Sinhgad Institutes

NBN Sinhgad School of Engineering, Pune

Laboratory Practice-II

Department of Computer Engineering

TE(Computer) **Academic Year: 2023-24**

Prepared By:

Ms.A.M.Chadchankar
(Assistant Professor, Department of Computer Engineering)

SINHGAD TECHNICAL EDUCATION SOCIETY'S

NBN SINHGAD SCHOOL OF ENGINEERING

Pune

Department of Computer Engineering

LABORATORY MANUAL

AY: 2022-23

LABORATORY PRACTICE -II

T.E. COMPUTER ENGINEERING

SEMESTER-II

TEACHING SCHEME	CREDIT	EXAMINATION SCHEME
Practical: 4 Hrs / Week	PR: 02	University Term work: 50 Marks
		Practical: 25 Marks

Prepared By:

Ms.A.M.Chadchankar

(Assistant Professor, Department of Computer Engineering)

PREFACE

Laboratory Practice -II

Operating System recommended: 64-bit Open source Linux or its derivative

Programming Languages: C++/JAVA/PYTHON/R

Programming tools recommended: Front End: Java/Perl/PHP/Python/Ruby/.net,

Backend: Mongo DB/MYSQL/Oracle.

Database Connectivity: ODBC/JDBC, Additional Tools: Octave, Matlab, WEKA**.**

MAIN OBJECTIVES OF LAB:

Artificial Intelligence –

1. Design system using different informed search / uninformed search or heuristic approaches
2. Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning
3. Design and develop an expert system

Information Technology -

1. Use tools and techniques in the area of Information Security.
2. Use the knowledge of security for problem solving.
3. Apply the concepts of Information Security to design and develop applications.

CERTIFICATE

This is to certify that Mr. /Miss___________________________________of class TE Roll No. ___________________Examination Seat No.__________________________has completed all the practical work in the Laboratory Practice-II satisfactorily, as prescribed by Savitribai Phule Pune University in the Academic Year 2023-2024.

Staff In-charge **Head of Department** **Principal**

NBN Sinhgad School of Engineering

Department of Computer Engineering

List of Laboratory Assignments

	Artificial Intelligence
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 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
4	Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem.
5	Develop an elementary chatbot for any suitable customer interaction application.
6	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

Information Security	
7	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.
8	Write a Java/C/C++/Python program to perform encryption and decryption using the method of Transposition technique.
9	Write a Java/C/C++/Python program to implement DES algorithm.
10	Write a Java/C/C++/Python program to implement AES Algorithm.
11	Write a Java/C/C++/Python program to implement RSA algorithm

Assignment No : 1	**Date:**
Title: 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.	
Sign:	**Remark:**

Artificial Intelligence

Group A

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.**

BFS is one of the traversing algorithm used in graphs. This algorithm is implemented using a queue data structure. In this algorithm, the main focus is on the vertices of the graph. Select a starting node or vertex at first, mark the starting node or vertex as visited and store it in a queue. Then visit the vertices or nodes which are adjacent to the starting node, mark them as visited and store these vertices or nodes in a queue. Repeat this process until all the nodes or vertices are completely visited.

Advantages of BFS

1. It can be useful in order to find whether the graph has connected components or not.
2. It always finds or returns the shortest path if there is more than one path between two vertices.

Disadvantages of BFS

1. The execution time of this algorithm is very slow because the time complexity of this algorithm is exponential.
2. This algorithm is not useful when large graphs are used.

Explanation:

1. Create a graph.
2. Initialize a starting node.
3. Send the graph and initial node as parameters to the bfs function.
4. Mark the initial node as visited and push it into the queue.
5. Explore the initial node and add its neighbors to the queue and remove the initial node from the queue.
6. Check if the neighbors node of a neighboring node is already visited.
7. If not, visit the neighboring node neighbors and mark them as visited.

8. Repeat this process until all the nodes in a graph are visited and the queue becomes empty.

Output:

```
['A', 'B', 'C', 'E', 'D', 'F', 'G']
```

Implementation of BFS in Python (Breadth First Search)

Source Code I: BFS in Python

```
graph = {'A': ['B', 'C', 'E'],
'B': ['A','D', 'E'],
'C': ['A', 'F', 'G'],
'D': ['B'],
'E': ['A', 'B','D'],
'F': ['C'], 'G': ['C']} def bfs(graph, initial):
visited = [] queue = [initial] while queue:
node = queue.pop(0) if node not in visited:
visited.append(node) neighbours = graph[node]
for neighbour in neighbours:
queue.append(neighbour) return visited
print(bfs(graph,'A'))
```

Depth First Search (DFS)

Depth first Search or Depth first traversal is a recursive algorithm for searching all the vertices of a graph or tree data structure. Traversal means visiting all the nodes of a graph.

