Types of Chatbots

Chatbots are categorized into two different types. Let us look at both and see how they function.

Rule-based chatbots

Chatbots follow a set of established rules or flows to respond to questions posted by a user. All your simple applications contain rule-based chatbots, which respond to queries based on the rules they are trained on. For instance, a weather application, where you ask for weather forecast and it fetches the data from different sources and responds with the information.

Rule-based chatbots may not be able to hold complex conversations. It can only accomplish the tasks it is programmed to perform unless more improvements are made by the developer.

Machine Learning-based chatbots

Chatbots that are based on machine learning can hold more complex conversations as they try to process the question and understand the meaning behind the question. It learns from the previous conversation and enables itself to handle more complex questions in the future.

Now, let’s take a look at some of its use cases.

Use Case CRM

Using chatbots in CRM can be very helpful as it can handle all the mundane tasks, allowing the users to handle other important tasks.

For a sales team, it can help with automating the data entry process, so they can focus more on customer interactions. It has been found that 20 percent of sales personnel efforts are spent in filling out details on the CRM. To address this problem, Fireflies- a bot, fetches or mines data from audio conversations and finds relevant information to be fed to the CRM.

Salesforce has developed a bot, which fetches customer data for the personally talking to the customer on Slack. Though there may be a variety of data present in the database, it only fetches the relevant data to be displayed on Slack of Chatbots

Algorithm: How do you make a chatbot?
To create your own chatbot:

1. Identify your business goals and customer needs.
2. Choose a chatbot builder that you can use on your desired channels.
3. Design your bot conversation flow by using the right nodes.
4. Test your chatbot and collect messages to get more insights.
5. Use data and feedback from customers to train your bot.

Conclusion: Smart solutions are important for the success of any business. From providing 24/7 customer service, improving current marketing activities, saving time spent on engaging with users to improving internal processes, chatbots can yield the much-needed competitive advantage. If you are looking to develop a chatbot, the best thing to do is to approach a company that will understand your business needs to develop a chatbot that helps you achieve your business goals.
Review Questions:
Why Python is used for chatbot?
What are the requirements for developing chatbot?
How do you evaluate a chatbot performance?
How do I improve my chatbot accuracy?

ASSINGMENT NO.	Group C 1
TITLE	Expert System
PROBLEM STATEMENT /DEFINITION	Implement any one of the following Expert System 1. Information management 2. Hospital and medical facilities 3. Help desk management 4. Employee performance evaluation 5. Stock market trading 6. Airline scheduling and cargo schedules
OBJECTIVE	**4. To understand How does an expert system work** 5. To analyze working of expert system with respect to completeness, optimality, time complexity and space complexity **6. To understand What are the components of an expert system?** 7. To understand the working of Inference system
OUTCOME	After execution of this assignment, the student will be able to **1. To understand How does an expert system work** 2. To analyze working of expert system with respect to completeness, optimality, time complexity and space complexity **3. To understand What are the components of an expert system?** 4. To understand the working of Inference system
S/W PACKAGES AND HARDWARE APPARATUS USED	Hardware- 64 bit Windows OS and Linux Software- C/C++/Java/Python
REFERENCES	1. Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Third edition, Pearson, 2003, ISBN :10: 0136042597 2. Deepak Khemani, “A First Course in Artificial Intelligence”, McGraw Hill Education(India), 2013, ISBN : 978-1-25-902998-1 3. Elaine Rich, Kevin Knight and Nair, “Artificial Intelligence”, TMH, ISBN-978-0-07-

	008770-5
STEPS	
INSTRUCTIONS FOR WRITING JOURNAL	Same as other assignments

Prerequisites:
What is and Agent?
Differentiate between Types of Agent?
Agents Environment
Role of Knowledge & experience.

Concepts related Theory:

What is an expert system?

An expert system is a computer program that uses artificial intelligence (AI) technologies to simulate the judgment and behavior of a human or an organization that has expertise and experience in a particular field.

Expert systems are usually intended to complement, not replace, human experts.

How does an expert system work?

Modern expert knowledge systems use machine learning and artificial intelligence to simulate the behavior or judgment of domain experts. These systems can improve their performance over time as they gain more experience, just as humans do.

Expert systems accumulate experience and facts in a knowledge base and integrate them with an inference or rules engine -- a set of rules for applying the knowledge base to situations provided to the program.

The inference engine uses one of two methods for acquiring information from the knowledge base:

1. **Forward chaining** reads and processes a set of facts to make a logical prediction about what will happen next. An example of forward chaining would be making predictions about the movement of the stock market.
2. **Backward chaining** reads and processes a set of facts to reach a logical conclusion about why something happened. An example of backward chaining would be examining a set of symptoms to reach a medical diagnosis.

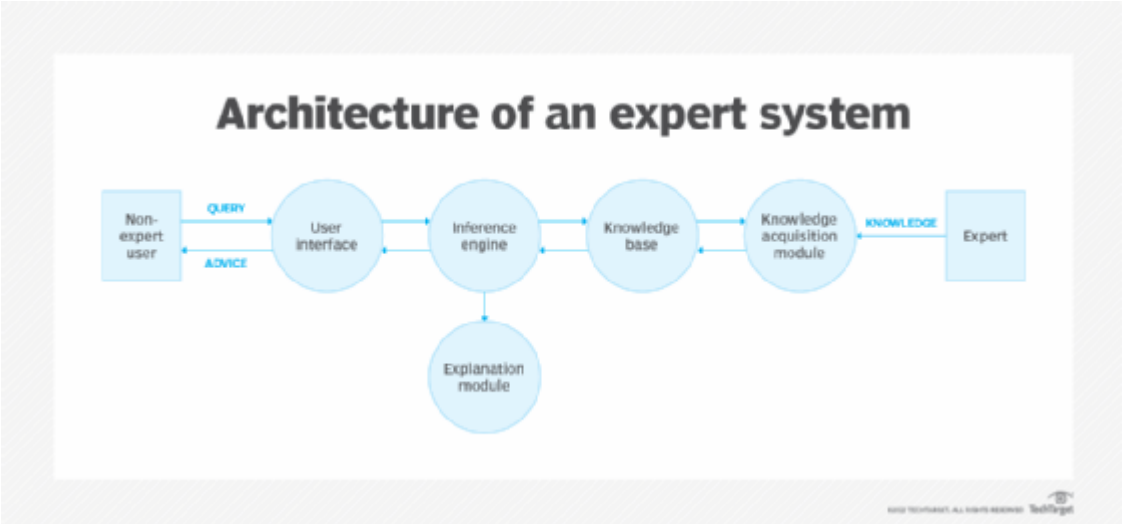
An expert system relies on having a good knowledge base. Experts add information to the knowledge base, and non-experts use the system to solve complex problems that would usually require a human expert.

The process of building and maintaining an expert system is called knowledge engineering. Knowledge engineers ensure that expert systems have all the necessary information to solve a problem. They use various knowledge representation methodologies, such as symbolic patterns, to do this. The system's capabilities can be enhanced by expanding the knowledge base or creating new sets of rules.

What are the components of an expert system?

There are three main components of an expert system:

- **The knowledge base.** This is where the information the expert system draws upon is stored. Human experts provide facts about the expert system's particular domain or subject area are provided that are organized in the knowledge base. The knowledge base often contains a knowledge acquisition module that enables the system to gather knowledge from external sources and store it in the knowledge base.
- **The inference engine.** This part of the system pulls relevant information from the knowledge base to solve a user's problem. It is a rules-based system that maps known information from the knowledge base to a set of rules and makes decisions based on those inputs. Inference engines often include an explanation module that shows users how the system came to its conclusion.
- **The user interface.** This is the part of the expert system that end users interact with to get an answer to their question or problem.



Applications of expert systems

These systems have played a large role in many industries, including the following:

- **Financial services,** where they make decisions about asset management, act as robo-advisors and make predictions about the behavior of various markets and other financial indicators.
- **Mechanical engineering,** where they troubleshoot complex electromechanical machinery.

- **Telecommunications,** where they are used to make decisions about network technologies used and maintenance of existing networks.
- **Healthcare,** where they assist with medical diagnoses.
- **Agriculture,** where they forecast crop damage.
- **Customer service,** where they help schedule orders, route customer requests and solve problems.
- **Transportation,** where they contribute in a range of areas, including pavement conditions, traffic light control, highway design, bus and train scheduling and maintenance, and aviation flight patterns and air traffic control.
- **Law,** where automation is starting to be used to deliver legal services, and to make civil case evaluations and assess product liability.

Advantages of expert systems

Expert systems have several benefits over the use of human experts:

- **Accuracy**. Expert systems are not prone to human error or emotional influence. They make decisions based on defined rules and facts.
- **Permanence.** Human experts eventually leave their role, and a lot of specific knowledge may go with them. Knowledge-based systems provide a permanent repository for knowledge and information.
- **Logical deduction.** Expert systems draw conclusions from existing facts using various types of rules, such as if-then rules.
- **Cost control.** Expert systems are relatively inexpensive compared to the cost of employing human experts. They can help reach decisions more efficiently, which saves time and cuts costs.
- **Multiple experts.** Multiple experts contribute to an expert system's knowledge base. This provides more knowledge to draw from and prevents any one expert from skewing the decision-making.

Challenges of expert systems

Among expert systems' shortcomings are the following:

- **Linear thinking.** Expert systems lack true problem-solving ability. One of the advantages of human intelligence is that it can reason in nonlinear ways and use ancillary information to draw conclusions.
- **Lack of intuition.** Human intuition enables people to use common sense and gut feelings to solve problems. Machines don't have intuition. And emulating gut-feeling decision-making using mechanical logic could take much longer than an expert using intrinsic heuristic knowledge to come to a quick conclusion.
- **Lack of emotion.** In some cases -- medical diagnoses, for example -- human emotion is useful and necessary. For example, the disclosure of sensitive medical information to a patient requires emotional intelligence that an expert system may not have.

- **Points of failure.** Expert systems are only as good as the quality of their knowledge base. If they are supplied with inaccurate information, it can compromise their decisions.

Example Development of Expert System

Here, we will explain the working of an expert system by taking an example of MYCIN ES. Below are some steps to build an MYCIN:

- Firstly, ES should be fed with expert knowledge. In the case of MYCIN, human experts specialized in the medical field of bacterial infection, provide information about the causes, symptoms, and other knowledge in that domain.
- The KB of the MYCIN is updated successfully. In order to test it, the doctor provides a new problem to it. The problem is to identify the presence of the bacteria by inputting the details of a patient, including the symptoms, current condition, and medical history.
- The ES will need a questionnaire to be filled by the patient to know the general information about the patient, such as gender, age, etc.
- Now the system has collected all the information, so it will find the solution for the problem by applying if-then rules using the inference engine and using the facts stored within the KB.
- In the end, it will provide a response to the patient by using the user interface.

Conclusion:

Thus we can develop an Expert system after going through above steps. And it can be analyzed based on its Knowledge base and inference.

Review Questions:

What is an Expert system
What is Knowledge base
What are the Advantages of an expert system
What are the disadvantages or limitations of an expert system
Name the components of expert systems.
What is and Agent?
Differentiate between Types of Agent?
Agents Environment
Role of Knowledge & experience.
How does an expert system works?.

Part II : Elective II Cloud Computing

ASSINGMENT NO.	1
TITLE	Microsoft azure or Amazon EC2
PROBLEM STATEMENT /DEFINITION	Case study on Microsoft azure. Microsoft Azure is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft-managed data centers. OR

	Case study on Amazon EC2 and learn about Amazon EC2 web services.
OBJECTIVE	1. To learn cloud computing environment. 2. To study how to use Microsoft Azure/Amazon EC2
OUTCOME	Understand cloud computing environment
S/W PACKAGES AND HARDWARE APPARATUS USED	Web based plafform:
REFERENCES	Azure: https://azure.microsoft.com/en-in/ AWS: https://aws.amazon.com/ec2/
STEPS	Create the login at the respective website. Use the components of Microsoft Azure and AWS.
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

Theory:

Windows Azure is one of them, which is provided by Microsoft. Azure can be described as the managed data centers that are used to build, deploy, manage the applications and provide services through a global network. The services provided by Microsoft Azure are PaaS and IaaS. Many programming languages and frameworks are supported by it.

Azure as PaaS (Platform as a Service)
As the name suggests, a platform is provided to clients to develop and deploy software. The clients can focus on the application development rather than having to worry about hardware

and infrastructure. It also takes care of most of the operating systems, servers and networking issues.

Pros

- The overall cost is low as the resources are allocated on demand and servers are automatically updated.
- It is less vulnerable as servers are automatically updated and being checked for all known security issues. The whole process is not visible to developer and thus does not pose a risk of data breach.
- Since new versions of development tools are tested by the Azure team, it becomes easy for developers to move on to new tools. This also helps the developers to meet the customer's demand by quickly adapting to new versions.

Cons

- There are portability issues with using PaaS. There can be a different environment at Azure, thus the application might have to be adapted accordingly.

Azure as IaaS (Infrastructure as a Service)
It is a managed compute service that gives complete control of the operating systems and the application platform stack to the application developers. It lets the user to access, manage and monitor the data centers by themselves.

- This is ideal for the application where complete control is required. The virtual machine can be completely adapted to the requirements of the organization or business.
- IaaS facilitates very efficient design time portability. This means application can be migrated to Windows Azure without rework. All the application dependencies such as database can also be migrated to Azure.
- IaaS allows quick transition of services to clouds, which helps the vendors to offer services to their clients easily. This also helps the vendors to expand their business by selling the existing software or services in new markets.

Azure Management Portal
Azure Management Portal is an interface to manage the services and infrastructure launched in 2012. All the services and applications are displayed in it and it lets the user manage them.

started

A free trial account can be created on Azure management portal by visiting the following link - manage.windowsazure.com
The screen that pops up is as shown in the following image. The account can be created using our existing Gmail, Hotmail or Yahoo account.

Compute / Execution Models
This is the interface for executing the application, which is one of the basic functions of Azure.

Data Management

Data management can be done by using SQL server Database component or the simple data storage module offered by Windows Azure. SQL server database can be used for relational database. The storage module can store unrelated tables (without foreign key or any relation) and blobs. Blobs include binary data in the form of images, audio, video, and text files.

Networking

Azure traffic manager routes the requests of a user intelligently to an available datacenter. The process involves finding the nearest datacenter to the user who makes the request for web application, and if the nearest datacenter is not available due to various reasons, the traffic manager deviates the request to another datacenter. However, rules are set by the owner of the application as to how a traffic manager should behave.

The virtual network is another feature that is part of networking in services offered by Windows Azure. The virtual network allows a network between local machines at your premise and virtual machine in Azure Datacenter. IPs to virtual machines can be assigned in a way that makes them appear to be residing in your own premise. The virtual network is set up using a Virtual Private Network (VPN) device.

Software Development Kit (SDK)

Azure applications can be produced by the developers in various programming languages. Microsoft currently provides language-specific SDKs for Java, .NET, PHP, Node.js, Ruby, and Python. There is also a general Windows Azure SDK that supports language, such as C++.

Amazon EC2 (Elastic Compute Cloud) is a web service interface that provides resizable compute capacity in the AWS cloud. It is designed for developers to have complete control over web-scaling and computing resources.

EC2 instances can be resized and the number of instances scaled up or down as per our requirement. These instances can be launched in one or more geographical locations or regions, and **Availability Zones (AZs)**. Each region comprises of several AZs at distinct locations, connected by low latency networks in the same region.

EC2 Components

In AWS EC2, the users must be aware about the EC2 components, their operating systems support, security measures, pricing structures, etc.

ng System Support

Amazon EC2 supports multiple OS in which we need to pay additional licensing fees like: Red Hat Enterprise, SUSE Enterprise and Oracle Enterprise Linux, UNIX, Windows Server, etc. These OS needs to be implemented in conjunction with Amazon Virtual Private Cloud (VPC).

y

Users have complete control over the visibility of their AWS account. In AWS EC2, the security systems allow create groups and place running instances into it as per the requirement. You can specify the groups with which other groups may communicate, as well as the groups with which IP subnets on the Internet may talk.

Pricing

AWS offers a variety of pricing options, depending on the type of resources, types of applications and database. It allows the users to configure their resources and compute the charges accordingly.

Fault tolerance

Amazon EC2 allows the users to access its resources to design fault-tolerant applications. EC2 also comprises geographic regions and isolated locations known as availability zones for fault tolerance and stability. It doesn't share the exact locations of regional data centers for security reasons.

When the users launch an instance, they must select an AMI that's in the same region where the instance will run. Instances are distributed across multiple availability zones to provide continuous services in failures, and Elastic IP (EIPs) addresses are used to quickly map failed instance addresses to concurrent running instances in other zones to avoid delay in services

Migration

This service allows the users to move existing applications into EC2. It costs $80.00 per storage device and $2.49 per hour for data loading. This service suits those users having large amount of data to move.

Features of EC2

Here is a list of some of the prominent features of EC2 −

- **Reliable** − Amazon EC2 offers a highly reliable environment where replacement of instances is rapidly possible. Service Level Agreement commitment is 99.9% availability for each Amazon EC2 region.
- **Designed for Amazon Web Services** − Amazon EC2 works fine with Amazon services like Amazon S3, Amazon RDS, Amazon DynamoDB, and Amazon SQS. It provides a complete solution for computing, query processing, and storage across a wide range of applications.
- **Secure** − Amazon EC2 works in Amazon Virtual Private Cloud to provide a secure and robust network to resources.
- **Flexible Tools** − Amazon EC2 provides the tools for developers and system administrators to build failure applications and isolate themselves from common failure situations.
- **Inexpensive** − Amazon EC2 wants us to pay only for the resources that we use. It includes multiple purchase plans such as On-Demand Instances, Reserved Instances, Spot Instances, etc. which we can choose as per our requirement.

How to Use AWS EC2

Step 1 − Sign-in to AWS account and open IAM console by using the following link https://console.aws.amazon.com/iam/.

Step 2 − In the navigation Panel, create/view groups and follow the instructions.

Step 3 − Create IAM user. Choose users in the navigation pane. Then create new users and add users to the groups.

Step 4 − Create a Virtual Private Cloud using the following instructions.

- Open the Amazon VPC console by using the following link − https://console.aws.amazon.com/vpc/
- Select VPC from the navigation panel. Then select the same region in which we have created key-pair.
- Select start VPC wizard on VPC dashboard.
- Select VPC configuration page and make sure that VPC with single subnet is selected. The choose Select.
- VPC with a single public subnet page will open. Enter the VPC name in the name field and leave other configurations as default.

- Select create VPC, then select Ok.

Step 5 − Create WebServerSG security groups and add rules using the following instructions.

- On the VPC console, select Security groups in the navigation panel.
- Select create security group and fill the required details like group name, name tag, etc.
- Select your VPC ID from the menu. Then select yes, create button.
- Now a group is created. Select the edit option in the inbound rules tab to create rules.

Step 6 − Launch EC2 instance into VPC using the following instructions.

- Open EC2 console by using the following link − https://console.aws.amazon.com/ec2/
- Select launch instance option in the dashboard.
- A new page will open. Choose Instance Type and provide the configuration. Then select Next: Configure Instance Details.
- A new page will open. Select VPC from the network list. Select subnet from the subnet list and leave the other settings as default.
- Click Next until the Tag Instances page appears.

Step 7 − On the Tag Instances page, provide a tag with a name to the instances. Select Next: Configure Security Group.

Step 8 − On the Configure Security Group page, choose the Select an existing security group option. Select the WebServerSG group that we created previously, and then choose Review and Launch.

Step 9 − Check Instance details on Review Instance Launch page then click the Launch button.

Step 10 − A pop up dialog box will open. Select an existing key pair or create a new key pair. Then select the acknowledgement check box and click the Launch Instances button.

ASSINGMENT NO.	2
TITLE	Installation and configure Google App Engine. OR Installation and Configuration of virtualization using KVM.
PROBLEM STATEMENT /DEFINITION	Installation and configure Google App Engine. OR Installation and Configuration of virtualization using KVM.
OBJECTIVE	Learn Installation and configure Google App Engine. OR Installation and Configuration of virtualization using KVM.
OUTCOME	Understand Installation and configure Google App Engine. OR Installation and Configuration of virtualization using KVM.
S/W PACKAGES AND HARDWARE APPARATUS USED	Google App Engine SDK
REFERENCES	https://cloud.google.com/appengine
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition

	4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

Theory:

Installing and Running the Google App Engine On Windows

This document describes the installation of the Google App Engine Software Development Kit (SDK) on a Microsoft Windows and running a simple “hello world” application.

The App Engine SDK allows you to run Google App Engine Applications on your local computer. It simulates the run-time environment of the Google App Engine infrastructure.

Pre-Requisites: Python 2.5.4

If you don't already have Python 2.5.4 installed in your computer, download and Install Python 2.5.4 from:

http://www.python.org/download/releases/2.5.4/

Download and Install

You can download the Google App Engine SDK by going to:

http://code.google.com/appengine/downloads.html

and download the appropriate install package.

Download the Windows installer – the simplest thing is to download it to your Desktop or another folder that you remember.

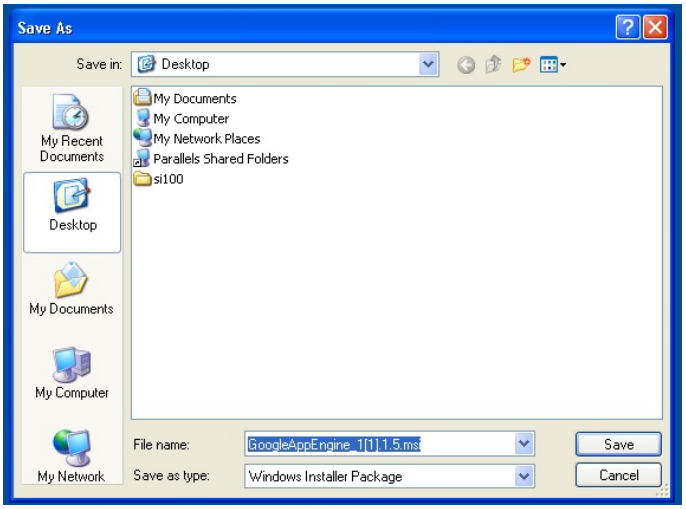
Download the Google App Engine SDK

Before downloading, please read the Terms that govern your use of the App Engine SDK.

Please note: The App Engine SDK is under **active development**, please keep this in mind as you explore its capabilities. See the SDK Release Notes for the information on the most recent changes to the App Engine SDK. If you discover any issues, please feel free to notify us via our Issue Tracker.

Platform	Version	Package	Size	SHA1 Checksum
Windows	1.1.5 - 10/03/08	GoogleAppEngine_1.1.5.msi	2.5 MB	e974312b4aefc0b3873ff0d93eb4c525d5e88c30
Mac OS X	1.1.5 - 10/03/08	GoogleAppEngineLauncher-1.1.5.dmg	3.6 MB	f62208ac01c1b3e39796e58100d5f1b2f052d3e7
Linux/Other Platforms	1.1.5 - 10/03/08	google_appengine_1.1.5.zip	2.6 MB	cbb9ce817bdabf1c4f181d9544864e55ee253de1

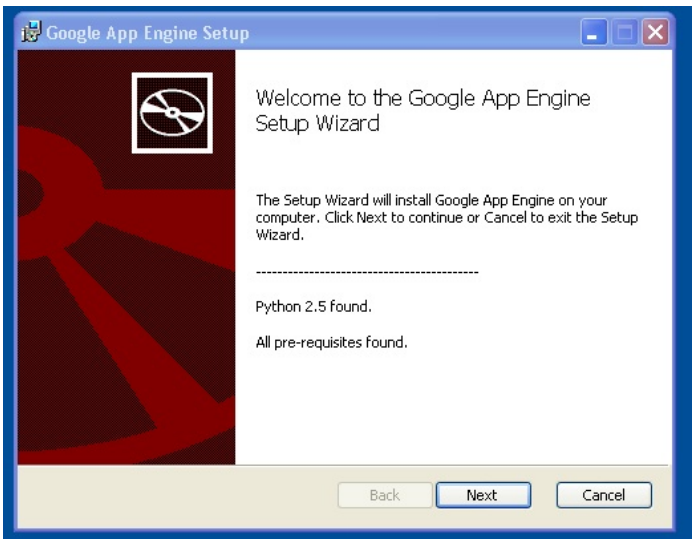
1



Double Click on the **GoogleApplicationEngine**installer.