Depth First Search Algorithm

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

1. Visited
2. Not Visited

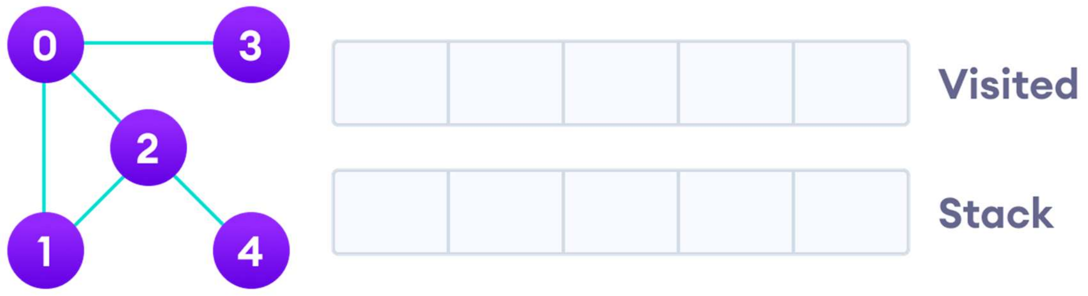
The purpose of the algorithm is to mark each vertex as visited while avoiding cycles.

The DFS algorithm works as follows:

1. Start by putting any one of the graph's vertices on top of a stack.
2. Take the top item of the stack 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 top of the stack.
4. Keep repeating steps 2 and 3 until the stack is empty.

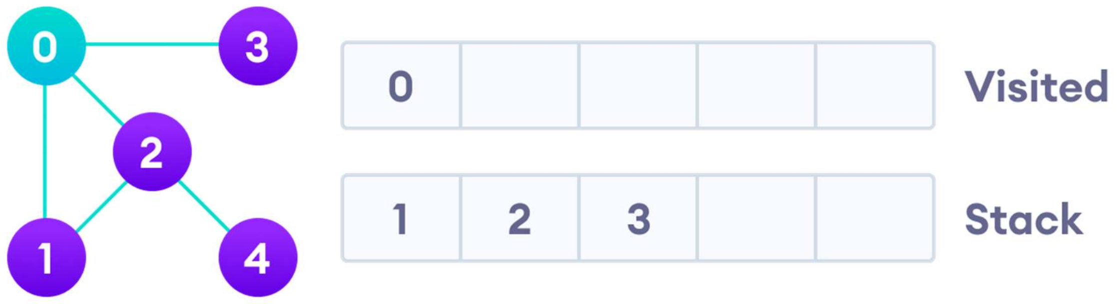
Depth First Search Example

Let's see how the Depth First Search algorithm works with an example. We use an undirected graph with 5 vertices.



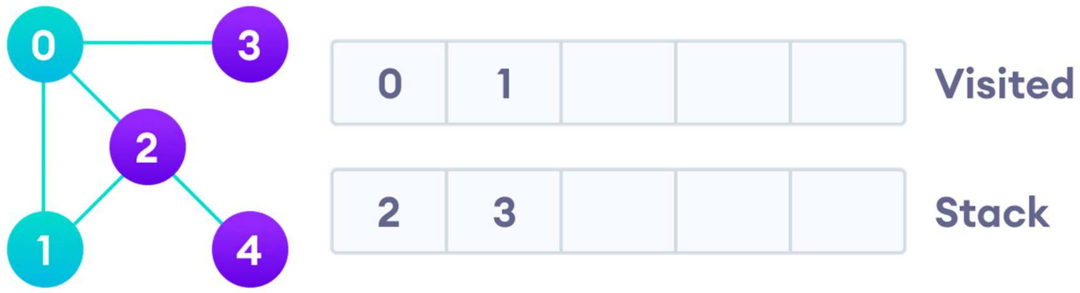
Undirected graph with 5 vertices

We start from vertex 0, the DFS algorithm starts by putting it in the Visited list and putting all its adjacent vertices in the stack.



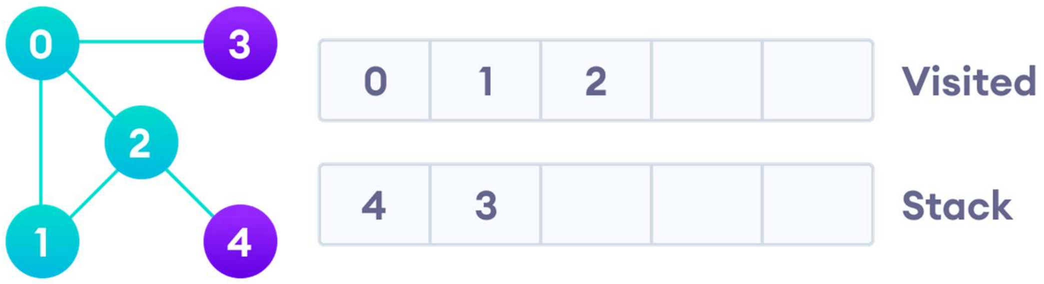
Visit the element and put it in the visited list

Next, we visit the element at the top of stack i.e. 1 and go to its adjacent nodes.
Since 0 has already been visited, we visit 2 instead.