Click through the installation wizard, and it should install the App Engine. If you do not have Python 2.5, it will install Python 2.5 as well.

Once the install is complete you can discard the downloaded installer



2

Making your First Application

Now you need to create a simple application. We could use the "+" option to have the launcher make us an application – but instead we will do it by hand to get a better sense of what is going on.

Make a folder for your Google App Engine applications. I am going to make the Folder on my Desktop called "**apps**" – the path to this folder is:

C:\Documents and Settings\csev\Desktop\apps
And then make a sub-folder in within **apps** called "**ae-01-trivial**" – the path to this

folder would be:

C:\ Documents and Settings \csev\Desktop\apps\ae-01-trivial
Using a text editor such as JEdit (www.jedit.org), create a file called **app.yaml**in the

ae-01-trivial folder with the following contents:

```
    application: ae-01-trivial
    version: 1
    runtime: python
api_version: 1
    handlers:
    - url: /.*
      script: index.py
```

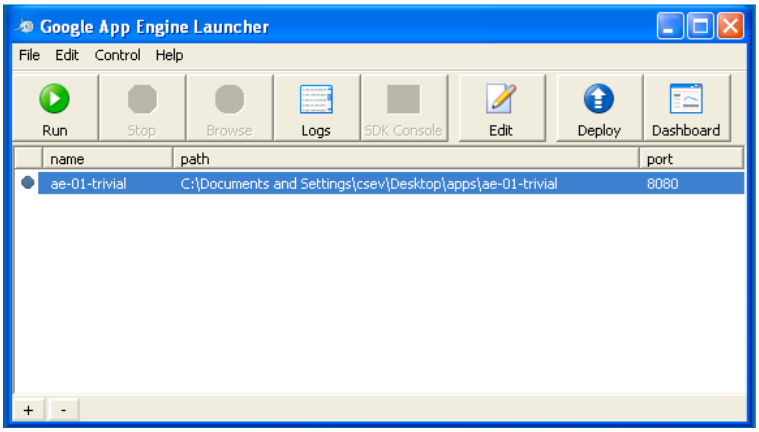
Note: Please do not copy and paste these lines into your text editor – you might end up with strange characters – simply type them into your editor.

Then create a file in the **ae-01-trivial** folder called **index.py** with three lines in it:

```
print 'Content-Type: text/plain'
print ' '
print 'Hello there Chuck'
```

Then start the **GoogleAppEngineLauncher**program that can be found under **Applications**. Use the **File -> Add Existing Application** command and navigate into the **apps** directory and select the **ae-01-trivial** folder. Once you have added the application, select it so that you can control the application using the launcher.

3



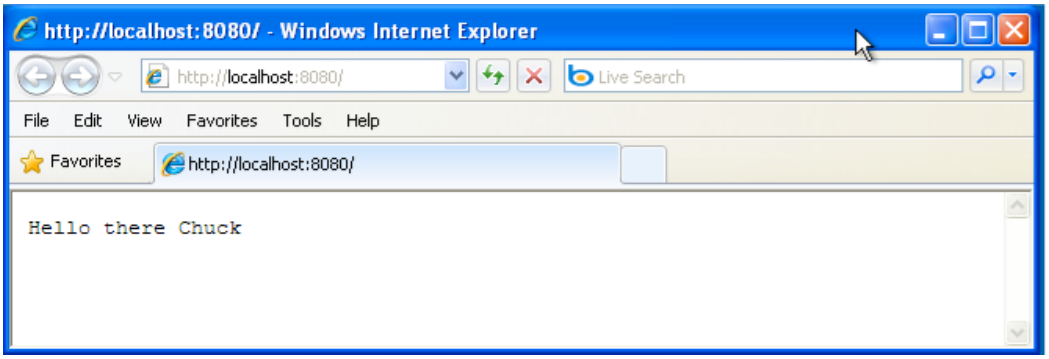
Once you have selected your application and press **Run**. After a few moments your application will start and the launcher will show a little green icon next to your application. Then press **Browse** to open a browser pointing at your application which is running at **http://localhost:8080/**

Paste **http://localhost:8080** into your browser and you should see your application as follows:

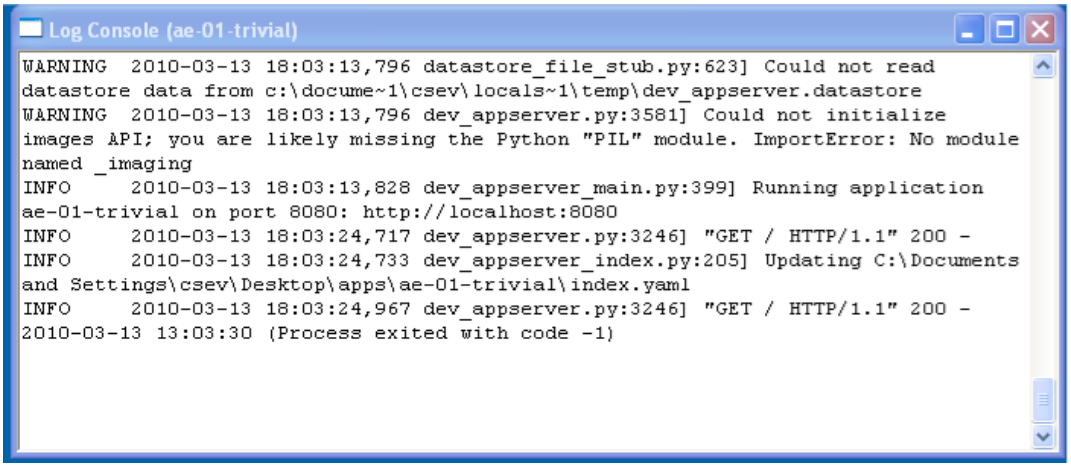
Just for fun, edit the **index.py** to change the name “Chuck” to your own name and press Refresh in the browser to verify your updates.

Watching the Log

You can watch the internal log of the actions that the web server is performing when you are interacting with your application in the browser. Select your application in the Launcher and press the **Logs** button to bring up a log window:



4

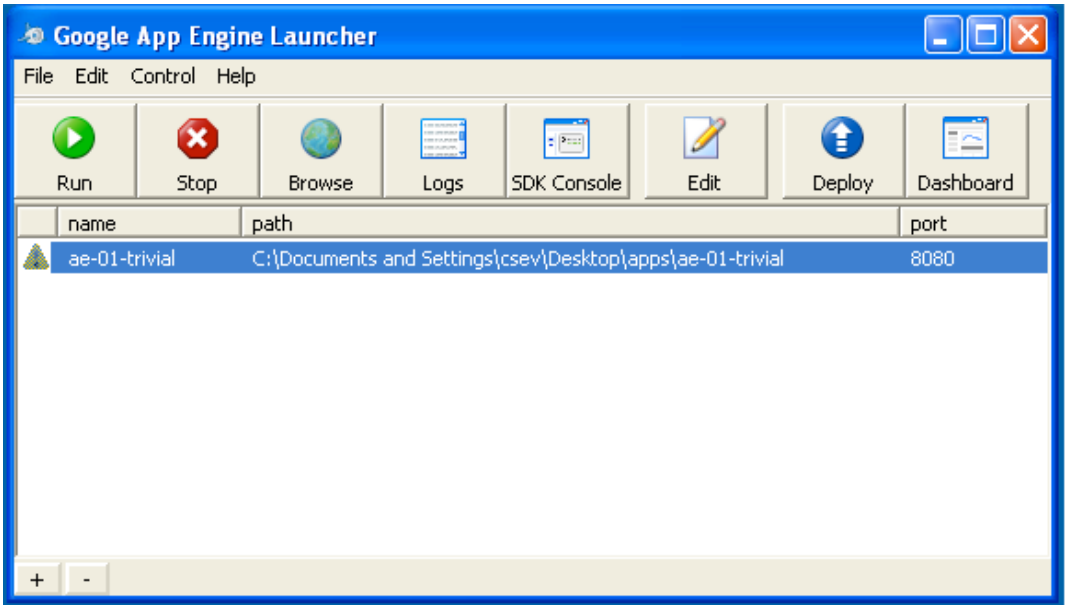


Each time you press **Refresh** in your browser – you can see it retrieving the output with a **GET** request.

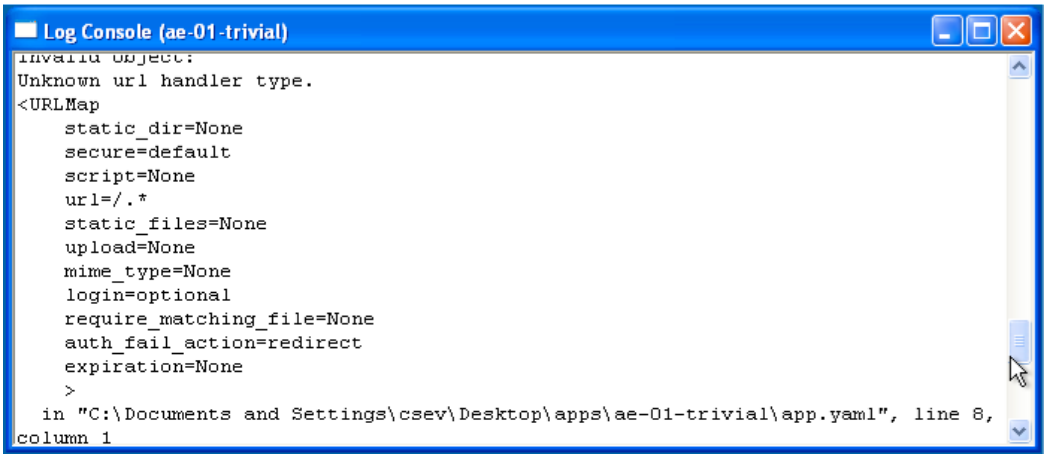
Dealing With Errors

With two files to edit, there are two general categories of errors that you may encounter. If you make a mistake on the **app.yaml**file, the App Engine will not start and your launcher will show a yellow icon near your application:

To get more detail on what is going wrong, take a look at the log for the application:



5



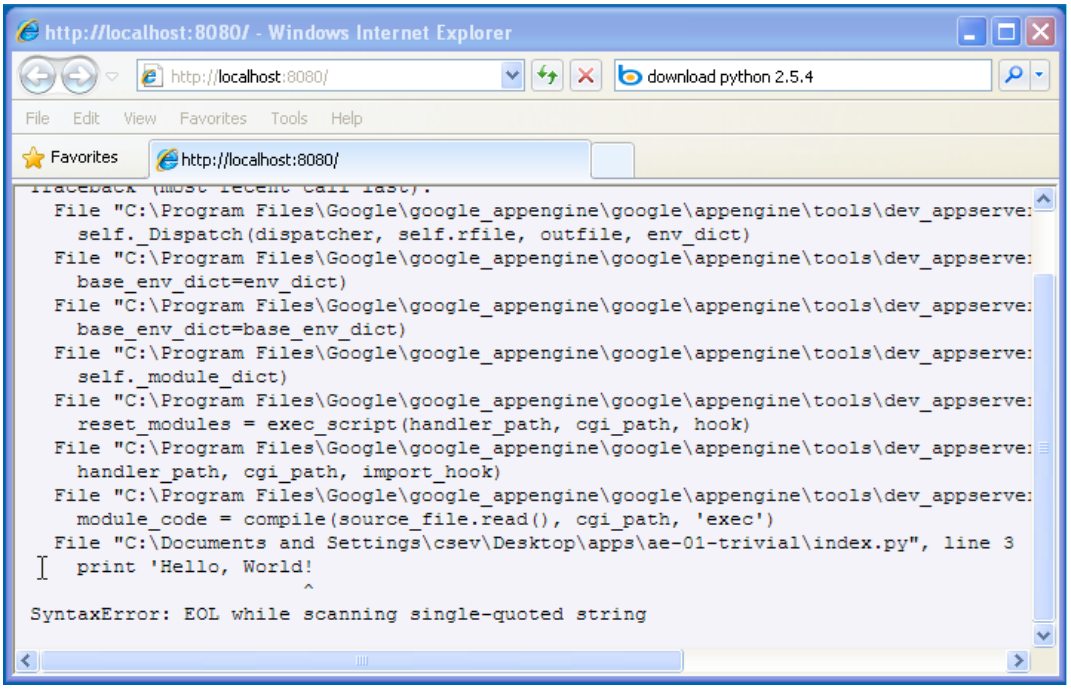
In this instance – the mistake is mis-indenting the last line in the **app.yaml**(line 8). If you make a syntax error in the **index.py** file, a Python trace back error will appear

in your browser.

The error you need to see is likely to be the last few lines of the output – in this case I made a Python syntax error on line one of our one-line application.

Reference: http://en.wikipedia.org/wiki/Stack_trace

When you make a mistake in the **app.yaml**file – you must the fix the mistake and attempt to start the application again.



6

If you make a mistake in a file like i**ndex.py**, you can simply fix the file and press refresh in your browser – there is no need to restart the server.

Shutting Down the Server

To shut down the server, use the Launcher, select your application and press the **Stop** button.

OR

Installation and Configuration of virtualization using KVM.

Linux Kernel based Virtual Machines (KVM) was introduced in Linux kernel version 2.6.20 (Feb 2007) and utilizes hardware virtualization extensions of capable processors. Targeted processors are the Intel VT capable processors and the AMD AMD-V capable processors.

Host virtualization is enabled by KVM and QEMU working together to provide a Linux hypervisor. KVM provides the hardware device abstraction and interface for QEMU while QEMU provides the processor emulation layer. KVM is a Linux kernel module (/lib/modules/*version-number*/kernel/arch/x86/kvm/kvm.ko) that turns Linux into a hypervisor. The guest OS running on KVM is executed in user space thus making each guest instance look like a regular process to the host kernel. Regular process management commands like nice, renice,ps and kill can all operate on the guest VM process. There is also one QEMU/KVM process for each guest OS running on the host system. Look for a process named qemu-system-x86_64.

Libvirt is an API library, a daemon (libvirtd) and a command line tool (virsh).

The BIOS settings also have to be set to enable the processor VM features. For example, on an HP system one enters BIOS settings (ESC on boot) + Security and set the following:

- Virtualization Technology (VT-x): Enable
- Intel VT for Directed I/O (VT-d): Enable
  (reporting I/O device assignment to VMM through DMAR ACPI tables)
- Intel TXT(TL) Support: Disable
  (Trusted Execution Technology support)
- Save changes and Exit

Check to see if your processor is KVM capable: egrep '(vmx|svm)' --color=always /proc/cpuinfo

KVM Installation:

- Red Hat/CentOS: Use the "Add/Remove Programs" GUI: System + Administration + Add/Remove Programs + Virtualization
  (View program categories: yum grouplist | grep -ivirt)
  or yum install kvmvirt-manager libvirt
- Ubuntu 16.04/Debian: sudo apt-get install qemu-kvmlibvirt-bin virt-manager

Verify installation is copacetic: virt-host-validate

```
QEMU: Checking for hardware virtualization                    : PASS
QEMU: Checking if device /dev/kvm exists                      : PASS
QEMU: Checking if device /dev/kvm is accessible               : PASS
QEMU: Checking if device /dev/vhost-net exists                : PASS
QEMU: Checking if device /dev/net/tun exists                  : PASS
QEMU: Checking for cgroup 'memory' controller support         : PASS
QEMU: Checking for cgroup 'memory' controller mount-point     : PASS
...
...
```

KVM Configuration:

- Turn on network packet forwarding:
  Edit file: /etc/sysctl.conf
  Set: net.ipv4.ip_forward = 1
  (Default value)
- Guest VMs location: /var/lib/libvirt/...
- Red Hat/CentOS 6: Load VMs upon system boot: chkconfiglibvirtd on
  (init script: /etc/init.d/libvirtd)
  SeLinux may be an issue and can be turned off: setenforce 0
- Ubuntu init scripts:
  - /etc/init.d/libvirt-bin
  - /etc/init.d/libvirt-guests

Guest OS VM Installation:

List supported guest operating systems (as key word attribute and description. eg centos7.0, fedora22, rhel7.0, win7, etc):

- RHEL/CentOS 6: virt-install --os-variant=list
- Ubuntu 16.04: osinfo-query os
  (Install: sudo apt install libosinfo-bin)

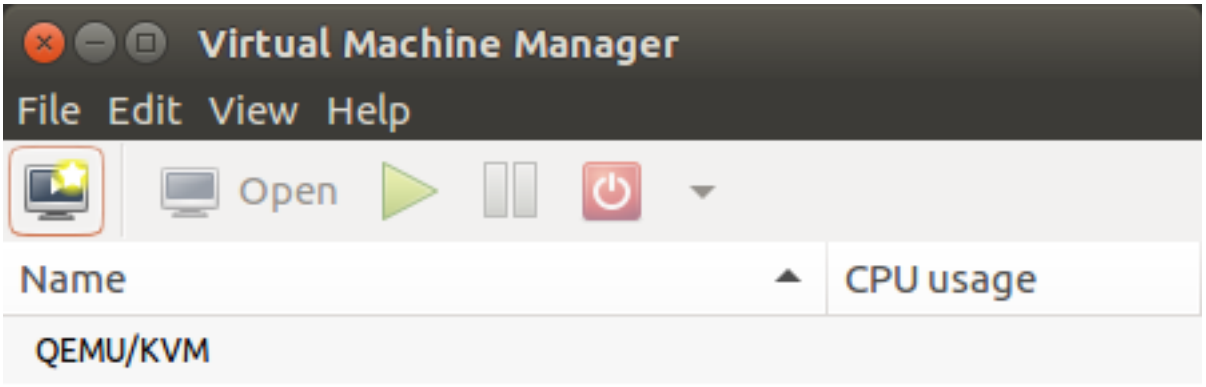
Example installation of a downloaded Linux **ISO CentOS-7-x86_64-Minimal-1708.iso**, installed to a new guest VM instance with 2GB ram and 4GB disk on an Ubuntu 16.04 host.

Command-line installation:

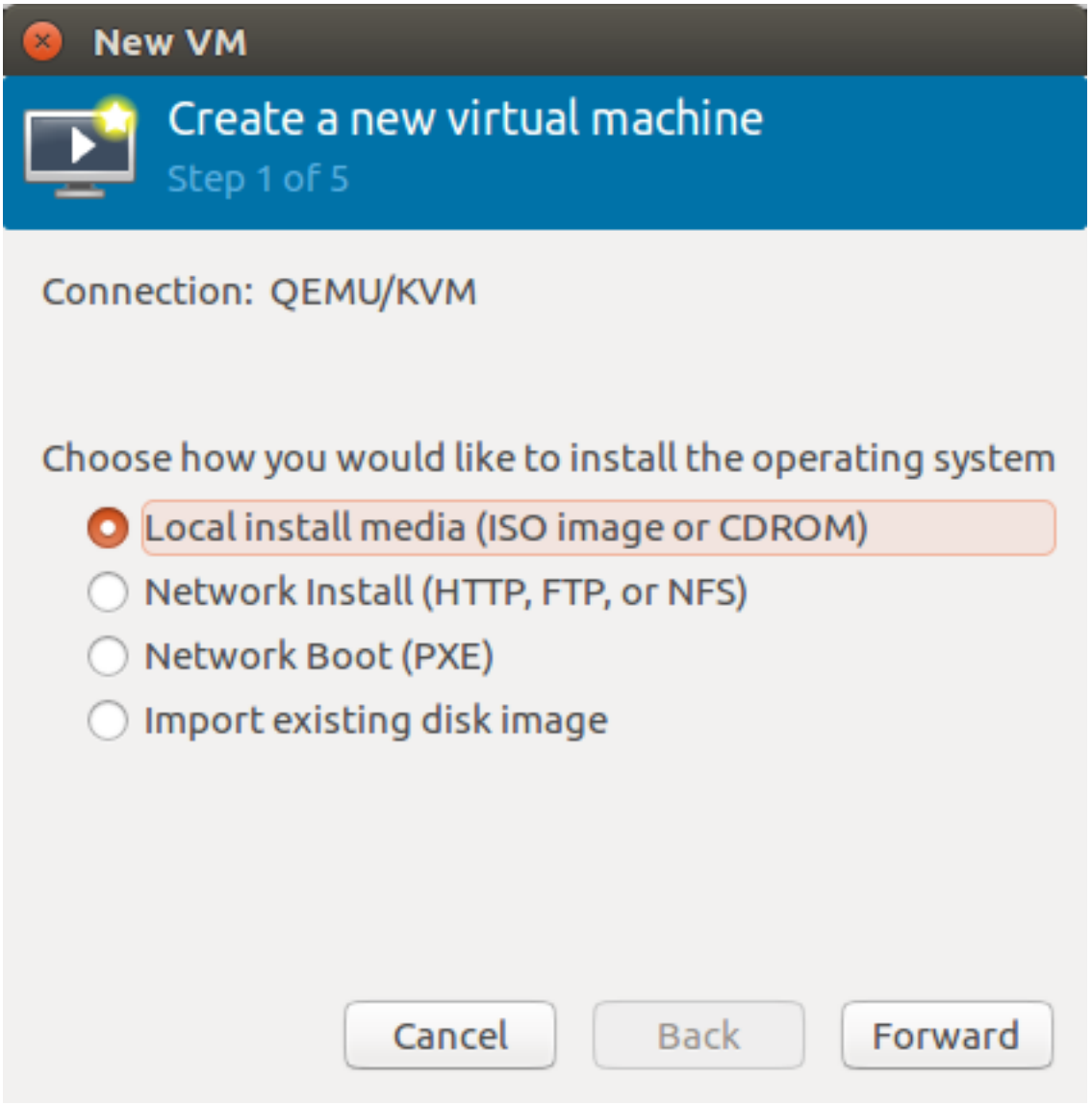
```
sudovirt-install --os-variant=rhel6 --network bridge=br0  \
        --disk /var/lib/libvirt/images/guestos1.img,size=8 \
        --disk path=/iso/CentOS-7-x86_64-Minimal-1708.iso,media=cdrom \
        --graphics none --vcpus=1 --ram=2048 --name=guestos1
```

GUI Install:

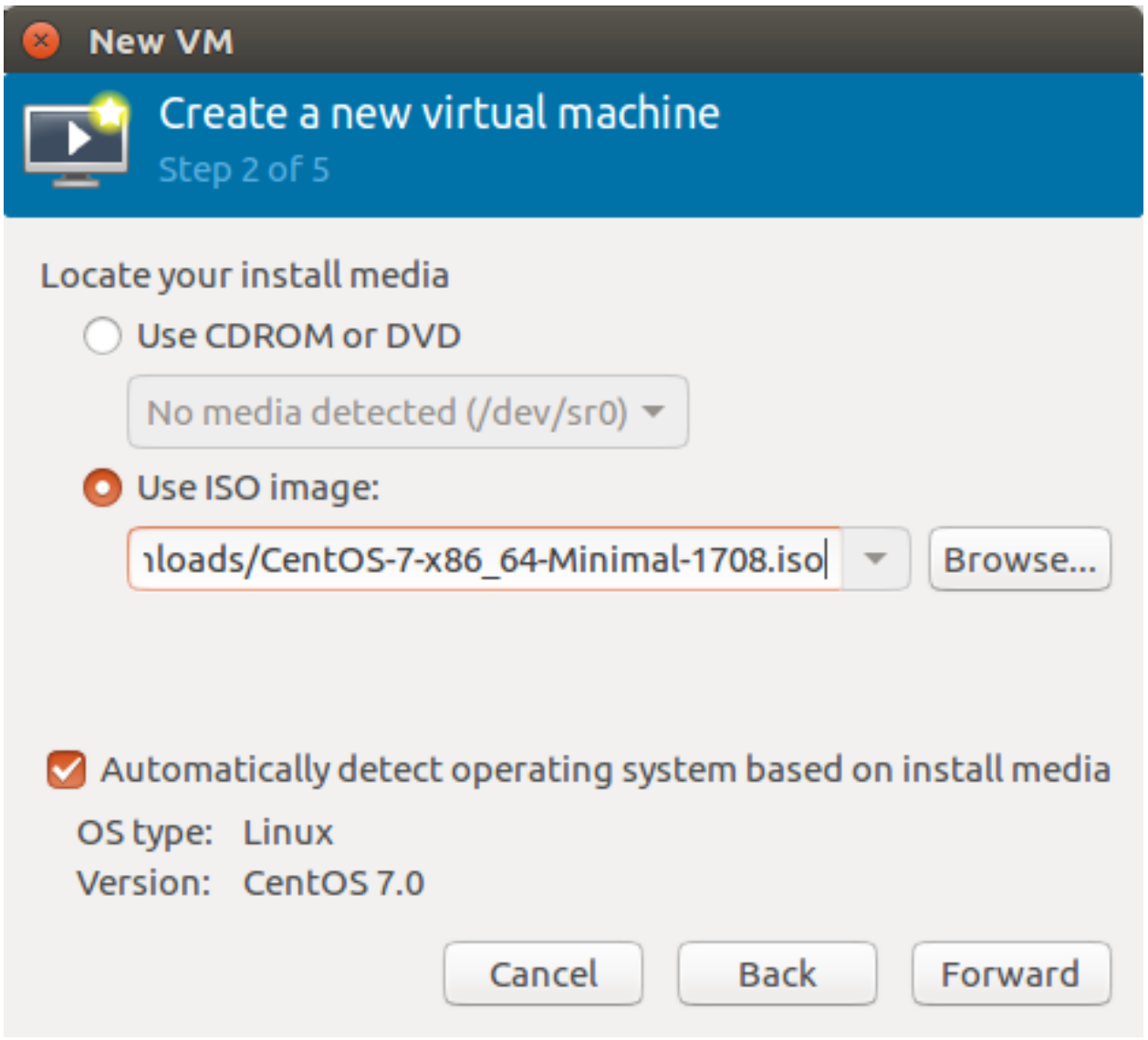
sudovirt-manager



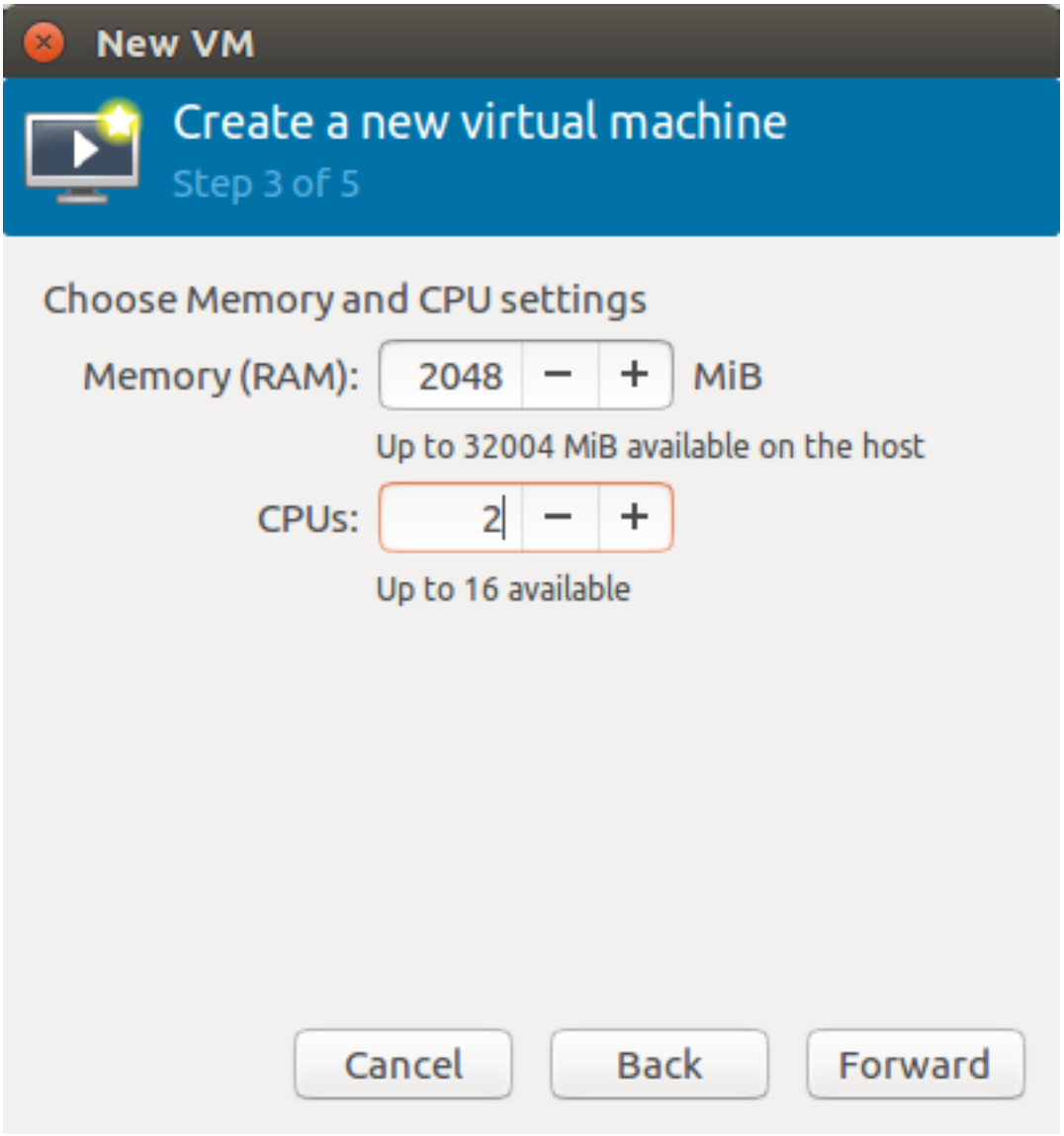
Right click + New


New VM
Create a new virtual machine
Step 1 of 5
Connection: QEMU/KVM
Choose how you would like to install the operating system
Local install media (ISO image or CDROM)
Network Install (HTTP, FTP, or NFS)
Network Boot (PXE)
Import existing disk image
Cancel
Back
Forward

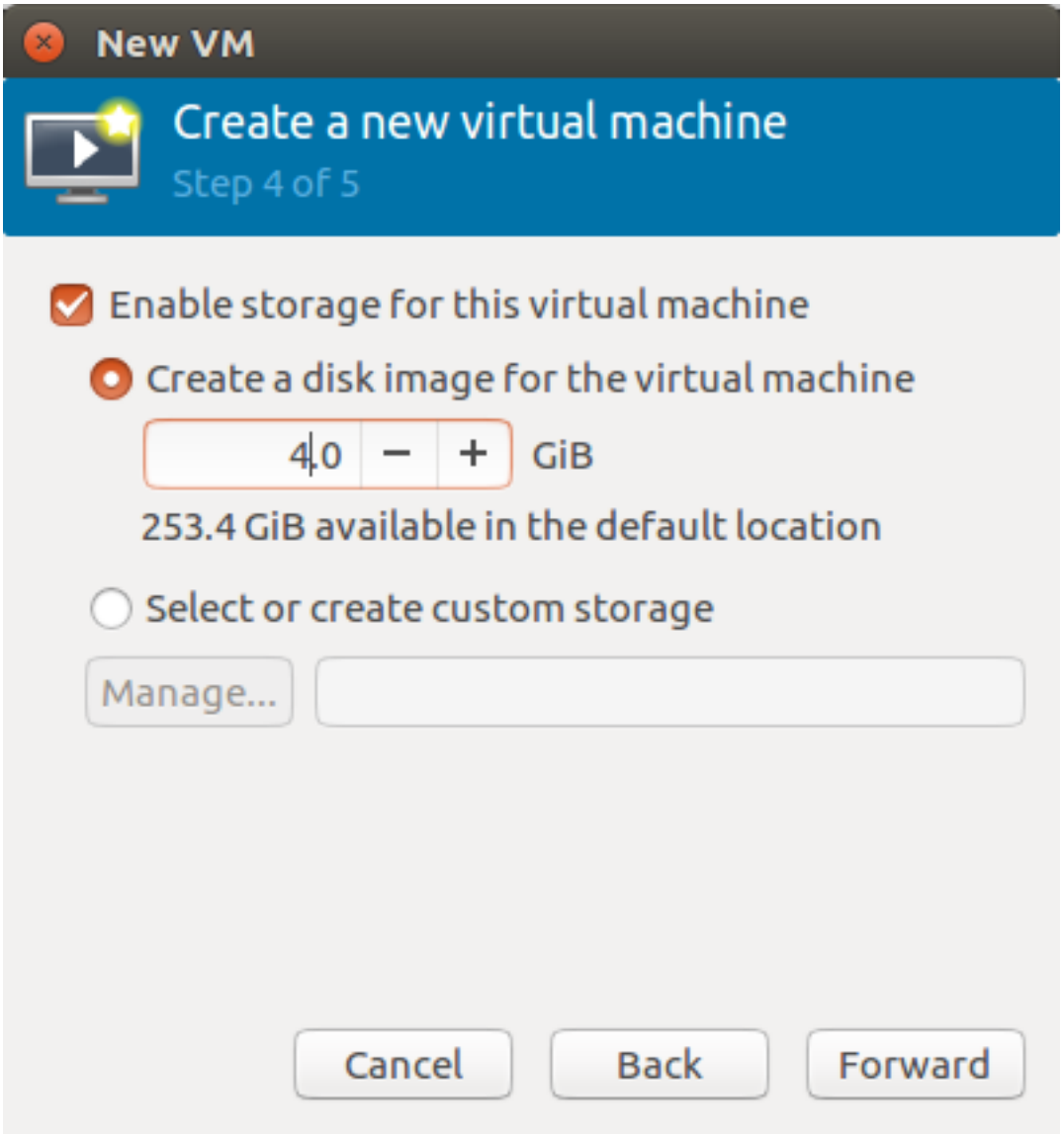
Forward


New VM
Create a new virtual machine
Step 2 of 5
Locate your install media
Use CDROM or DVD
No media detected (/dev/sr0)
Use ISO image:
ıloads/CentOS-7-x86_64-Minimal-1708.iso
Browse...
Automatically detect operating system based on install media
OS type: Linux
Version: CentOS 7.0
Cancel
Back
Forward

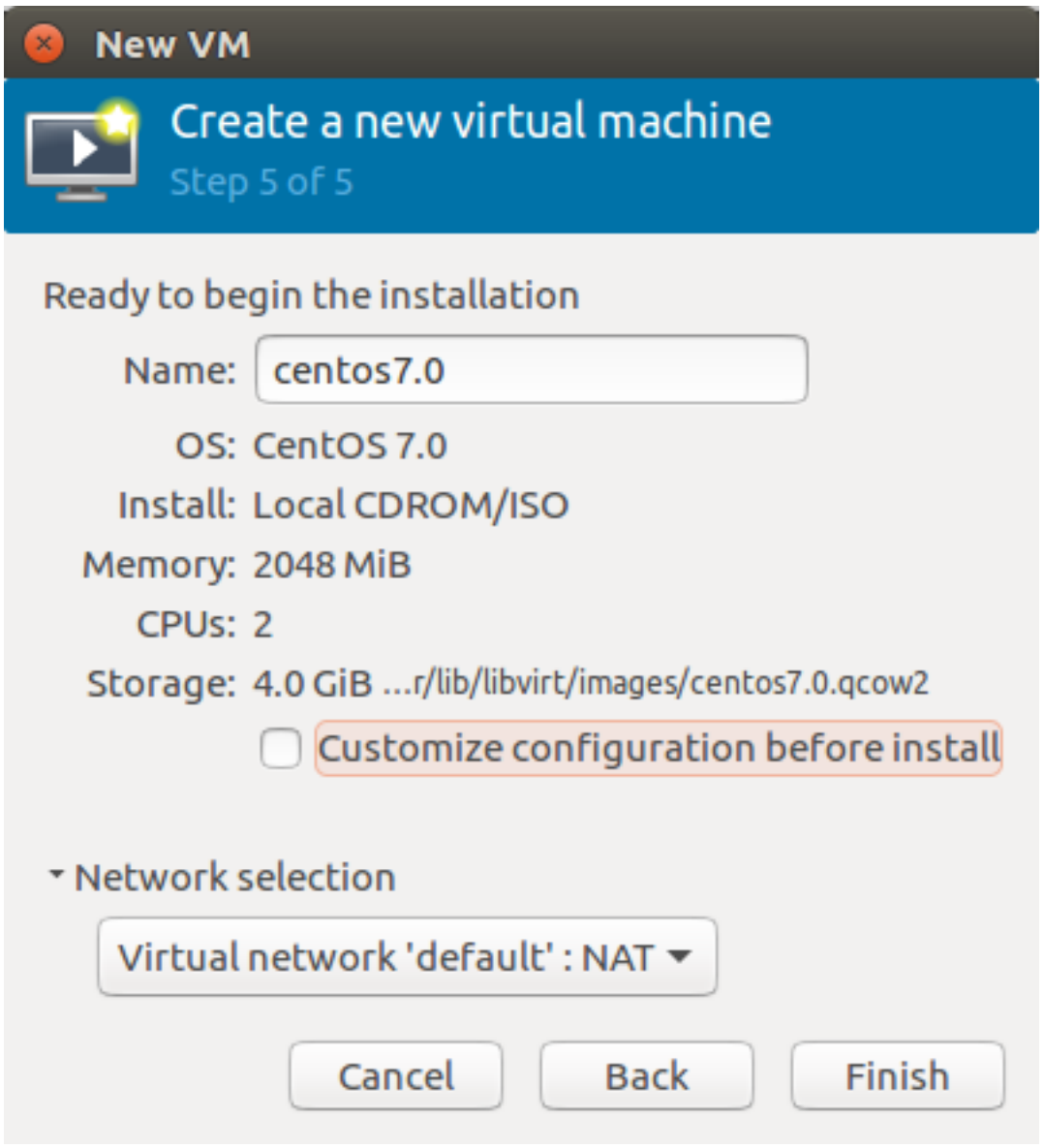
Forward


New VM
Create a new virtual machine
Step 3 of 5
Choose Memory and CPU settings
Memory (RAM):
2048
MiB
Up to 32004 MiB available on the host
CPUs:
2
Up to 16 available
Cancel
Back
Forward

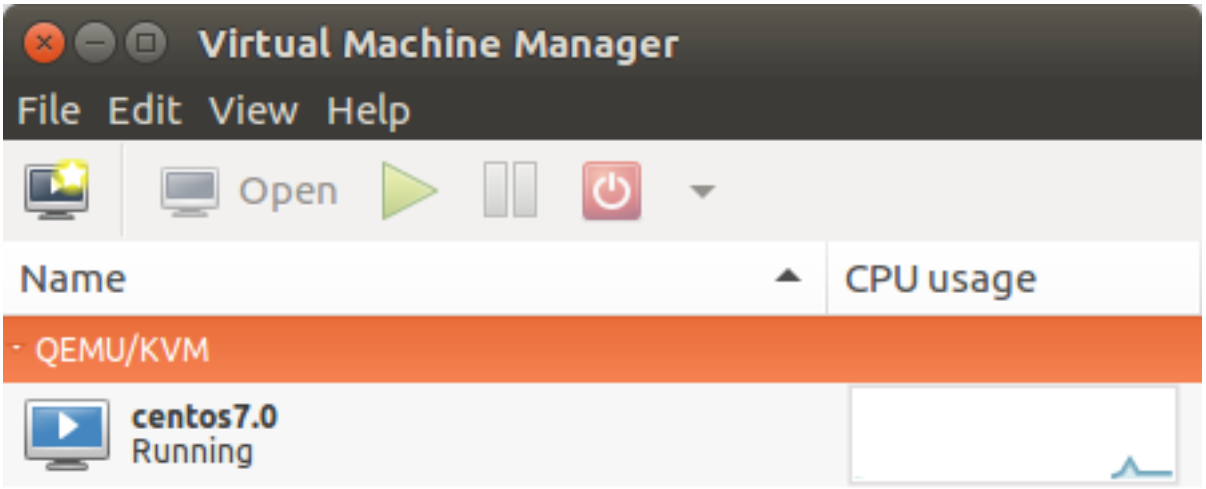
Forward


New VM
Create a new virtual machine
Step 4 of 5
Enable storage for this virtual machine
Create a disk image for the virtual machine
40
GiB
253.4 GiB available in the default location
Select or create custom storage
Manage...
Cancel
Back
Forward

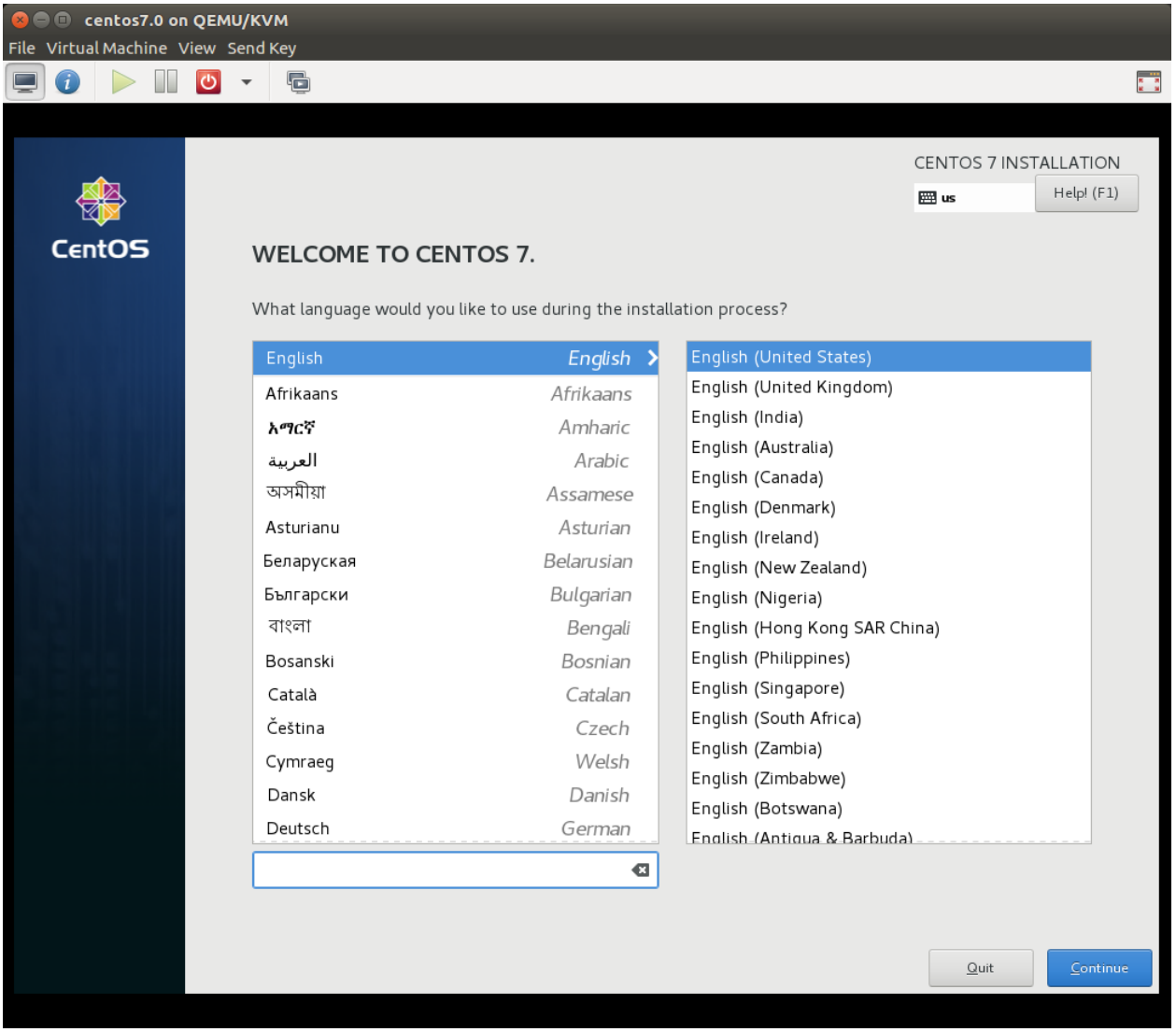
Forward

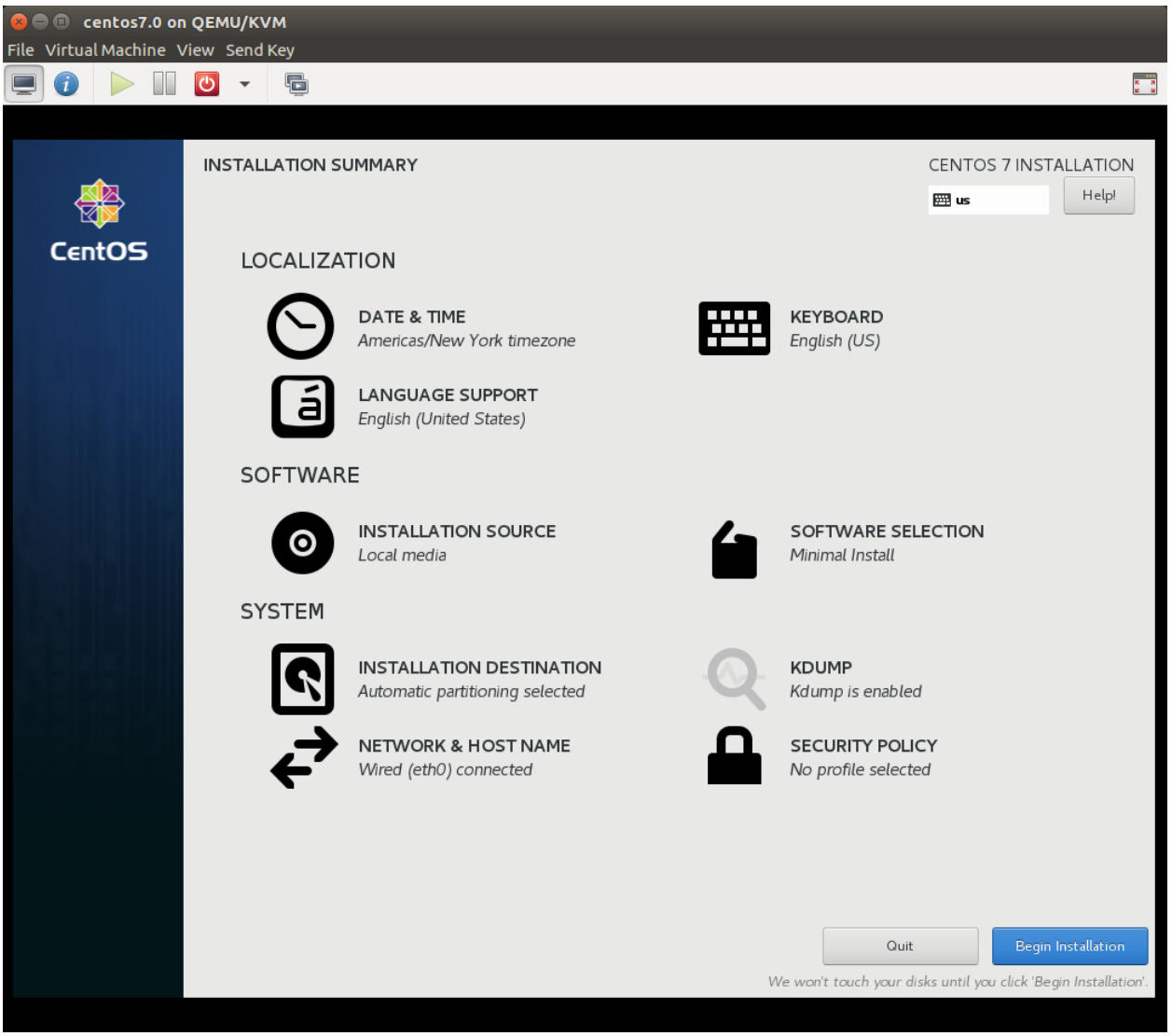

New VM
Create a new virtual machine
Step 5 of 5
Ready to begin the installation
Name: centos7.0
OS: CentOS 7.0
Install: Local CDROM/ISO
Memory: 2048 MiB
CPUs: 2
Storage: 4.0 GiB ...r/lib/libvirt/images/centos7.0.qcow2
Customize configuration before install
Network selection
Virtual network 'default' : NAT
Cancel
Back
Finish

Finish

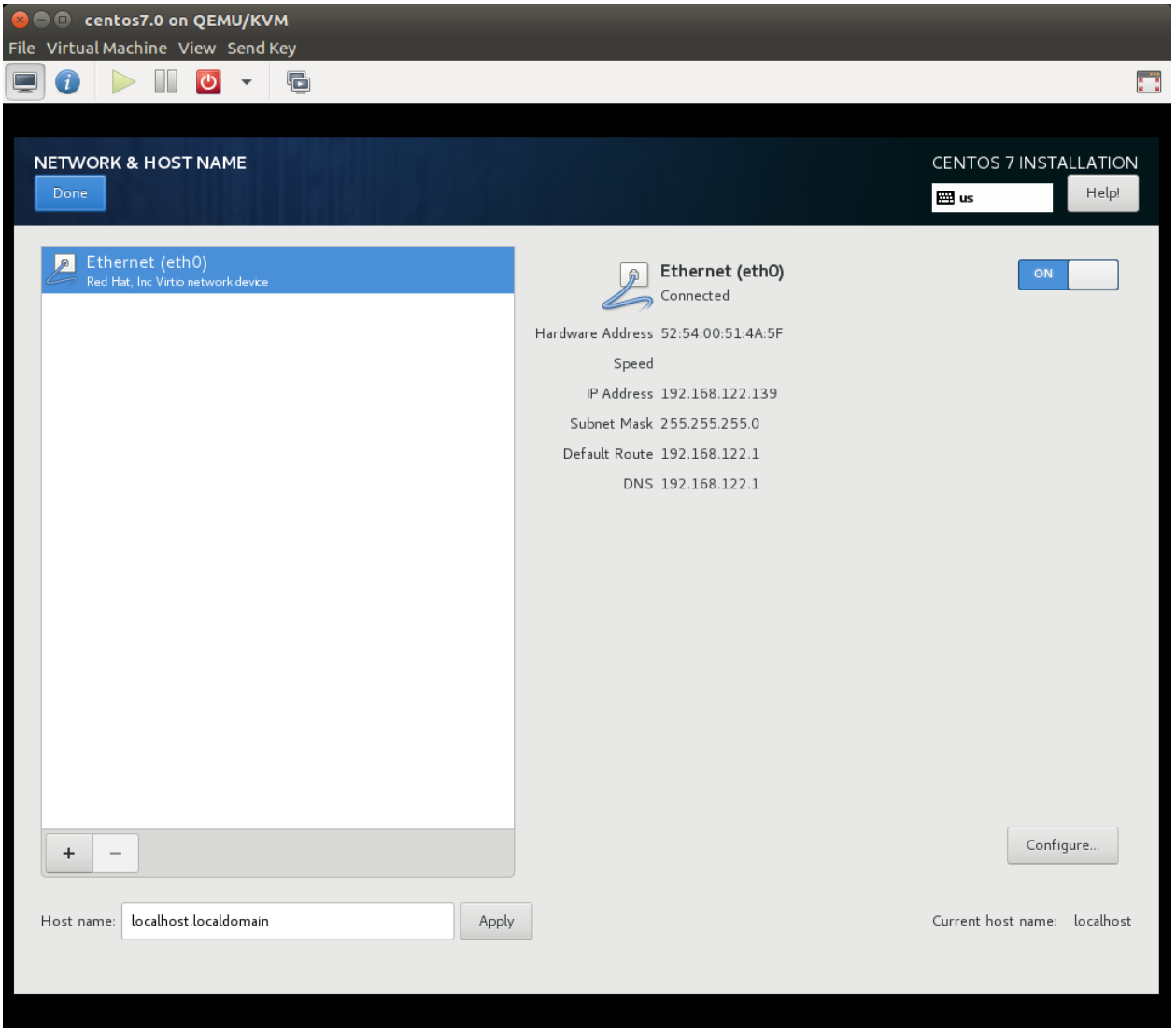


From here you will get a new console with the installation instructions for the OS you are installing (in this case CentOS 7):

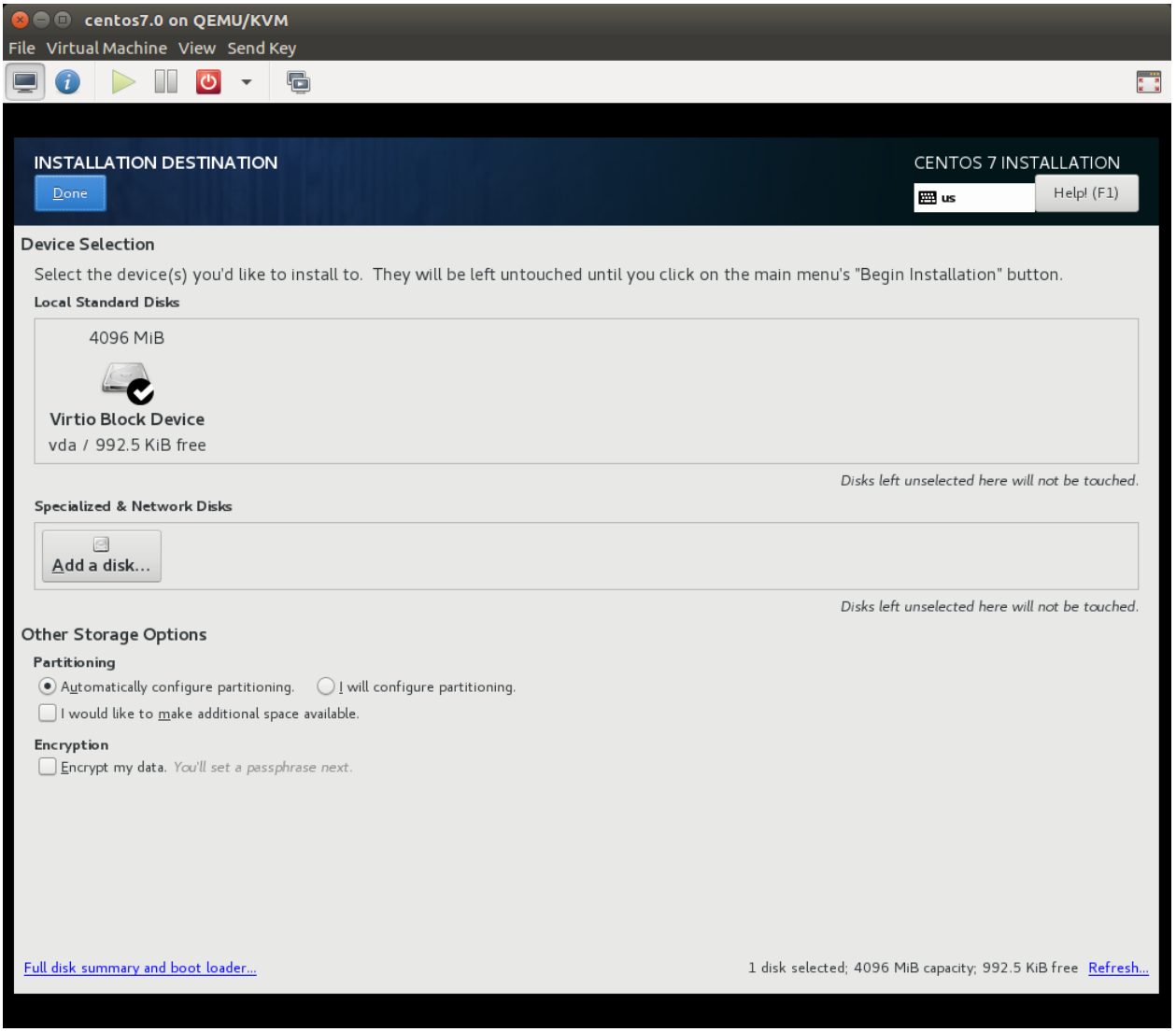

centos7.0 on QEMU/KVM
File Virtual Machine View Send Key
CentOS
CENTOS 7 INSTALLATION
us
Help! (F1)
WELCOME TO CENTOS 7.
What language would you like to use during the installation process?
English English
Afrikaans Afrikaans
አማርኛ Amharic
العربية Arabic
অসমীয়া Assamese
Asturianu Asturian
Беларуская Belarusian
Български Bulgarian
বাংলা Bengali
Bosanski Bosnian
Català Catalan
Čeština Czech
Cymraeg Welsh
Dansk Danish
Deutsch German
English (United States)
English (United Kingdom)
English (India)
English (Australia)
English (Canada)
English (Denmark)
English (Ireland)
English (New Zealand)
English (Nigeria)
English (Hong Kong SAR China)
English (Philippines)
English (Singapore)
English (South Africa)
English (Zambia)
English (Zimbabwe)
English (Botswana)
Quit
Continue



Select"Network& Host Name" and "Installation Destination"

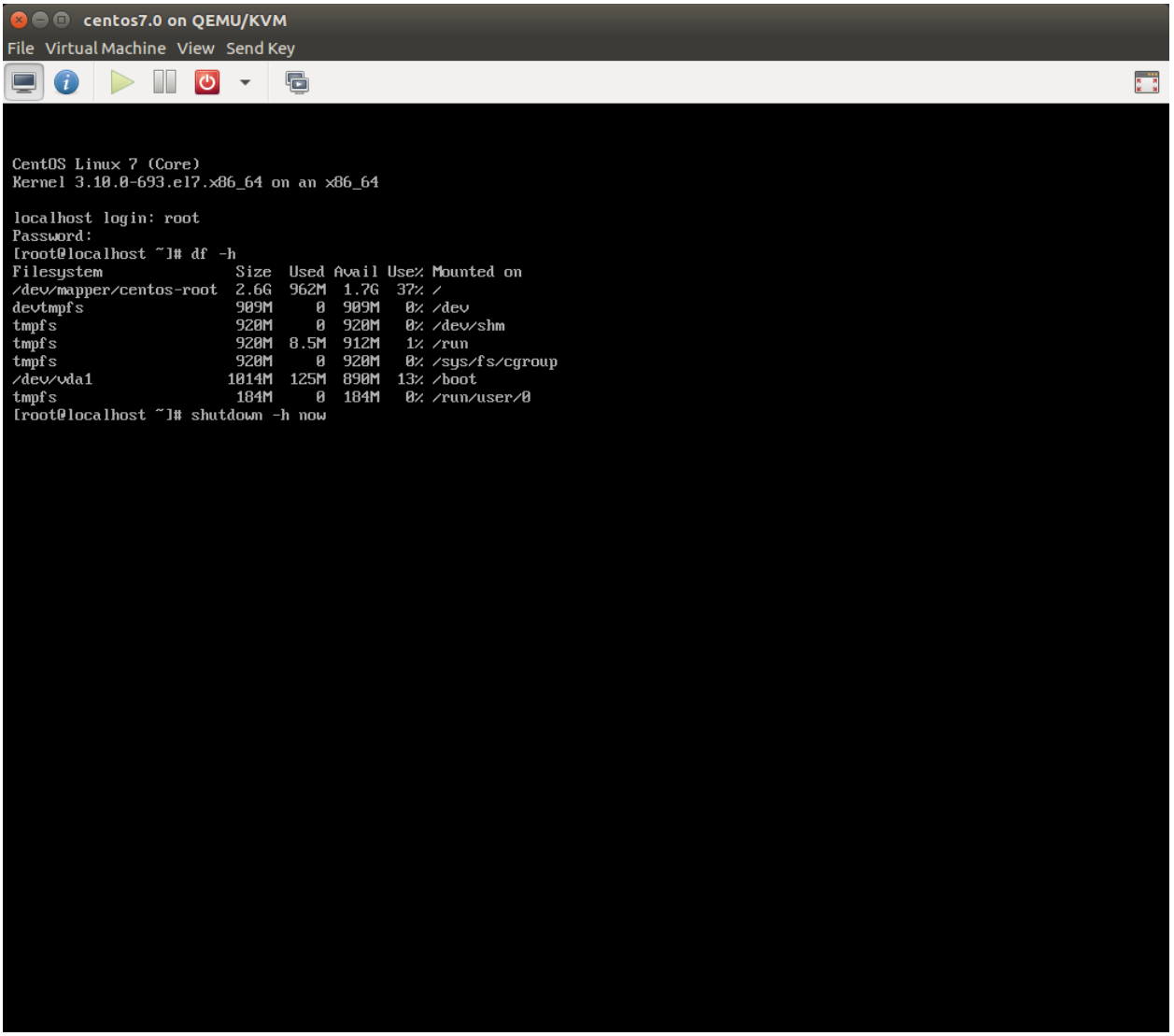


Networking is off by default, turn on.
Done

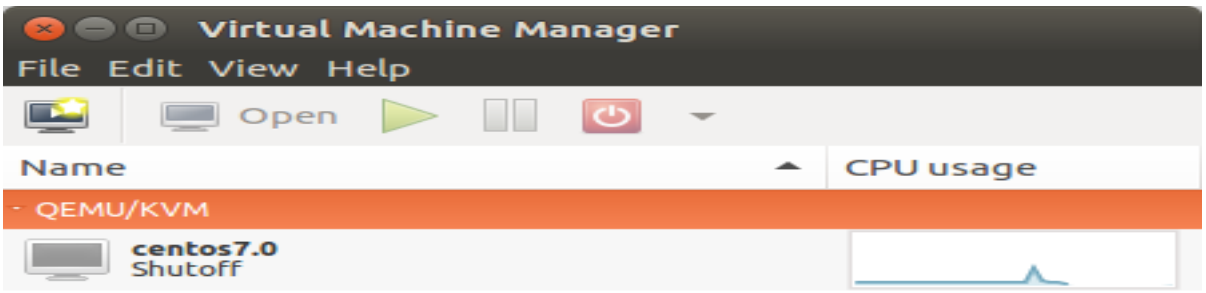


Unless this panel is selected, the "Begin Installation" button remains dim
Done

- Select "Begin Installation" on the Installation summary panel.
- Set passwords for root and a user. The installation will then proceed.
- The CentOS 7 installation ends with a request for a "Reboot". This will reboot the guest OS in the VM and not the host OS.



The CentOS 7 VM is up and running. It can be shutown like any other system: shutdown -h now



Use the virt-manager GUI to launch/re-launch any of the VMS. A console window can also be opened by selecting the VM and "Open".

Serving install media over http:

ASSINGMENT NO.	3
TITLE	Application in SalesForce.com using Apex programming Language.

PROBLEM STATEMENT /DEFINITION	Creating an Application in SalesForce.com using Apex programming Language.
OBJECTIVE	1. To student Apex Programming language 2. To create an application in SalesForce.com
OUTCOME	Application in SalesForce.com using Apex programming Language.
S/W PACKAGES AND HARDWARE APPARATUS USED	Web based platform
REFERENCES	SalesForce.com Help about Apex Programming: https://help.salesforce.com/s/articleView?id=sf.code_about.htm&type=5 https://www.salesforcetutorial.com/introduction-to-apex-programming/
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

Theory:

What is Apex in Salesforce?
Apex is an object-oriented and strongly typed programming language developed by Salesforce for building Software as a Service (SaaS) and Customer Relationship Management

(CRM). Apex helps developers to create third-party SaaS applications and add business logic to system events by providing back-end database support and client-server interfaces.
Apex helps developers to add business logic to the system events like button clicks, related record updates, and Visualforce pages. Apex has a similar syntax to Java.

Features of Apex Programming Language

Here are the important features of Salesforce Apex:

- Apex is a case insensitive language.
- You can perform DML operations like INSERT, UPDATE, UPSERT, DELETE on sObject records using apex.
- You can query sObject records using SOQL(salesforce object query language) and SOSL(salesforce object search language) in apex.
- Allows you to create a unit test and execute them to verify the code coverage and efficiency of the code in apex.
- Apex executes in a multi-tenant environment, and Salesforce has defined some governor limits that prevent a user from controlling the shared resources. Any code that crosses the salesforce governor limit fails, an error shows up.
- Salesforce object can be used as a datatype in apex. For example –

  ```
  Account acc = new Account();
  ```
  ,here Account is a standard salesforce object.

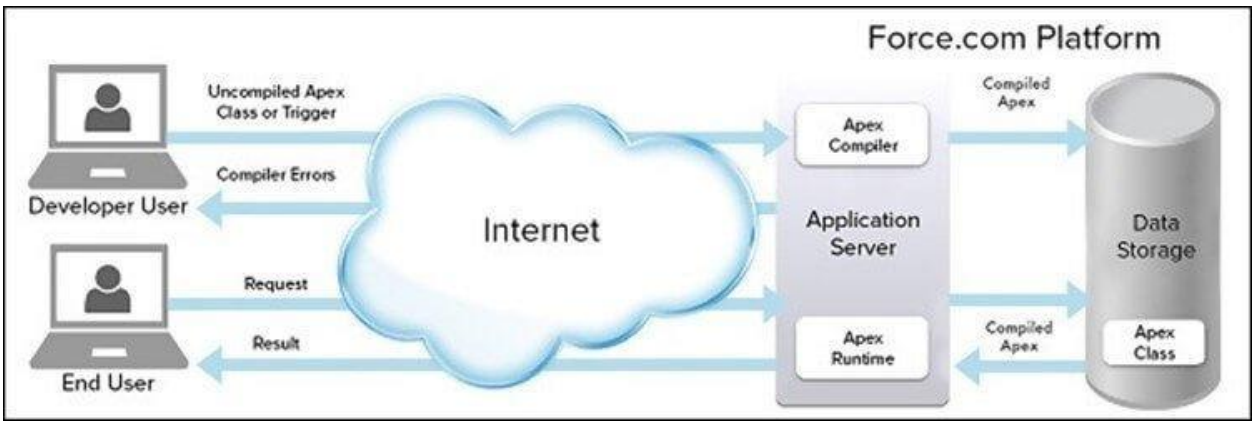
- Apex automatically upgrades with every Salesforce release.

When Should Developer Choose Apex
Apex code should only be written if a business scenario is too complex and can't be implemented using the pre-built functionality provided by Salesforce.
Following are the few scenarios where we need to write apex code:

- To create web services that integrate Salesforce with other applications.
- To implement custom validation on sobjects.
- To execute custom apex logic when a DML operation is performed.
- To implement functionality that can't be implemented using existing workflows flows and process builders functionality.
- To setup email services, you need to include processing the contents, headers, and attachments of email using apex code.

Following are the flow of actions for an apex code:

- Developer Action: All the apex code written by a developer is compiled into a set of instructions that can be understood by apex runtime interpreter when the developer saves the code to the platform and these instructions then save as metadata to the platform.
- End User Action: When the user event executes an apex code, the platform server gets the compiled instructions from metadata and runs them through the apex interpreter before returning the result.



ASSINGMENT NO.	4
TITLE	Custom Application (Mini Project) using Sales force Cloud.
PROBLEM STATEMENT /DEFINITION	Design and develop custom Application (Mini Project) using Sales force Cloud.

OBJECTIVE	To develop custom application (Mini Project) using salesforce cloud
OUTCOME	Application in SalesForce.com using Apex programming Language.
S/W PACKAGES AND HARDWARE APPARATUS USED	Web based platform
REFERENCES	SalesForce.com Help about Apex Programming: https://help.salesforce.com/s/articleView?id=sf.code_about.htm&type=5 https://www.salesforcetutorial.com/introduction-to-apex-programming/ https://www.edureka.co/blog/salesforce-tutorial
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

Theory:

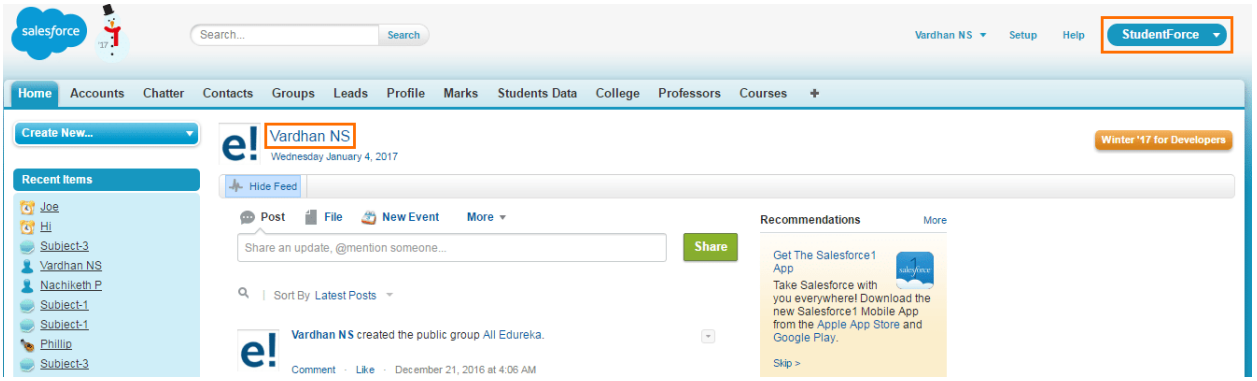
creating an app called *StudentForce* which can be used to maintain student records.

This app will contain three different objects (tables) to store data. The first object called *Students Data* will contain the names of students and their personal details like email id, phone number and native city. The college, students belong to, will be stored in the second object called *College* and the third object called *Marks* will contain the marks obtained by the students in various subjects.

Salesforce Apps

The primary function of a Salesforce app is to manage customer data. Salesforce apps provide a simple UI to access customer records stored in objects (tables). Apps also help in establishing relationship between objects by linking fields.

Apps contain a set of related tabs and objects which are visible to the end user. The below screenshot shows, how the *StudentForce* app looks like.



The highlighted portion in the top right corner of the screenshot displays the app name: *StudentForce*. You can do the following stpes.
Steps To Add Tabs
Steps To Create Custom Tabs

Salesforce Profiles

Steps To Create A Profile

Objects, Fields And Records In Salesforce

Steps To Add Custom Fields

Data Types Of Fields

Object Relationship In Salesforce

Master-Detail Relationship (1:n)

Lookup Relationship (1:n)

Self-Relationship

Junction Relationship (Many-To-Many)

Elective-II Augmented & Virtual Reality

ASSIGNMENT NO.	1
TITLE	Installation of Unity(Hub & Editor) and Visual Studio
PROBLEM STATEMENT /DEFINITION	Installation of Unity and Visual Studio, setting up Unity for VR development, understanding documentation of the same.
OBJECTIVE	To have Unity Game Engine ready for managing AVR assignment/projects
OUTCOME	Students will be able to ● Use Unity to create new AR & VR projects with associated components on Linux PCs ● Understand Unity documentation to work & follow the steps for creating projects in it.
S/W PACKAGES AND HARDWARE APPARATUS USED	**Operating system version:** Ubuntu 16.04 OR Ubuntu 18.04 CPU X64 architecture with SSE2 instruction set support **Graphics API: OpenGL 3.2+ or Vulkan-capable, Nvidia and AMD GPUs**
REFERENCES	**Unity Official website** **https://docs.unity3d.com/2020.1/Documentation/Manual**
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 9. Conclusion/Analysis

Installing the Unity Hub

The Hub is the primary way to install the Unity Editor, create projects, and manage your Unity experience. It provides a central location to manage your Editor installations, Accounts and Licenses, and Projects

Note: If Unity Hub fails to launch while you are using Linux, you might need to give UnityHub.AppImage executable permissions. To do this:

1. Open your terminal.
2. Go to the directory where UnityHub.AppImage is. This will be the Unity Hub directory.
3. Run `chmod +x UnityHub.AppImage`.

To install and use the Unity Editor, you must have a Unity Developer Network (UDN) account. If you already have an account, sign in, choose your licenses type, and proceed to the Installing the Unity Editor section.
If you do not have an account, follow the prompts to create one. You can choose to create a Unity ID or use one of the social sign-ins

To install the Unity Hub for Windows, macOS, and Linux visit Download Unity on the Unity website. As given below

https://unity.com/download

Steps to install

Installing the Hub on Linux

To install the Unity Hub on a Debian or Ubuntu Linux distribution, you need to add the Unity Hub Debian repository along with the public signing key to verify the integrity of the packages.

To add the Unity Hub repository, you need an entry in /etc/apt/sources.list.

1. Run the following command to add the Unity Hub repository:

```
$ sudo sh -c 'echo "deb https://hub.unity3d.com/linux/repos/deb stable main" > /etc/apt/sources.list.d/unityhub.list'
```

2. To add the public signing key, run the following command:

   ```
   $ wget -qO - https://hub.unity3d.com/linux/keys/public | sudo apt-key add -
   ```

3. Then update the package cache and install the package using:

   ```
   $ sudo apt update
   ```

   ```
   $ sudo apt-get install unityhub
   ```

Installing the Unity Editor

To install the Editor:

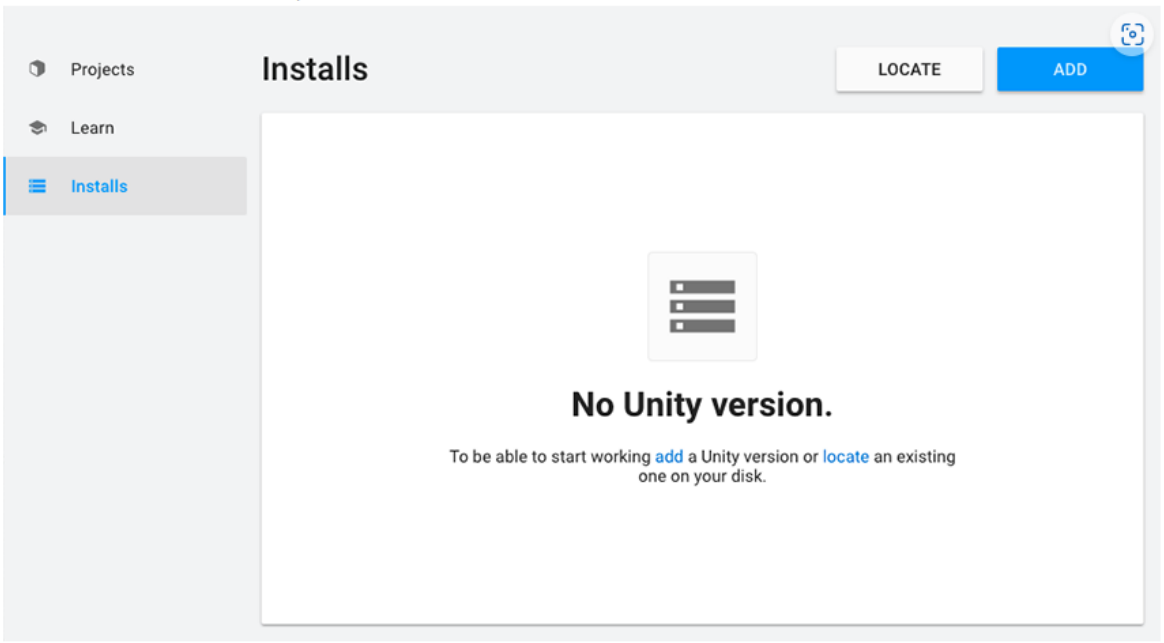
1. Click the **Installs** tab. The default install locations are:

   `~/Unity/Hub/Editor`

Note: If you want to change the default installation location, follow these steps:

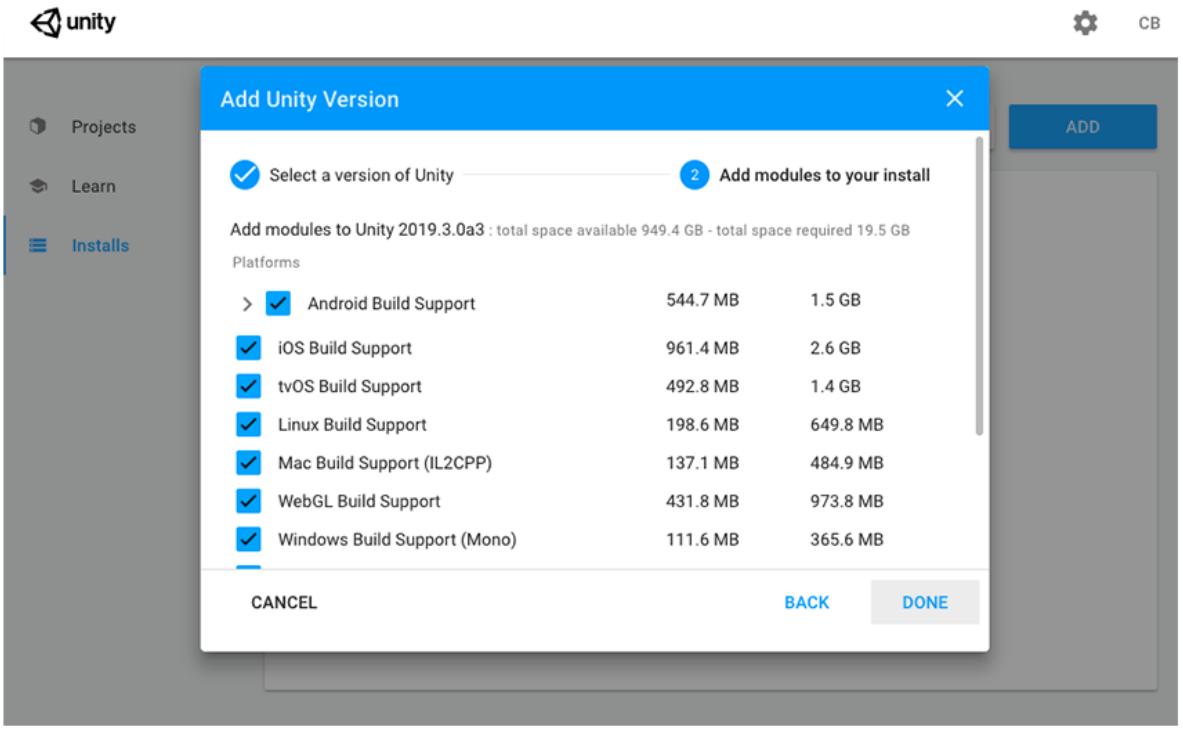
1. From the top right corner of the Hub window, click the Gear icon.
2. In the **Editor Folder Location** dialog box, enter the new installation location and click **Done**.

2. Click the **Add** button and select a specific version of the Editor.



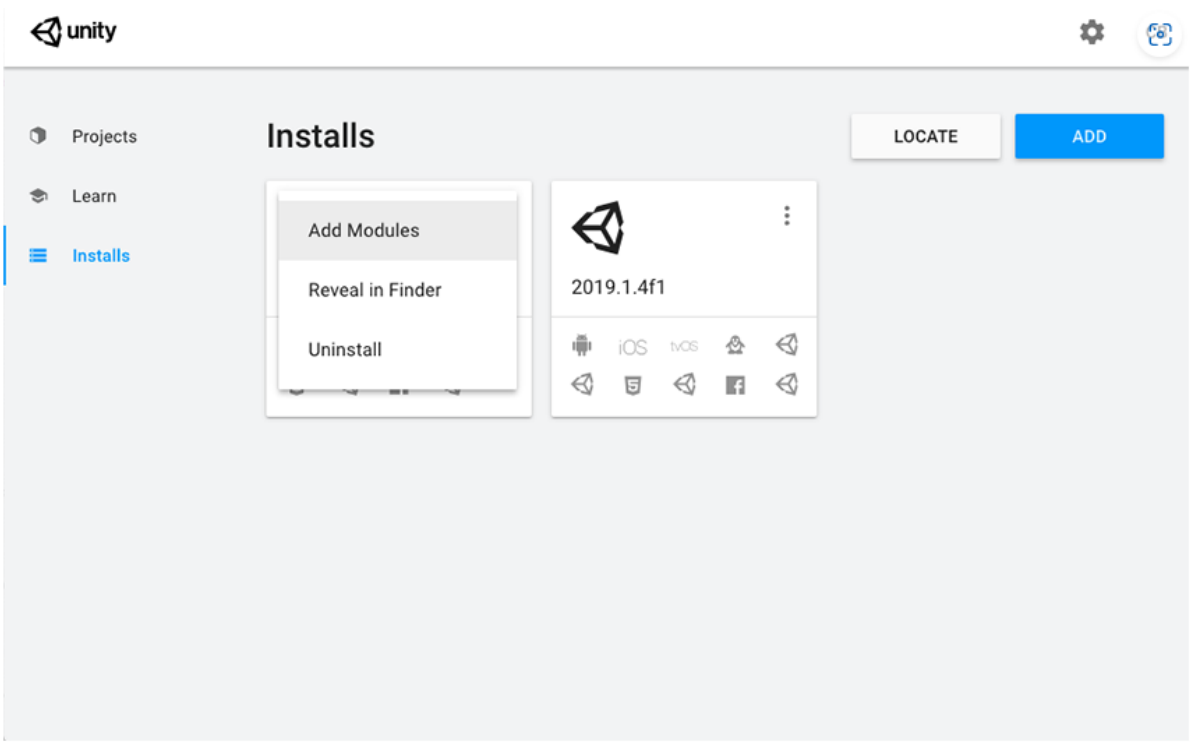
Hub install screen

3. Click the **Next** button and select the modules you want to install with the Editor. If you don't install a component now, you can add it later if you need to. When you've selected all the modules you need, click **Done**.



The Hub displays the installation location of each Editor under the corresponding version label.

To add modules to an Editor, locate its files, or uninstall it, click the three dots next to that Editor version.



Troubleshooting for Linux

If Unity fails to start, you might need to install a missing **dependency**. On Ubuntu-based distributions, use:

```
sudo apt install libgconf-2-4
```

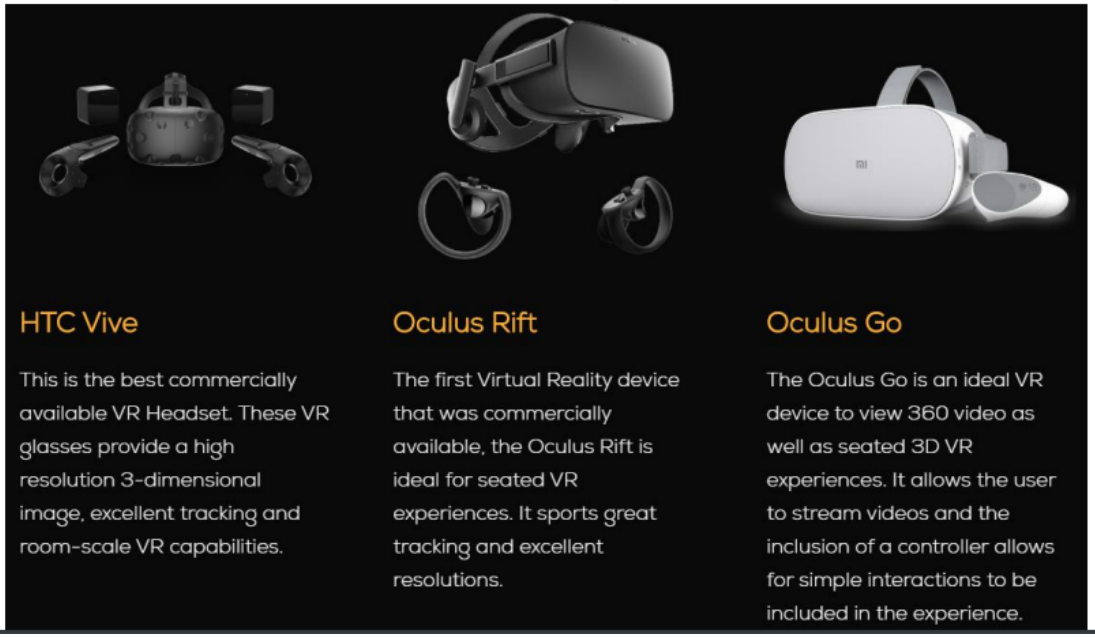
Conclusion: By following above steps we can install Unity Hub & Editor for Linux system & after successful sign in & verification of user account, can start using it for creating projects.

ASSIGNMENT NO.	2
TITLE	Demonstration of VR devices.
PROBLEM STATEMENT /DEFINITION	Demonstration of the working of HTC Vive, Google Daydream or Samsung gear VR.
OBJECTIVE	To get familiar with VR devices & get experience of VR world
OUTCOME	Students will be able to ● Use any of the available VR devices ● Experience the virtual world & Understand the basics of it
S/W PACKAGES AND HARDWARE APPARATUS USED	VR devices: HTC Vive, Google Daydream or Samsung gear VR.
REFERENCES	**Demonstration link** **https://www.youtube.com/watch?v=1t4_uXr9YiY**
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 9. Conclusion/Analysis

*Video Demonstration link:

1. https://www.youtube.com/watch?v=1t4_uXr9YiY: Use of VR for Psychosis
2. https://www.youtube.com/watch?v=RPnXcxSwggo : **Matterport Hotel Virtual Tour Example Walkthrough**

Virtual Reality Devices



HTC Vive

This is the best commercially available VR Headset. These VR glasses provide a high resolution 3-dimensional image, excellent tracking and room-scale VR capabilities.

Oculus Rift

The first Virtual Reality device that was commercially available, the Oculus Rift is ideal for seated VR experiences. It sports great tracking and excellent resolutions.

Oculus Go

The Oculus Go is an ideal VR device to view 360 video as well as seated 3D VR experiences. It allows the user to stream videos and the inclusion of a controller allows for simple interactions to be included in the experience.

Virtual Reality Devices

Oculus Quest

The Oculus Quest is the next-generation VR headset which allows tracking without outside sensors and can run VR without a powerful PC.

Mixed Reality Device

The Mixed Reality Devices from Microsoft are ideally suited for mobile VR experiences. These devices use inside-out tracking which means that they can track their position without setting up external tracking devices.

VR Cardboard

Google Cardboard are devices that make it possible to view 360 videos or seated VR experiences using your own mobile phone. These cardboards can be designed to match your own brand identity and make a great gift for your clients.

Use Cases of VR

Training Footballers using VR

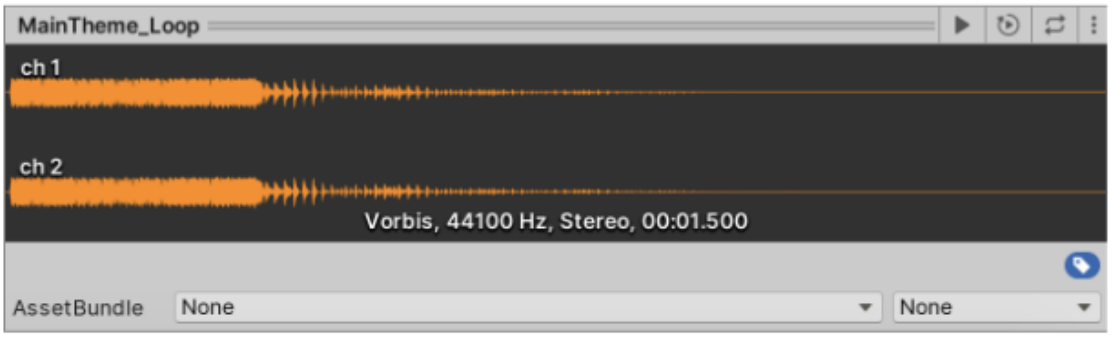
VR in Surgical Training

ASSIGNMENT NO.	3
TITLE	Unity Scene development with 3D Game Objects
PROBLEM STATEMENT /DEFINITION	Develop a scene in Unity that includes: i. A cube, plane and sphere, apply transformations on the 3 game objects. ii. Add a video and audio source.
OBJECTIVE	• To get familiar with Unity Editor UI • To perform 3D transformations on 3D objects in Unity • To use audio & video resources in Unity
OUTCOME	Students will be able to • Use & implement 3D transformation tools/options available in Unity editor • Use UI objects for audio & video files from Unity editor
S/W PACKAGES AND HARDWARE APPARATUS USED	• Operating system version: Ubuntu 16.04 OR Ubuntu 18.04 • CPU X64 architecture • Unity
REFERENCES	https://learn.unity.com/tutorial/working-with-audio-components-2019-3#5f8fa275edbc2a284332bcec Play video in unity 3D - Gyanendu Shekhar's Blog
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 9. Conclusion/Analysis

PART I: Audio Component

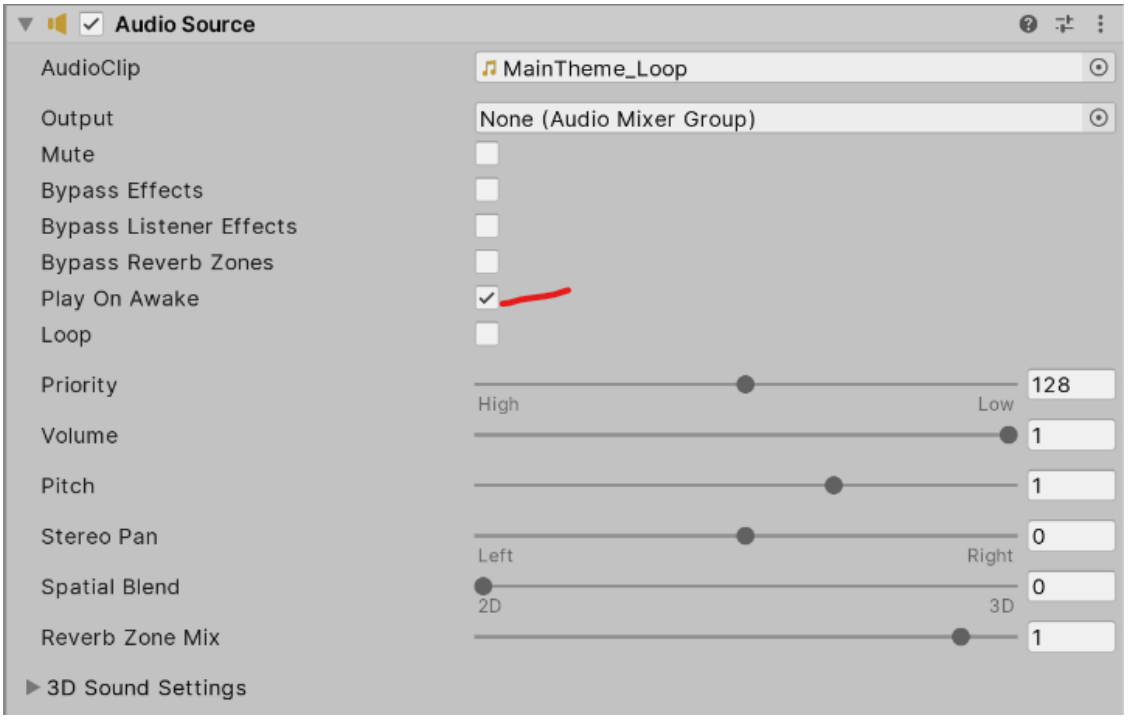
1. Create or save audio & video files required for unity
   Note: Unity supports AIFF, WAV, & Ogg formats (for audio)
2. Import your audio files in Unity Project. Either drag the audio file into Project Panel or place the audio file in the **Assets** directory of the Unity Project directory.
3. In the **Project** panel, select the imported Audio asset
4. In the **Inspector** panel, change import settings if necessary.

5. At the bottom of the Inspector panel is a waveform preview of the imported audio file. you can preview the audio by pressing play as shown in image below:

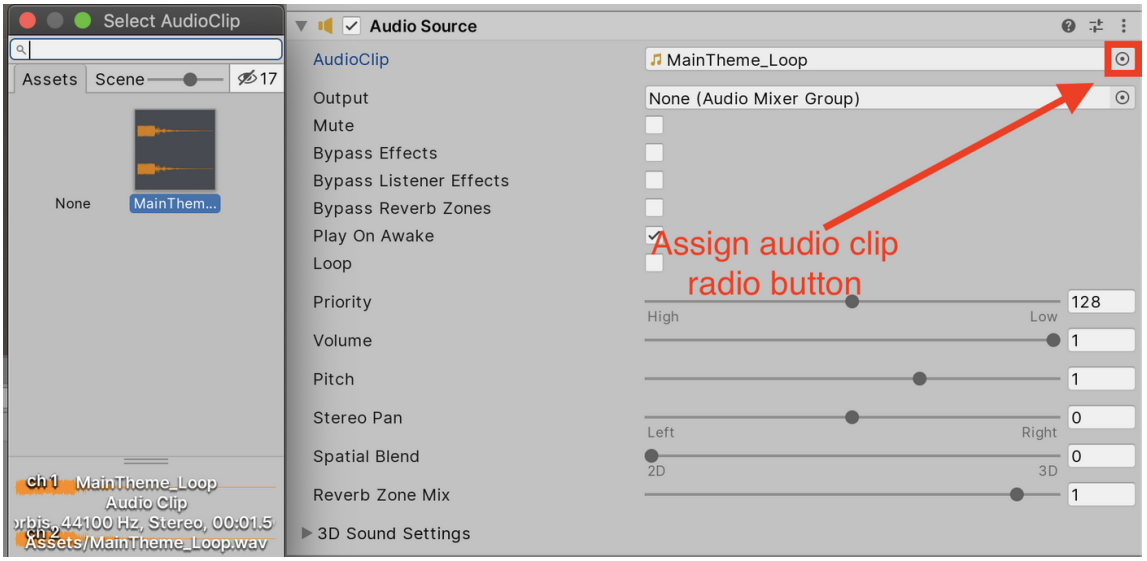


To add an Audio Source Component:

1. Select the **GameObject menu > Audio > Audio Source** …. This will create a GameObject in the scene with an Audio Source component attached



2. Assign the previously imported audio file to the **Audio Clip** property of the Audio Source Component in the Inspector. You can do this by dragging the audio clip from the Project panel into the Audio Clip property field or clicking the radio button next to the property field & selecting the audio clip from Assets window as shown below.



PART II: Video Component

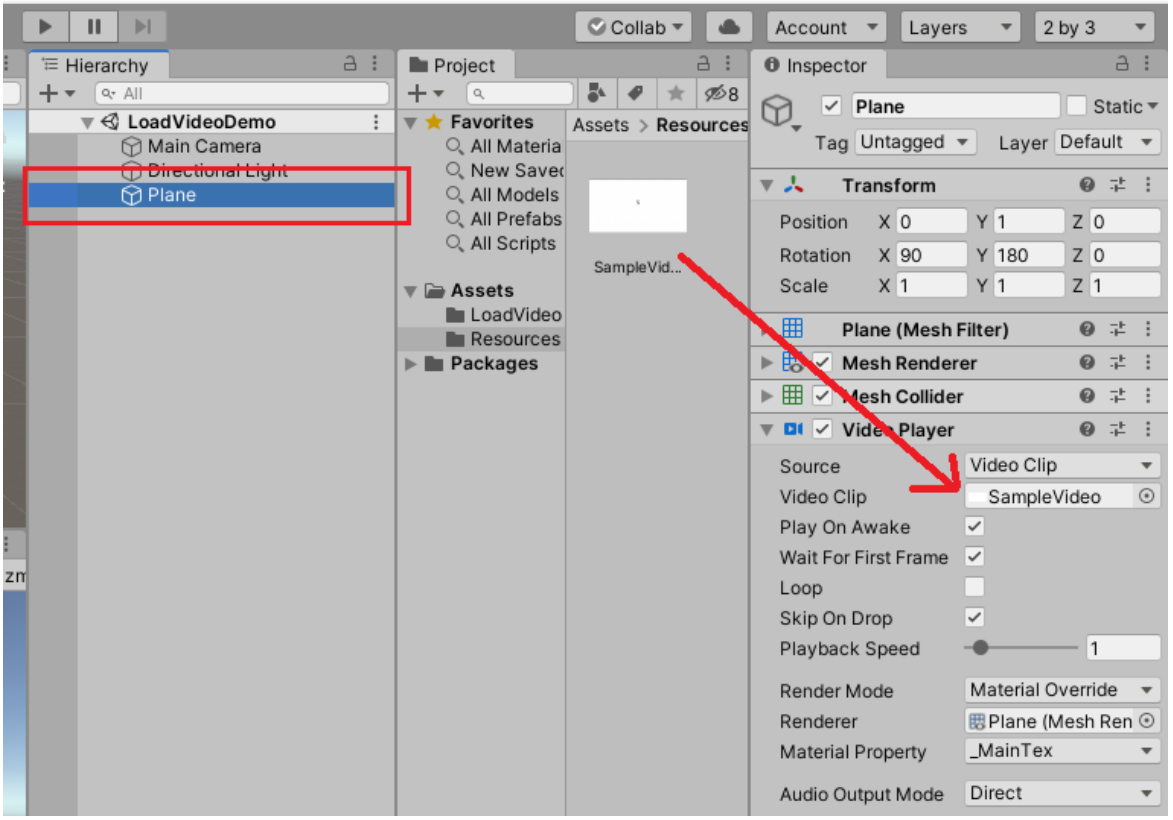
To play a video, we need to add a **VideoPlayer** component to the game object.

Play video on a plane

Step 1: Create a Plane primitive game object. (Game Object -> 3D Object -> Plane)

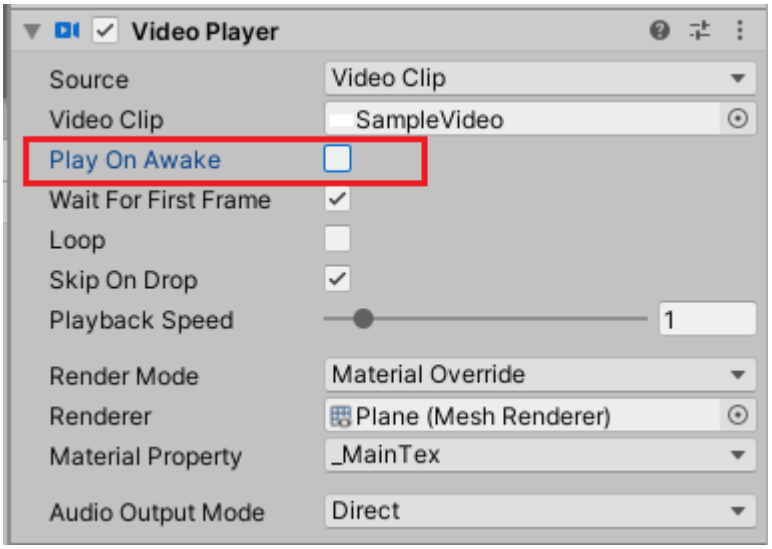
Step 2: Add VideoPlayer component to it.

Step 3: Drag and drop video file to Video Clip input field in the editor



Play video at runtime

By default, video will play on awake. Uncheck this checkbox, to play video at runtime.



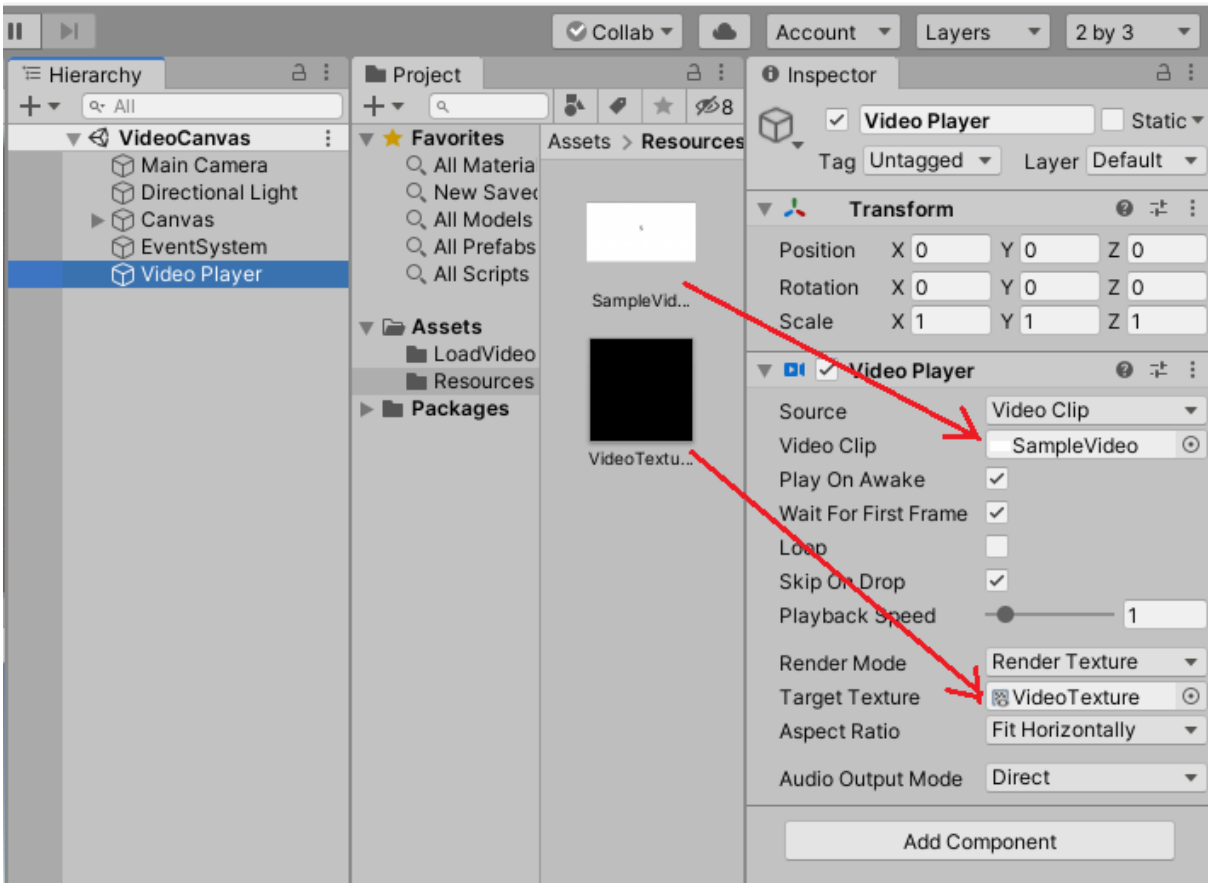
Use the below script to play at runtime.

```
using UnityEngine;
using UnityEngine.Video;

public class PlayRuntime : MonoBehaviour
{
   private VideoPlayer MyVideoPlayer;

   private void Start()
   {
      MyVideoPlayer = GetComponent<VideoPlayer>();
      // play video player
      MyVideoPlayer.Play();
   }
}
```

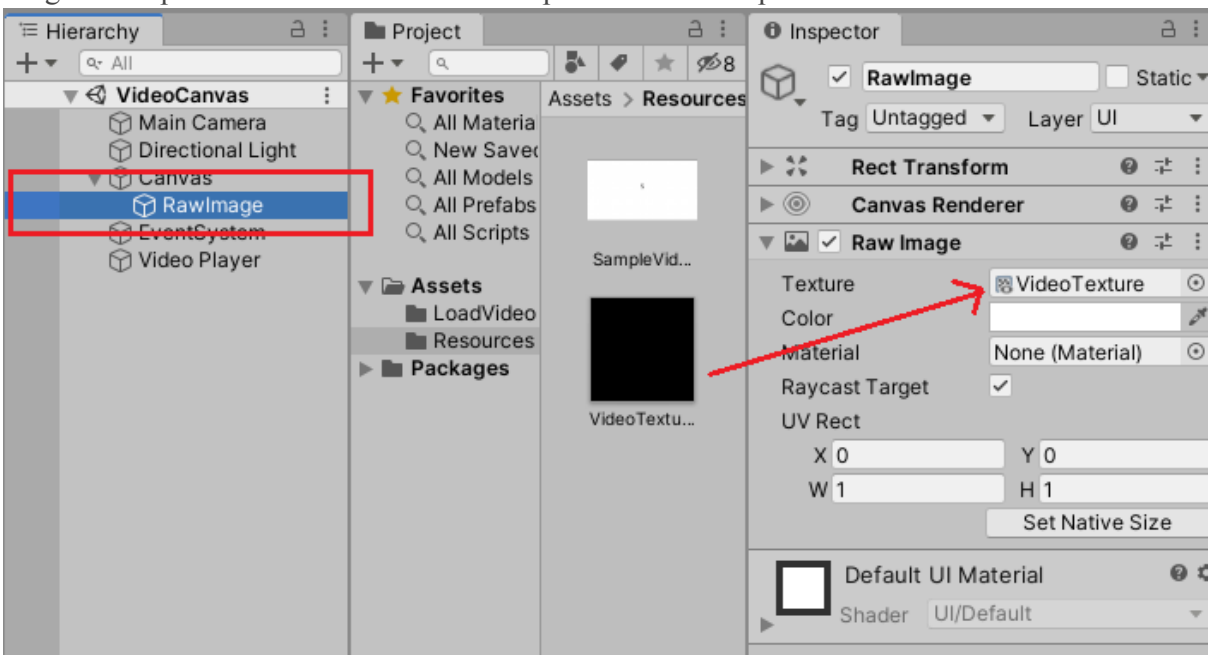
Play Video on Canvas element

Step 1: Create a Video Player game object in the scene. (**Game Object -> Video -> Video Player**). Drag and drop your video to the Video Clip field like step 3 above section.

Step 2: Create a Render Texture in Assets folder. (**Assets -> Create -> Render Texture**). Drag and drop this render texture to the Target Texture field of Video Player component.



Step 3: Create Raw Image Game Object in the Scene. Resize the game object as per your need. Drag and drop Render Texture of the last step to its Texture input field.



Setup is now complete. Run the unity to see the result.

Conclusion: By Using above steps, we can play audio & video in Unity scene.

ASSIGNMENT NO.	4
TITLE	To Change 3D Game Objects' material & color with ith button click event in Unity Scene
PROBLEM STATEMENT /DEFINITION	Develop a scene in Unity that includes a cube, plane and sphere. Create a new material and texture separately for three Game objects. Change the color, material and texture of each Game object separately in the scene. Write a C# program in visual studio to change the color and material/texture of the game objects dynamically on button click.
OBJECTIVE	● To get familiar with Unity Editor UI ● To use material & Color property of objects in Unity ● To perform button click event handling script execution on 3D objects in Unity
OUTCOME	Students will be able to ● Use material & Color property of Game objects in Unity ● Perform Button click event handling with C# script in Unity
S/W PACKAGES AND HARDWARE APPARATUS USED	● Operating system version: Ubuntu 16.04 OR Ubuntu 18.04 ● CPU X64 architecture ● Unity
REFERENCES	**https://learn.unity.com/tutorial/working-with-audio-components-2019-3#5f8fa275edbc2a284332bcec**
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 9. Conclusion/Analysis

ASSIGNMENT NO.	5
TITLE	Marker Based Tracking in AR App
PROBLEM STATEMENT /DEFINITION	Develop and deploy a simple marker based AR app in which you have to write a C# program to play video on tracking a particular marker.
OBJECTIVE	• To get familiar with AR tool & technique for Marker based tracking • To develop AR app using simple C# script for playing video
OUTCOME	Students will be able to • Explore marker based detection techniques • develop simple AR application for marker based detection or tracking.
S/W PACKAGES AND HARDWARE APPARATUS USED	• Operating system version: Ubuntu 16.04 OR Ubuntu 18.04 • CPU X64 architecture • Unity
REFERENCES	**Build a Marker-Based AR App in 6 minutes | Source Code included | kandi tutorial - YouTube**
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 9. Conclusion/Analysis

INFORMATION SECURITY

ASSINGMENT NO.	1 (Information Security)
TITLE	AND / XOR
PROBLEM STATEMENT /DEFINITION	Write a Java/C/C++/Python program that contains a string (char pointer) with a value \Hello World'. The program should AND or and XOR each character in this string with 127 and display the result.
OBJECTIVE	To understand the concepts & implementation of AND or and XOR.
OUTCOME	Implementation of AND or and XOR .
S/W PACKAGES AND HARDWARE APPARATUS USED	Core 2 DUO/i3/i5/i7 64-bit processor OS-LINUX 64 bit OS Editor-gedit/Eclipse S/W- C++/JAVA//Python
REFERENCES	1. Bernard Menezes, "Network Security and Cryptography", Cengage Learning India, 2014, ISBN No.: 8131513491 2. Nina Godbole, Sunit Belapure, "Cyber Security", Wiley India, 2014, ISBN No.: 978-81-345-2179-1 3. Atul Kahate, "Cryptography and Network Security", Mc Graw Hill Publication, 2nd Edition, 2008, ISBN: 978-0-07-064823- 4. William Stallings, "Cryptography and network security Principles and practices", Pearson, 6th Edition, ISBN: 978-93-325-1877-3 5. Forouzan, "Cryptography and Network Security (SIE)", Mc Graw Hill , ISBN 007070208X, 9780070702080
STEPS	1. Start 2. AND/XOR 3. End
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory

	7. Algorithm 8. Test cases 10. Conclusion/Analysis

Prerequisites: Discrete mathematics, any programming language Java/C++/Python.

Concepts related Theory:

A. String:

The String class represents character strings. All string literals in Java programs, such as "abc", are implemented as instances of this class.

Strings are constant; their values cannot be changed after they are created. String buffers support mutable strings. Because String objects are immutable they can be shared. For example:

```
String str = "abc";
```

B. AND Operation:

The three Binary Operations are AND Operation, OR and NOT Operation. A binary logic deals with variables having distinct values like true or false, yes or no, hot or cold, high or low, etc.

In a digital system design, assigning the values 1 or 0 to these two values of the variables because digital algebraic operations involve the binary number system.

Binary logics consist of binary variables such as x, y, z, a, b, c, etc. and three basic logical operations namely AND, OR and NOT. Each binary variable has two and only two distinct values 0 and 1.

Here we are discussing the AND operation. The **AND operation** is represented by a dot or by the absence of an operator. The **And Operation** is mostly used in the logical operation in digital electronics and for designing the digital circuit. The symbolic representation of the AND gate is shown below:

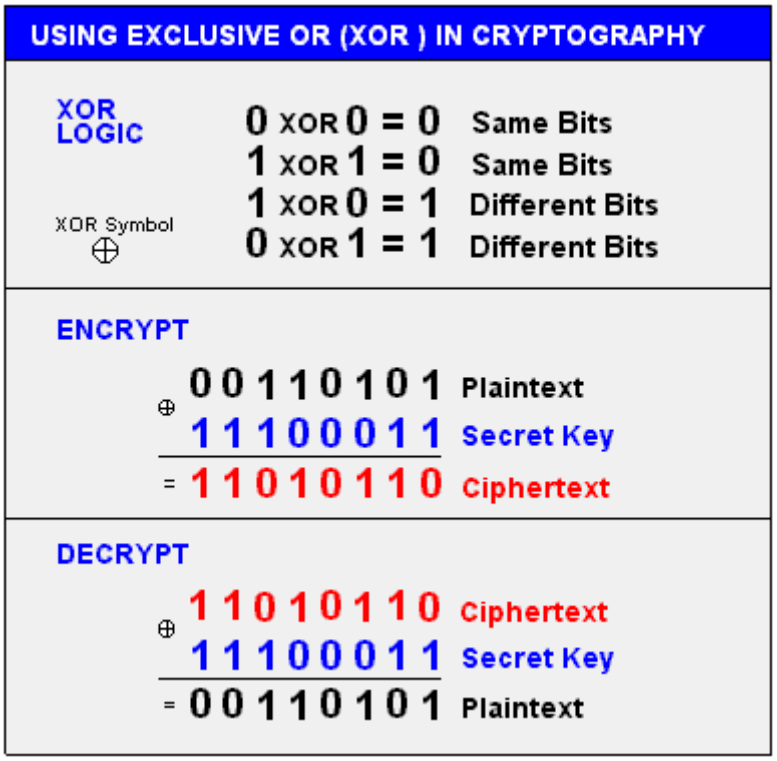
For example: x.y = z or xy = z and it is read as “x and y is equal to z”. The result of logic operations can be best demonstrated by a truth table.

The truth table for the AND operation is shown below:

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

C. XOR Operation

(eXclusive OR) A Boolean logic operation that is widely used in cryptography as well as in generating parity bits for error checking and fault tolerance. XOR compares two input bits and generates one output bit. The logic is simple. If the bits are the same, the result is 0. If the bits are different, the result is 1.



Conclusion:

Hence, we have learned the concepts & implementation of AND or/and XOR gates.

Review Questions:

1. Explain AND Operation?
2. What is OR Operation?
3. Define: String

ASSINGMENT NO.	2 (Information Security)
TITLE	Encryption and decryption using the method of Transposition technique.
PROBLEM STATEMENT /DEFINITION	Write a Java/C/C++/Python program to perform encryption and decryption using the method of Transposition technique.
OBJECTIVE	● To understand the concepts of encryption and decryption. ● To understand the use of Transposition technique.
OUTCOME	Implementation of Transposition technique for encryption and decryption.
S/W PACKAGES AND HARDWARE APPARATUS USED	Core 2 DUO/i3/i5/i7 64-bit processor OS-LINUX 64 bit OS Editor-gedit/Eclipse S/W- C++/JAVA//Python
REFERENCES	1. Bernard Menezes, "Network Security and Cryptography", Cengage Learning India, 2014, ISBN No.: 8131513491 2. Nina Godbole, Sunit Belapure, "Cyber Security", Wiley India, 2014, ISBN No.: 978-81-345-2179-1 3. Atul Kahate, "Cryptography and Network Security", Mc Graw Hill Publication, 2nd Edition, 2008, ISBN: 978-0-07-064823- 4. William Stallings, "Cryptography and network security Principles and practices", Pearson, 6th Edition, ISBN: 978-93-325-1877-3 5. Forouzan, "Cryptography and Network Security (SIE)", Mc Graw Hill , ISBN 007070208X, 9780070702080
STEPS	1.Key generation 2.Encryption 3. Decryption
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory

	7. Algorithm 8. Test cases 10. Conclusion/Analysis

Prerequisites: Discrete mathematics, any programming language Java/C++/Python.

Concepts related Theory:

Transposition cipher

In cryptography, a transposition cipher is a method of encryption by which the positions held by units of plaintext (which are commonly characters or groups of characters) are shifted according to a regular system, so that the ciphertext constitutes a permutation of the plaintext. That is, the order of the units is changed.Mathematically a bijective function is used on the characters' positions to encrypt and an inverse function to decrypt.

Following are some implementations.

1. Rail Fence Transposition
2. Columnar Transposition
3. Improved Columnar Transposition
4. Book Cipher/Running Key Cipher

1. Rail Fence Transposition:

The rail fence cipher is the simplest transposition cipher. The steps to obtain cipher text using this technique are as follow:

Step 1: The plain text is written as a sequence of diagonals.

Step 2: Then, to obtain the cipher text the text is read as a sequence of rows.

To understand this in a better way, let us take an example:

Plain Text: meet me Tomorrow

Now, we will write this plain text sequence wise in a diagonal form as you can see below:

Looking at the image, you would get it why it got named rail fence because it appears like the rail fence.

Once you have written the message as a sequence of diagonals, to obtain the cipher text out of it you have to read it as a sequence of rows. So, reading the first row the first half of cipher text will be:

m e m t m r o

reading the second row of the rail fence, we will get the second half of the cipher text:

e t e o o r w

Now, to obtain the complete cipher text combine both the halves of cipher text and the complete

cipher text will be:

Cipher Text: M E M T M R O E T E O O R W

Rail fence cipher is easy to implement and even easy for a cryptanalyst to break this technique. So,

there was a need for a more complex technique.

2. Columnar Transposition Technique

The columnar transposition cipher is more complex as compared to the rail fence. The steps to obtain cipher text using this technique are as follow:

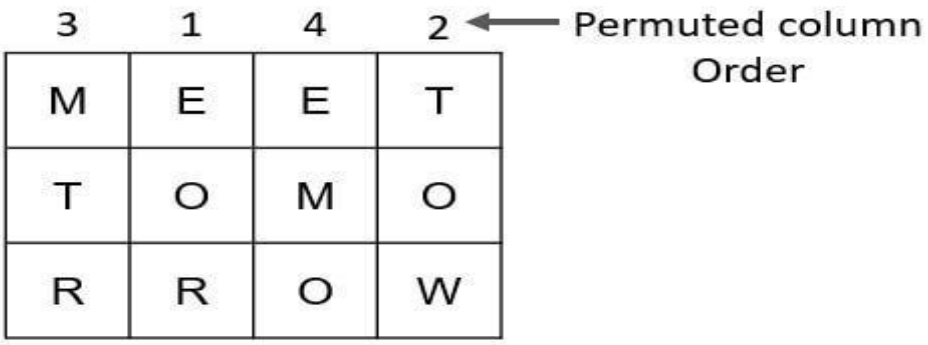
Step 1: The plain text is written in the rectangular matrix of the initially defined size in a row by row pattern.

Step 2: To obtain the cipher text read the text written in a rectangular matrix column by column. But you have to permute the order of column before reading it column by column. The obtained message is the cipher text message.

To understand the columnar transposition let us take an example:

Plain text: meet Tomorrow

Now, put the plain text in the rectangle of a predefined size. For our example, the predefined size of the rectangle would be 3×4. As you can see in the image below the plain text is placed in the rectangle of 3×4. And we have also permuted the order of the column.



Now, to obtain the cipher text we have to read the plain text column by column as the sequence of permuted column order. So, the cipher text obtained by the columnar transposition technique in this example is:

Cipher Text: MTREOREMOTOW.

Similar to the rail fence cipher, the columnar cipher can be easily broken. The cryptanalyst only has to try few permutation and combination over the order of column to obtain the permuted order of column and the get the original message. So, a more sophisticated technique was required to strengthen the encryption.

3. Columnar Transposition Technique with Multiple Rounds

It is similar to the basic columnar technique but is introduced with an improvement. The basic columnar technique is performed over the plain text but more than once. The steps for columnar technique with multiple rounds are as follow:

Step 1: The plain text is written in the rectangle of predetermined size row by row.

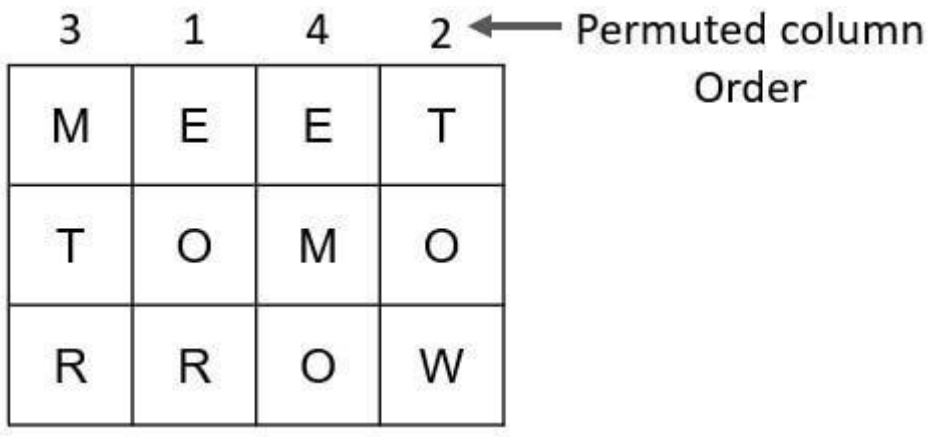
Step 2: To obtain the cipher text, read the plain text in the rectangle, column by column. Before reading the text in rectangle column by column, permute the order of columns the same as in basic columnar technique.

Step 3: To obtain the final cipher text repeat the steps above multiple time.

Let us discuss one example of a columnar transposition technique for better understanding. We will consider the same example of a basic columnar technique which will help in understanding the complexity of the method:

Plain Text: meet Tomorrow

Let us put this plain text in the rectangle of predefined size of 3×4. Proceeding with the next step, the order of the columns of the matrix is permuted as you can see in the image below:



Now after the first round the cipher text obtained is as follow:

Cipher Text round 1: MTREOREMOTOW

Now, again we have to put the cipher text of round 1 in the rectangle of size 3×4 row by row and permute the order of columns before reading the cipher text for round 2. In the second round, the permuted order of the column is 2, 3, 1, 4.

So, the obtained **cipher text for round 2** is MOOTRTREOEMW. In this way, we can perform as many iterations as requires. Increasing the number of iterations increases the complexity of the techniques.

4. Book Cipher or Running Key Cipher

The book cipher or the running key cipher works on the basic principle of one-time pad cipher. In onetime pad cipher the key is taken as long as the plain text and is discarded after the use. Every time a new key is taken for a new message.

The improvement to the onetime pad in Book cipher is that the key or the onetime pad is taken from the book. Let us discuss the steps:

Step 1: Convert the plain text in numeric form consider A=0, B=1, C=3 …, Z=25.

Step 2: Take an onetime pad or key from any of the books and convert it in the numeric form also. But the key must be as long as the length of plain text.

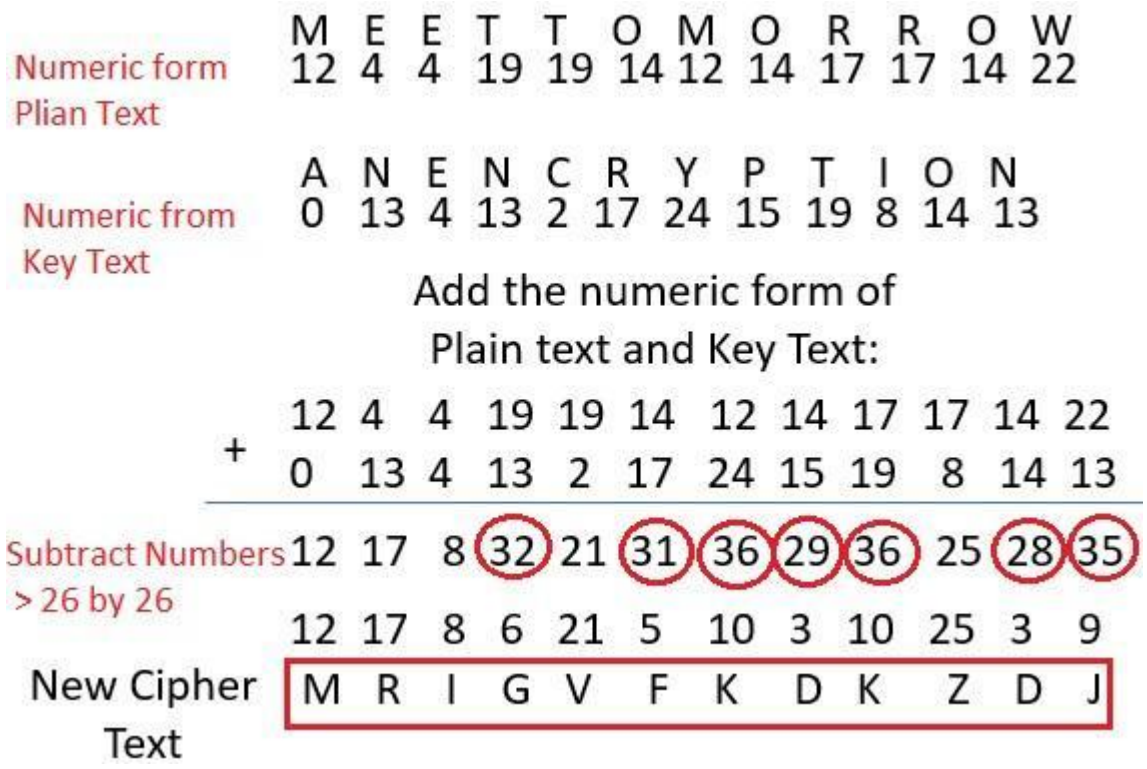
Step 3: Now add the numeric form of both plain text and key, each plain text letter with corresponding key text letter. If the addition of any plain text letter with corresponding key text letter is >26, then subtract it with 26.

Let us understand with the example:

Plain text: Meet Tomorrow

Key taken from the book: ANENCRYPTION.

Now we have to convert this plain text and key text in numeric form and add them to get cipher text as shown in the image below:



The **cipher text** obtained is MRIGVFKDKZDJ.

So, this is all about the Transposition technique, which involves the permutation over the plain text for converting plain text into the cipher text.

Conclusion:

Hence, we have studied the various Transposition techniques for encryption & decryption.

Review Questions:

1. What is encryption & decryption?
2. Explain Rail Fence Transposition.
3. Explain Running Key Cipher.

ASSIGNMENT NO	03 (Information Security)
TITLE	DES
PROBLEM STATEMENT/DEFINITION	Write a Java/C/C++/Python program to implement the DES algorithm.
OBJECTIVE	To understand the use & implementation of DES algorithm.
OUTCOME	Students will be able to implement DES algorithm successfully.
S/W PACKAGES AND HARDWARE APPARATUS USED	Core 2 DUO/i3/i5/i7 64-bit processor OS-LINUX 64 bit OS Editor-gedit/Eclipse S/w- Jupyter Notebook/ Weka/ Python
REFERENCES	Cryptography & Network Security, Behrouz A. Forouzan, Tata McGraw Hill.
STEPS	1. Key Generation 2. Encryption 3. Decryption
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment No. 3. Problem Definition 4. Learning Objective 5. Learning Outcome 6. Concepts Related Theory 7. Algorithm 8. Test Cases & Troubleshooting 9. Conclusion/Analysis

Prerequisites: Basic knowledge about Algorithms and any programming knowledge.

Concepts related Theory

Encryption:

Encryption is the process of converting normal message (plaintext) into meaningless message (Ciphertext).
Encryption is the process which takes place at the sender's end.
Any message can be encrypted with either secret key or public key.
In encryption process, sender sends the data to receiver after encrypted it.

Decryption:

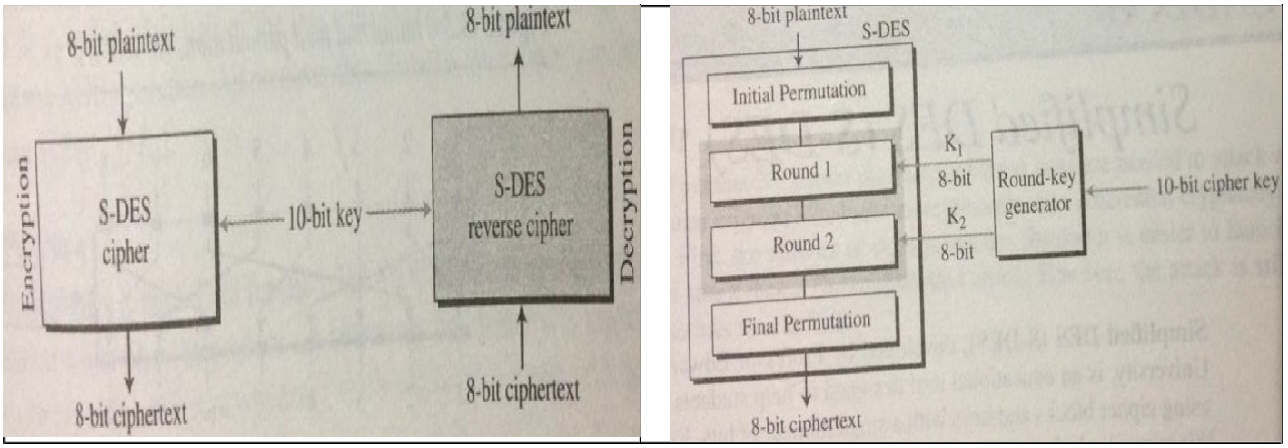
Decryption is the process of converting meaningless message (Ciphertext) into its original form (Plaintext).
Decryption is the process which takes place at the receiver's end.
Encrypted messages can be decrypted with either secret key or private key.
In the decryption process, receiver receives the information(Cipher text) and convert into plain text.

S – DES:

DES stands for Data Encryption Standard. There are certain machines that can be used to crack the DES algorithm.

Simplified DES (S – DES), developed by Professor Edward Schaefer of Santa Clara University, is an educational tool designed to help students learn the structure of DES using cipher blocks & keys with a small number of bits.

i. It is a block cipher
ii. It has 8–bits block size of plain text or cipher text
iii. It uses 10–bits key size for encryption
iv. It is a symmetric cipher
v. It has two round

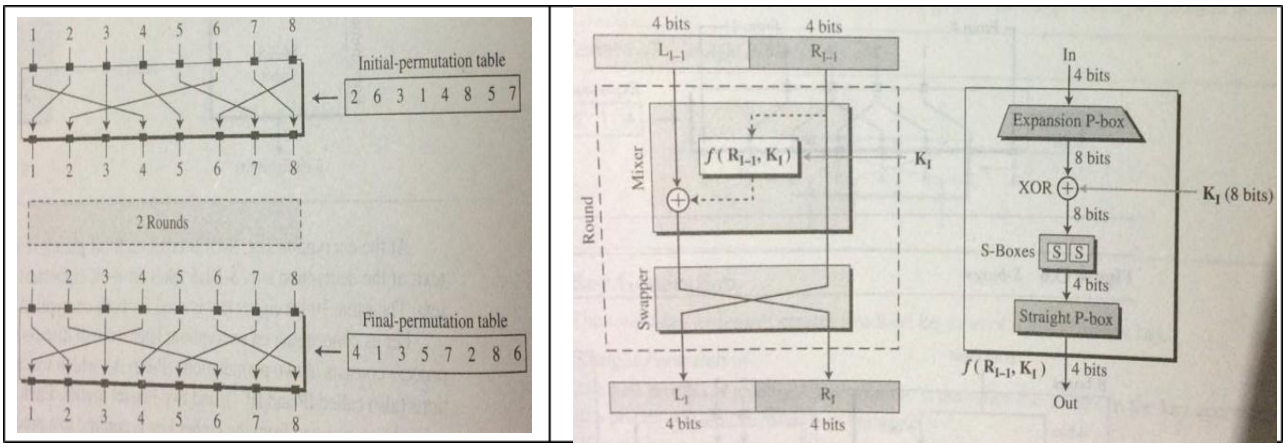


Steps:

1) Key generation
2) Encryption
3) Switch function
4) Decryption

Encryption algorithm involves five functions:

i. Initial permutation (IP)
ii. complex function f_K
iii. Simple permutation function that switches (SW) the two halves of the data
iv. function f_K again
v. inverse of initial permutation IP^{-1}



Encryption expressed as a composition function:
IP^{-1} o f_{K2} o SW o f_{K1} o IP

also written as
Ciphertext = IP^{-1} (f_{K2} (SW (f_{K1} (IP (plaintext)))))

where
K1 = P8 (Shift (P10 (Key)))
K2 = P8 (shift (shift (P10 (Key))))

Decryption:
Plaintext = IP^{-1} (f_{K1} (SW (f_{K2} (IP (ciphertext)))))

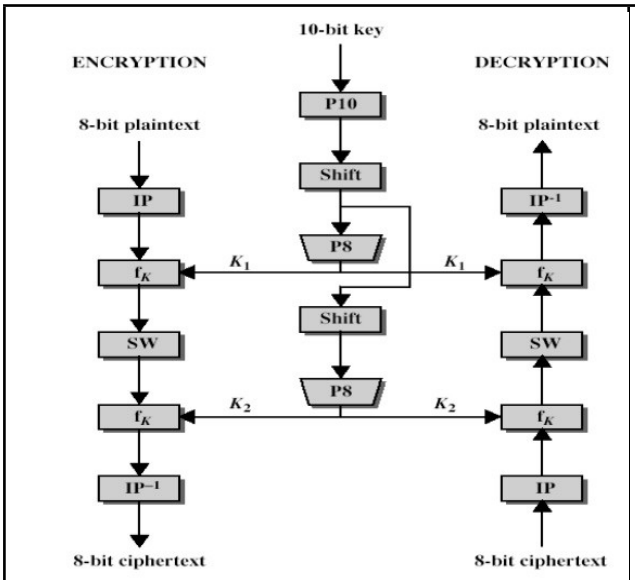
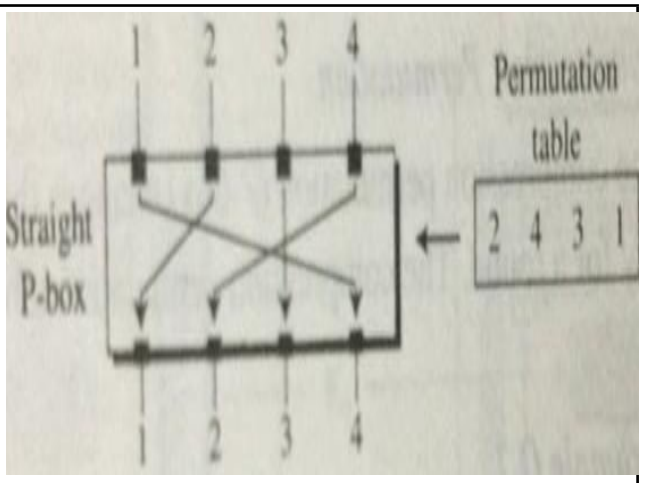
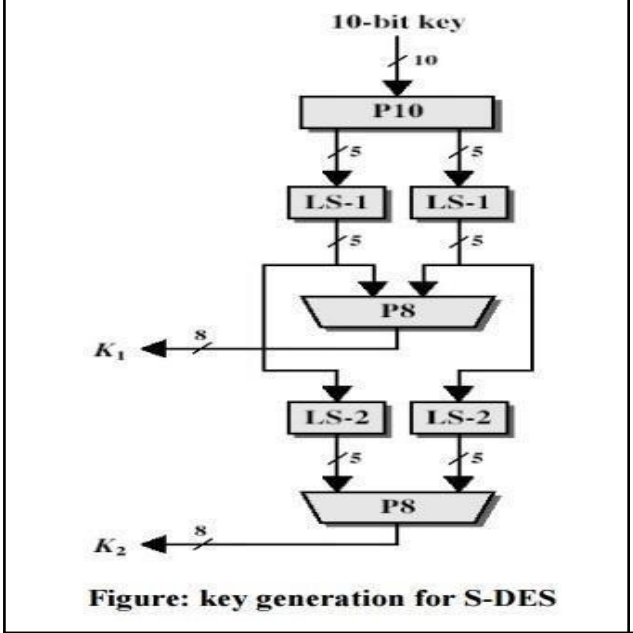
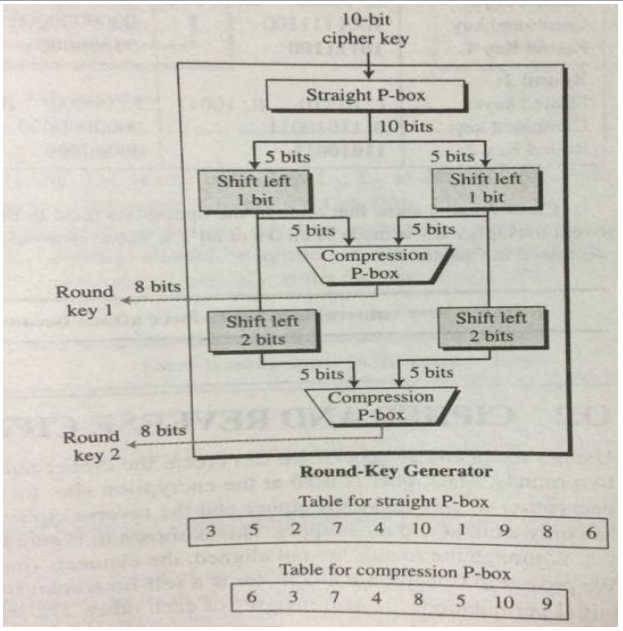
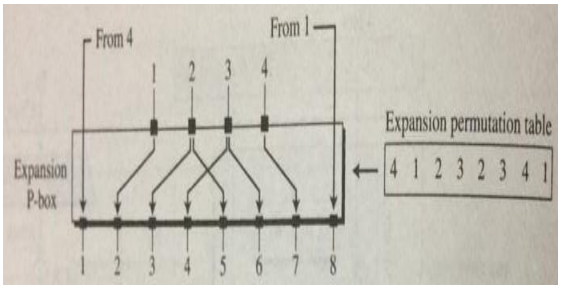
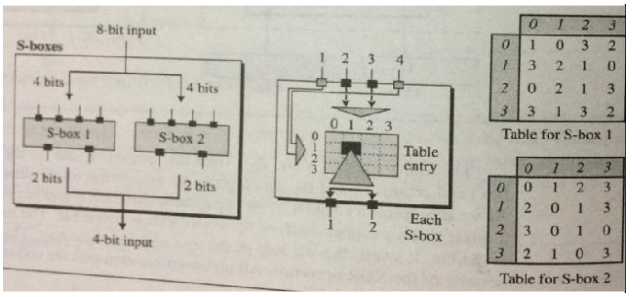
<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td> 

Figure: key generation for S-DES</td>
<td> 
</td>
</tr>
<tr>
<td>Expansion P–box</td>
<td>S–boxes</td>
</tr>
</table>

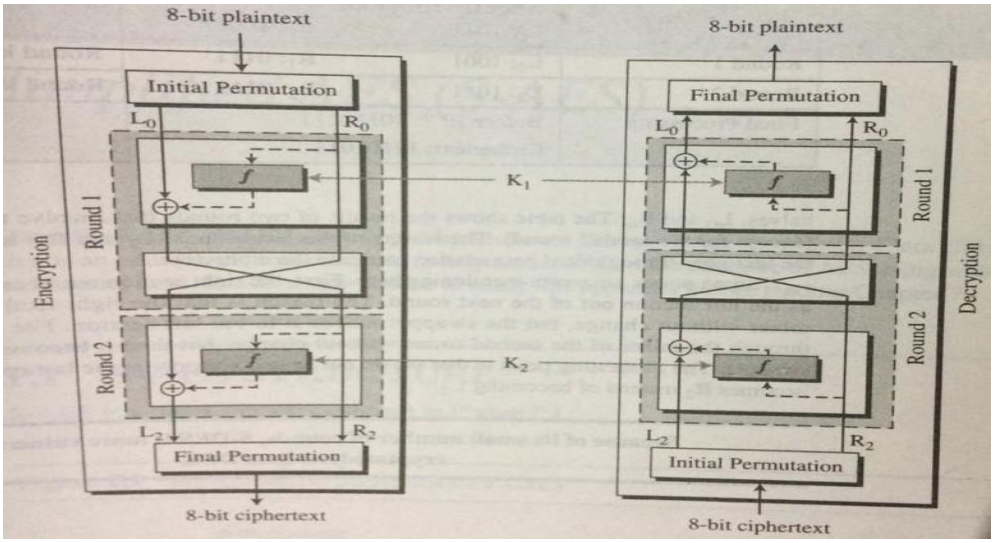




The exact realization of a Feistal network depends on the choice of following parameters & design features:

- Block size: increasing size improves security, but slows cipher
- Key size: increasing size improves security, makes exhaustive key searching harder, but may slow cipher
- Number of rounds: increasing number improves security, but slows cipher
- Subkey generation & round function: Greater complexity can make analysis harder, but slows cipher
- Fast software en/decryption & ease of analysis: are more recent concerns for practical use & testing.

S – DES Cipher & reverse Cipher



S–DES is very vulnerable to brute – force attack because of its key size (10– bits) Because of its small number of rounds, S–DES is more vulnerable to cryptanalysis that DES

None of the operations used in the key generation process is effective if the cipher key is made of all 0's or all 1's; ▙ these types of cipher keys need to be avoided.

Steps	*Case 1*	*Case 2*	*Case 3*
Cipher Key	**1011100110**	**0000000000**	**1111111111**
After permutation	1100101110	0000000000	1111111111
After splitting	L: 11001 R: 01110	L: 00000 R: 00000	L: 11111 R: 11111
Round 1:			
Shifted keys:	L: 10011 R: 11100	L: 00000 R: 00000	L: 11111 R: 11111
Combined key:	1001111100	0000000000	1111111111
Round Key 1:	**10111100**	**00000000**	**11111111**
Round 2:			
Shifted keys:	L: 01110 R: 10011	L: 00000 R: 00000	L: 11111 R: 11111
Combined key:	0111010011	0000000000	1111111111
Round Key 2:	**11010011**	**00000000**	**11111111**

Conclusion:
Thus, we have understood the implementation of DES algorithm.

Review Questions:

1. What is encryption & decryption?
2. Explain DES.

ASSIGNMENT NO	4 (Information Security)
TITLE	Implementation of AES
PROBLEM STATEMENT/ DEFINITION	Write a Java/C/C++/Python program to implement AES Algorithm.
OBJECTIVE	To understand how AES works
OUTCOME	Understanding and implementation of simplified AES algorithm
S/W PACKAGES AND HARDWARE APPARATUS USED	Core 2 DUO/i3/i5/i7 64-bitprocessor OS-LINUX 64 bit OS Editor-gedit/Eclipse S/W- C++/JAVA//Python
REFERENCES	1. Bernard Menezes, "Network Security and Cryptography", Cengage Learning India, 2014, ISBN No.: 8131513491 2. Nina Godbole, Sunit Belapure, "Cyber Security", Wiley India, 2014, ISBN No.: 978-81-345-2179-1 3. Atul Kahate, "Cryptography and Network Security", Mc Graw Hill Publication, 2nd Edition, 2008, ISBN: 978-0-07-064823- 4. William Stallings, "Cryptography and network security Principles and practices", Pearson, 6th Edition, ISBN: 978-93-325-1877-3 5. Forouzan, "Cryptography and Network Security (SIE)", Mc Graw Hill , ISBN 007070208X, 9780070702080 6.

STEPS	• Round 1
	  • Inverse Shift Row
	  • Inverse Nibble Sub
	  • Add Round Key 1
	  • Inverse Mix Columns
	• Round 2
	  • Inverse Shift Row
	  • Inverse Nibble Sub
	  • Add Round Key 0

INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment No. 3. Problem Definition 4. Learning Objective 5. Learning Outcome 6. Concepts Related Theory 7. Algorithm 8. Test Cases 9. Conclusion/Analysis

Pr-requisites: Number theory, Discrete mathematics and any programming language C++/Java/Python.

Theory:

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.

The features of AES are as follows −

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES
- Provide full specification and design details
- Software implementable in C and Java

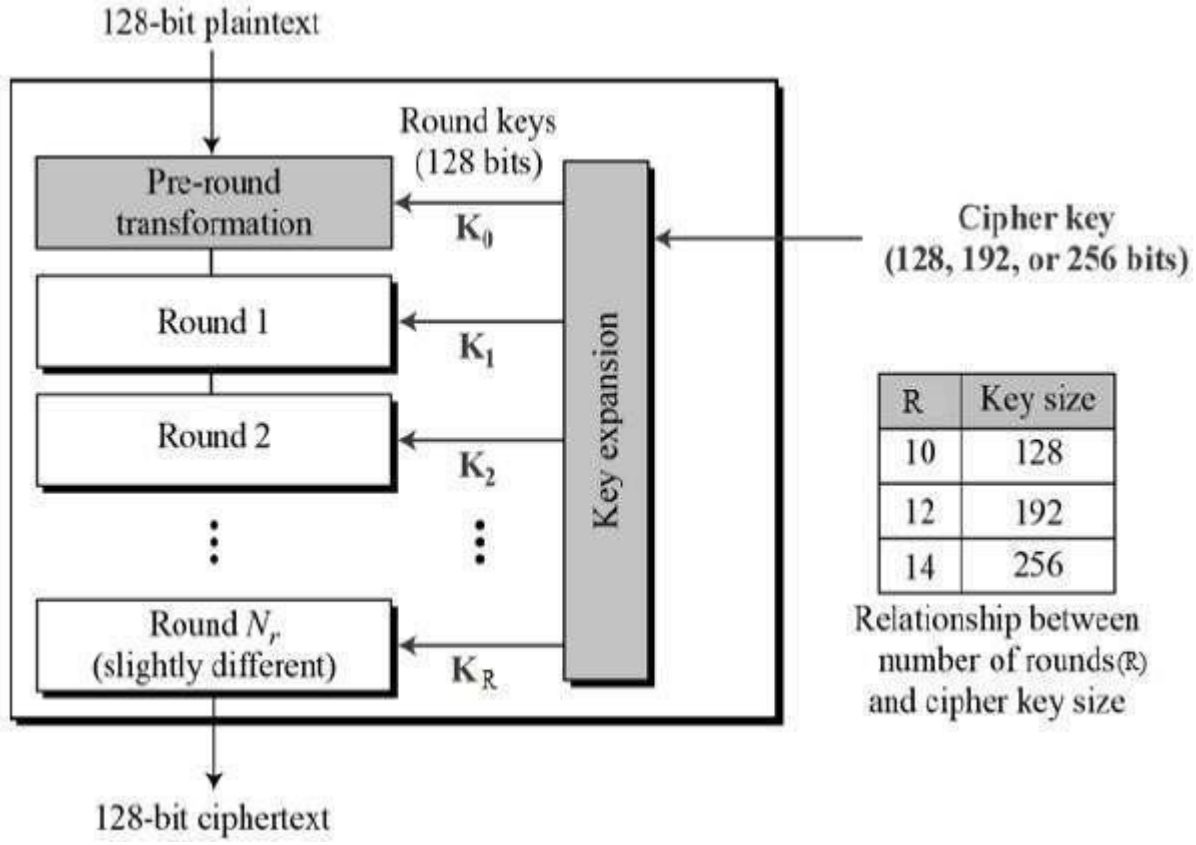
Operation of AES

AES is an iterative rather than Feistel cipher. It is based on 'substitution–permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix −

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

The schematic of AES structure is given in the following illustration −

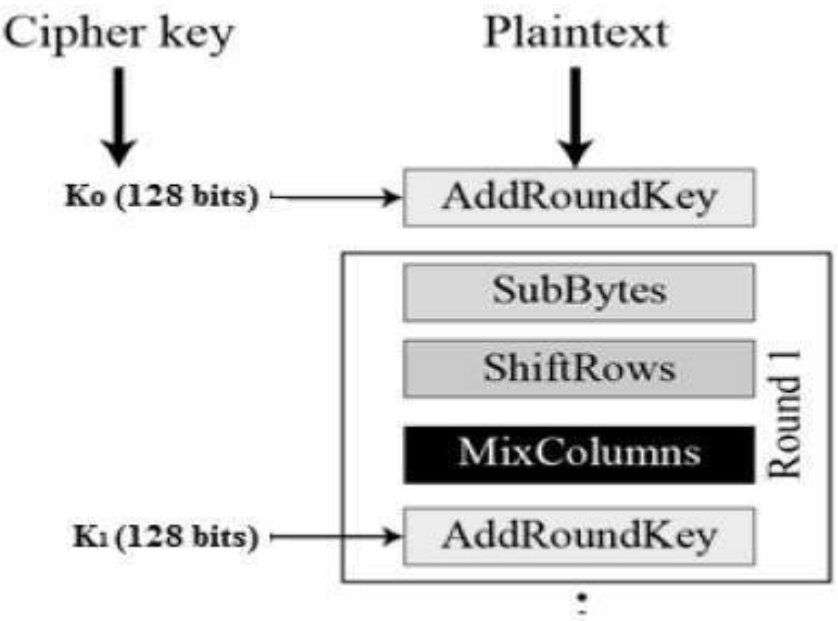


R	Key size
10	128
12	192
14	256

Relationship between number of rounds (R) and cipher key size

Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprises four sub-processes. The first round process is depicted below −



Byte Substitution

(SubBytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

Shiftrows

Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row. Shift is carried out as follows −

- First row is not shifted.
- Second row is shifted one (byte) position to the left.
- Third row is shifted two positions to the left.
- Fourth row is shifted three positions to the left.
- The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

MixColumns

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

Addroundkey

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

Decryption Process

The process of decryption of an AES ciphertext is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order −

- Add round key
- Mix columns
- Shift rows
- Byte substitution

Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms needs to be separately implemented, although they are very closely related.

AES Analysis

In present day cryptography, AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES has been discovered. Additionally, AES has built-in flexibility of key length, which allows a degree of 'future-proofing' against progress in the ability to perform exhaustive key searches.

However, just as for DES, the AES security is assured only if it is correctly implemented and good key management is employed.

Conclusion:

Hence, we have learned the workings of the AES algorithm.

Review Questions:

1. Explain the AES algorithm in detail.
2. What is a Private Key?
3. What is a Public Key?

ASSIGNMENT NO	5 (Information Security)
TITLE	Implementation of RSA
PROBLEM STATEMENT/ DEFINITION	Write a Java/C/C++/Python program to implement RSA algorithm.
OBJECTIVE	To understand how RSA algorithm works
OUTCOME	Understanding and implementation of asymmetric encryption using RSA.
S/W PACKAGES AND HARDWARE APPARATUS USED	Core 2 DUO/i3/i5/i7 64-bit processor OS-LINUX 64 bit OS Editor-gedit/Eclipse S/W- C++/JAVA//Python
REFERENCES	1. Bernard Menezes, "Network Security and Cryptography", Cengage Learning India, 2014, ISBN No.: 8131513491 2. Nina Godbole, Sunit Belapure, "Cyber Security", Wiley India, 2014, ISBN No.: 978-81-345-2179-1 3. Atul Kahate, "Cryptography and Network Security", Mc Graw Hill Publication, 2nd Edition, 2008, ISBN: 978-0-07-064823-4 4. William Stallings, "Cryptography and network security principles and practices", Pearson, 6th Edition, ISBN: 978-93-325-1877-3 5. Forouzan, "Cryptography and Network Security (SIE)", Mc Graw Hill,ISBN, 007070208X, 9780070702080
STEPS	1.Key generation 2.Encryption 3. Decryption
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment No. 3. Problem Definition

	4. Learning Objective 5. Concepts Related Theory 6. Algorithm 7. Test Cases 8. Conclusion/Analysis

Prerequisites:

Discrete mathematics, any programming language Java/C++/Python.

Concepts Related Theory:

RSA algorithm involves three steps

1. Key Generation
2. Encryption
3. Decryption

ASSIGNMENT NO	6 (Information Security)
TITLE	Implementation of Diffie-Hellman key exchange
PROBLEM STATEMENT/ DEFINITION	Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob).
OBJECTIVE	To understand how Diffie-Hellman key exchange algorithm works
OUTCOME	Understaning and implementation of key distribution algorithm
S/W PACKAGES AND HARDWARE APPARATUS USED	Core 2 DUO/i3/i5/i7 64-bit processor OS-LINUX 64 bit OS Editor-gedit/Eclipse S/W- C++/JAVA//Python
REFERENCES	1. Bernard Menezes, "Network Security and Cryptography", Cengage Learning India, 2014, ISBN No.: 8131513491 2. Nina Godbole, Sunit Belapure, "Cyber Security", Wiley India, 2014, ISBN No.: 978-81-345-2179-1 3. Atul Kahate, "Cryptography and Network Security", Mc Graw Hill Publication, 2nd Edition, 2008, ISBN: 978-0-07-064823-4 4. William Stallings, "Cryptography and network security principles and practices", Pearson, 6th Edition, ISBN: 978-93-325-1877-3 5. Forouzan, "Cryptography and Network Security (SIE)", Mc Graw Hill, ISBN, 007070208X, 9780070702080

STEPS	**1. Global Public Elements** q ; prime number α ; α < q and it is primitive root of q **2. USER A KEY GENERATION** Select Private key X_A $X_A<q$ Calculation of Public key Y_A $Y_A=a^{XA} \bmod q$ **3. USER B KEY GENERATION** Select Private key X_B $X_B<q$ Calculation of Public key Y_B $Y_B=a^{XB} \bmod q$ **4. Calculation of Secret Key by A** $k=(Y_B)^{XA} \bmod q$ **5. Calculation of Secret Key by B** $k=(Y_A)^{XB} \bmod q$
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment No. 3. Problem Definition 4. Learning Objective 5. Learning Outcome 6. Concepts Related Theory 7. Algorithm 8. Test Cases 9. Conclusion/Analysis

Pr-requisites: Discrete mathematics and any programming language C++/Java/Python.

Theory:

Diffie Hellman (DH) key exchange algorithm is a method for securely exchanging cryptographic keys over a public communications channel. Keys are not actually exchanged – they are jointly derived. It is named after their inventors Whitfield Diffie and Martin Hellman.

Working of Diffie-Hellman Algorithm:

1. In Public key encryption schemes are secure only if authenticity of the public key is assured.

2. Diffie-Hellman key exchange is a simple public key algorithm.

3. The protocol enables 2 users to establish a secret key using a public key scheme based on discrete algorithms.

4. The protocol is secure only if the authenticity of the 2 participants can be established.

5. There are 2 publicly known numbers :A prime number q and a*n integer α that is a primitive root of q.*

Note:*Premitive root of a prime number P is one, whose powers module P generate all the images from 1 to P-1*

For example:

2 is a primitive root mod 5, because for every number a relatively prime to 5, there is an integer z such that 2z≡a.
All the numbers relatively prime to 5 are 1, 2, 3, 4, and each of these (mod 5) is itself (for instance 2 (mod 5) = 2):

- 20=1, 1 (mod 5)=1, so 20≡1
- 21=2, 2 (mod 5)=2, so 21≡2
- 23=8, 8 (mod 5)=3, so 23≡3
- 22=4, 4 (mod 5)=4, so 22≡4.

4 is not a primitive root mod 5, because for every number relatively prime to 5 (again, 1, 2, 3, 4) there is not a power of 4 that is congruent. Powers of 4 (mod 5) are only congruent to 1 or 4.
There is no power of 4 that is congruent to 2 or 3:

- 40=1, 1(mod5)=1
- 41=4, 4(mod5)=4
- 42=16, 16(mod5)=1
- 43=64, 64(mod5)=4

6. Suppose users A and B wish to exchange the key.

User A selects a random integer $X_A < q$ and

computes $Y_A = a^{XA} \bmod q$

7. User B independently selects a random integer $XB < q$

and compute $Y_B = a^{XB} \bmod q$

8. Each side keeps X value private and makes Y value available publicly to the other side

user A computes the key as:

$k=(Y_B)^{X_A} \bmod q$

User B computes the key as :

$k=(Y_A)^{X_B} \bmod q$

The calculations produce identical results :

$k=(Y_B)^{X_A} \bmod q$–>calculated by user A

$=(a^{X_B} \bmod q)^{X_A} \bmod q=(a^{X_B})^{X_A} \pmod q$–> By rules of modular arithmetic$=a^{X_BX_A} \bmod q$

$= (a^{X_A})^{X_B} \bmod q$

$k=(a^{X_A} \bmod q)^{X_B} \bmod q$

9. Diffie Hellman key Exchange Algorithm

$k=(Y_A)^{X_B} \bmod q$ -> same as calculated by B

Global Public Elements

q ; prime number

α ; α < q and it is primitive root of q

USER A KEY GENERATION

Select Private key X_A $X_A<q$

Calculation of Public key Y_A $Y_A=a^{X_A} \bmod q$

USER B KEY GENERATION

Select Private key X_B $X_B<q$

Calculation of Public key Y_B $Y_B=a^{X_B} \bmod q$

Calculation of Secret Key by A

$k=(Y_B)^{X_A} \bmod q$

Calculation of Secret Key by B

$k=(Y_A)^{X_B} \bmod q$

10. The result is that two sides have exchanged a secret value.

11.Since X_A and X_B are private the other party can work only following ingredients:

q, a, X_A , X_B

Note: $YB=a^{XB}$ mod a

$X_B = d \log a, q(YB)$

↑

Discrete
Logarithm

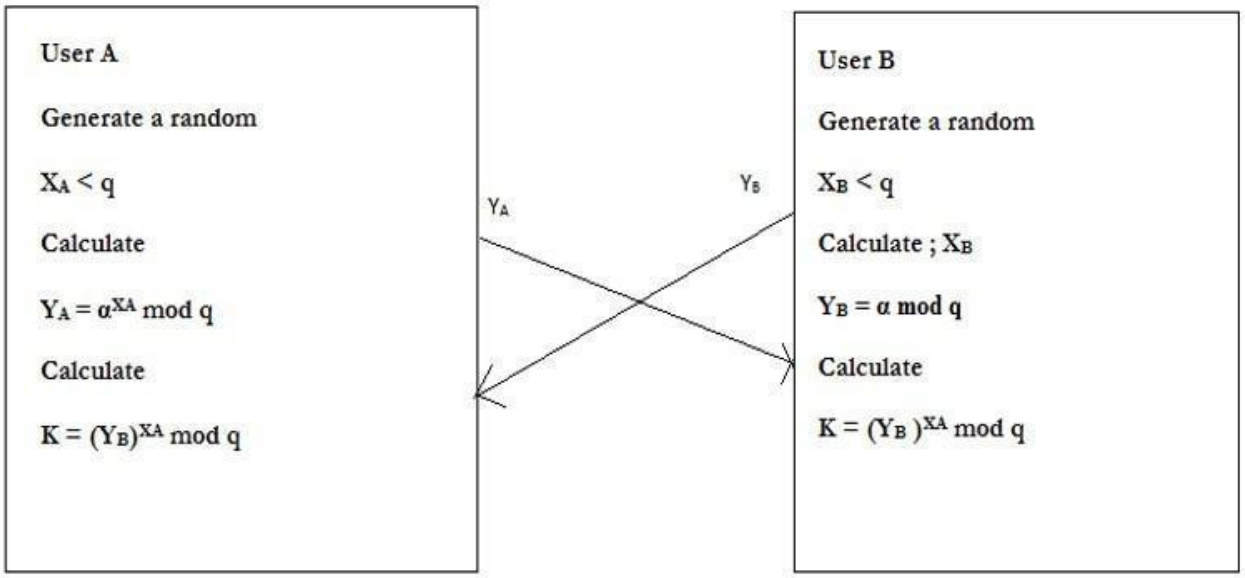
Discrete Logarithms:

Discrete logarithms are logarithms defined with regard to multiplicative cyclic groups. If G is a multiplicative cyclic group and g is a generator of G, then from the definition of cyclic groups, we know every element h in G can be written as g^x for some x. The discrete logarithm to the base g of h in the group G is defined to be x . For example, if the group is Z_5^*, and the generator is 2, then the discrete logarithm of 1 is 4 because $2^4 \equiv 1 \bmod 5$.

The discrete logarithm problem is defined as: given a group G, a generator g of the group and an element h of G, to find the discrete logarithm to the base g of h in the group G. Discrete logarithm problem is not always hard. The hardness of finding discrete logarithms depends on the groups. For example, a popular choice of groups for discrete logarithm based crypto-systems is Z_p^* where p is a prime number. However, if $p-1$ is a product of small primes, then the Pohlig– Hellman algorithm can solve the discrete logarithm problem in this group very efficiently. That's why we always want p to be a safe prime when using Z_p^* as the basis of discrete logarithm based crypto-systems. A safe prime is a prime number which equals $2q+1$ where q is a large prime number. This guarantees that $p-1 = 2q$ has a large prime factor so that the Pohlig–Hellman algorithm cannot solve the discrete logarithm problem easily. Even p is a safe prime, there is a sub-exponential algorithm which is called the index calculus. That means p must be very large (usually at least 1024-bit) to make the crypto-systems safe.

12. The algorithm security lies on the fact that it is easy to calculate exponential modulo a prime, last difficult to calculate to calculate discrete logarithm.

Figure: D-H Key exchange algorithms



Example:
Consider q=353, α= 3 (3 is primitive root of
353) A and B discrete private keys

XA=97*and XB*=223

Each computes its public key

A computes $YA=3^{97} \bmod 353 = 40$

B computes $YB=3^{233} \bmod 353 = 248$

After exchange of public keys, each can compute the common secret

key A computes $K = (YB)^{XA} \ mod\ 353 = (248)^{97} \ mod\ 353 = 160$

B computes $K = (YA)^{XB} \ mod\ 353 = (40)^{253} \ mod\ 353 = 160$

Uses of Diffie Hellman Algorithm

Aside from using the algorithm for generating public keys, there are some other places where DH Algorithm can be used:

- **Encryption:** Diffie Hellman key exchange algorithm can be used to do encryption, one of the first schemes to do it was ElGamal encryption. One modern example of it is called
  Integrated Encryption Scheme which provides security against chosen plain text and chosen clipboard attacks.

- **Password Authenticated Agreement:** When two parties share a password, a password- authenticated key agreement can be used to prevent the Man in the middle attack. This
  key Agreement can be in the form of Diffie-Hellman. Secure Remote Password Protocol is a good example that is based on this technique.
- **Forward Secrecy:** Forward secrecy based protocols can generate new key pairs for each new session, and they can automatically discard them at the end when the session is finished too. In these forward Secrecy protocols, more often than not, the Diffie Hellman key exchange is used.

Advantages of the Diffie Hellman Algorithm

- The sender and receiver don't need any prior knowledge of each other.
- Once the keys are exchanged, the communication of data can be done through an insecure
  channel.
- The sharing of the secret key is safe.

Disadvantages of the Diffie Hellman Algorithm

- The algorithm can not be used for any asymmetric key exchange.
- Similarly, it can not be used for signing digital signatures.
- Since it doesn't authenticate any party in the transmission, the Diffie Hellman key exchange is susceptible to a man-in-the-middle attack.

Test Cases: Observe the output for different values of q, a, X_A , X_B *respectively.*

Conclusion:

Hence, we have successfully learned the implementation of Diffie-Hellman Key Exchange mechanism.

Review Questions:

1. Differentiate between encryption & decryption.
2. Explain Diffie-Hellman Key Exchange mechanism in detail.
3. What is a public key?

ASSINGMENT NO.	7 (Information Security)
TITLE	Message digest of a text using the MD5 algorithm
PROBLEM STATEMENT /DEFINITION	Calculate the message digest of a text using the MD5 algorithm in JAVA.
OBJECTIVE	To understand the use of the MD5 algorithm.
OUTCOME	Implementation of MD5 algorithm.
S/W PACKAGES AND HARDWARE APPARATUS USED	1. Intel based Desktop PC: - RAM of 512 MB 2. Notepad/Notepad ++ editor 3.Net beans / Eclipse
REFERENCES	1. Bernard Menezes, "Network Security and Cryptography", Cengage Learning India, 2014, ISBN No.: 8131513491 2. Nina Godbole, Sunit Belapure, "Cyber Security", Wiley India, 2014, ISBN No.: 978-81-345-2179-1 3. Atul Kahate, "Cryptography and Network Security", Mc Graw Hill Publication, 2nd Edition, 2008, ISBN: 978-0-07-064823- 4. William Stallings, "Cryptography and network security Principles and practices", Pearson, 6th Edition, ISBN: 978-93-325-1877-3 5. Forouzan, "Cryptography and Network Security (SIE)", Mc Graw Hill , ISBN 007070208X, 9780070702080
STEPS	Step1: Append Padding Bits Step 2: Append Length Step 3: Initialize MD buffer. Step 4: Processing message in 16-word block
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

Prerequisites:

Concepts related Theory:

The MD5 hashing algorithm is a one-way cryptographic function that accepts a message of any length as input and returns as output a fixed-length digest value to be used for authenticating the original message.

The MD5 hash function was originally designed for use as a secure cryptographic hash algorithm for authenticating digital signatures(to validate the authenticity and integrity of a message, software or digital document).

MD5 is used for storing secure passwords in database servers.

MD5 generated a message digest of 128 bits.

Algorithm:

Step1: Append Padding Bits

- Padding means adding extra bits to the original message. So in MD5 original message is padded such that its length in bits is congruent to 448 modulo 512. Padding is done such that the total bits are 64 less, being a multiple of 512 bits length.
- Padding is done even if the length of the original message is already congruent to 448 modulo 512. In padding bits, the only first bit is 1, and the rest of the bits are 0.

Step 2: Append Length

After padding, 64 bits are inserted at the end, which is used to record the original input length. Modulo 2^64. At this point, the resulting message has a length multiple of 512 bits.

Step 3: Initialize MD buffer.

A four-word buffer (A, B, C, D) is used to compute the values for the message digest. Here A, B, C, D are 32- bit registers and are initialized in the following way

Word A	01	23	45	67

Word B	89	Ab	Cd	Ef
Word C	Fe	Dc	Ba	98
Word D	76	54	32	10

Step 4: Processing message in 16-word block

MD5 uses the auxiliary functions, which take the input as three 32-bit numbers and produce 32-bit output. These functions use logical operators like OR, XOR, NOR.

F(X, Y, Z)	XY v not (X)Z
G(X, Y, Z)	XZ v Y not (Z)
H(X, Y, Z)	X xor Y xor Z
I(X, Y, Z)	Y xor (X v not (Z))

The content of four buffers are mixed with the input using this auxiliary buffer, and 16 rounds are performed using 16 basic operations.

Conclusion:

Hence, we have learned the implementation of message digest of a text using the MD5 algorithm

Review Questions:

1. What is the MD5 algorithm?
2. What is Cryptography?
3. What is Java?