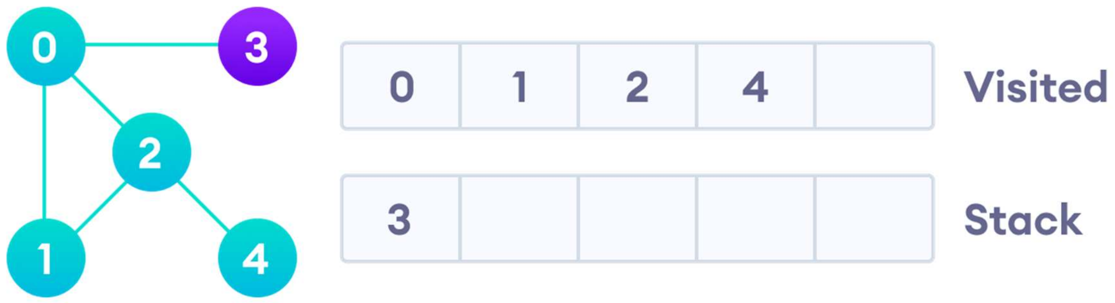


Visit the element at the top of stack

Vertex 2 has an unvisited adjacent vertex in 4, so we add that to the top of the stack and visit it.

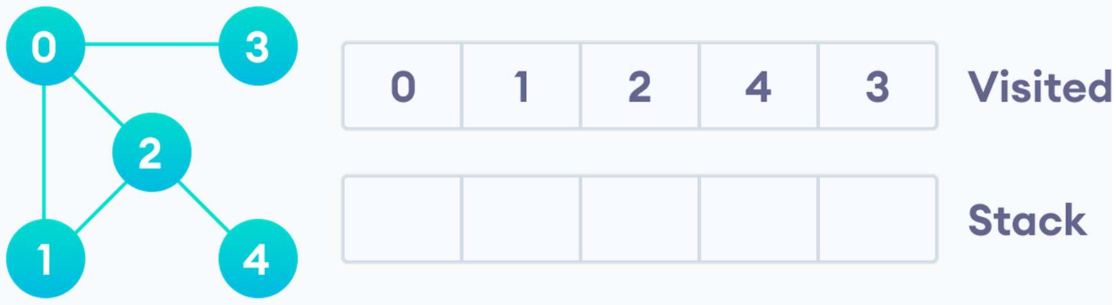


Vertex 2 has an unvisited adjacent vertex in 4, so we add that to the top of the stack and visit it.



Vertex 2 has an unvisited adjacent vertex in 4, so we add that to the top of the stack and visit it.

After we visit the last element 3, it doesn't have any unvisited adjacent nodes, so we have completed the Depth First Traversal of the graph.



After we visit the last element 3, it doesn't have any unvisited adjacent nodes, so we have completed the Depth First Traversal of the graph.

Complexity of Depth First Search

The time complexity of the DFS algorithm is represented in the form of `O(V + E)`, where `V` is the number of nodes and `E` is the number of edges.

The space complexity of the algorithm is `O(V)`.

Application of DFS Algorithm

1. For finding the path
2. To test if the graph is bipartite
3. For finding the strongly connected components of a graph
4. For detecting cycles in a graph

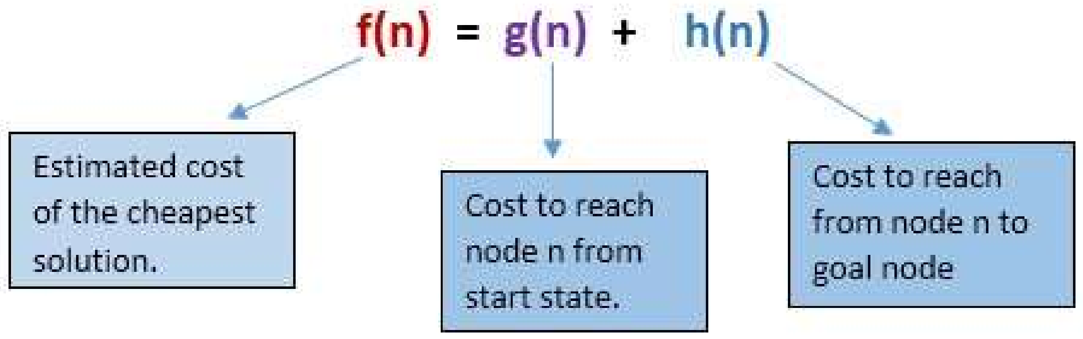
Assignment No : 2	**Date:**
Title: Implement A star Algorithm for any game search problem	
Sign:	**Remark:**

2. Implement A star Algorithm for any game search problem.

A* Search

A* search is the most commonly known form of best-first search. It uses heuristic function h(n), and cost to reach the node n from the start state g(n). It has combined features of UCS and greedy bestfirst search, by which it solve the problem efficiently. A* search algorithm finds the shortest path through the search space using the heuristic function. This search algorithm expands less search tree and provides optimal result faster. A* algorithm is similar to UCS except that it uses g(n)+h(n) instead of g(n).

In A* search algorithm, we use search heuristic as well as the cost to reach the node. Hence we can combine both costs as following, and this sum is called as a **fitness number**.


f(n) = g(n) + h(n)
Estimated cost of the cheapest solution.
Cost to reach node n from start state.
Cost to reach from node n to goal node

Algorithm of A* search:

Step1: Place the starting node in the OPEN list.

Step 2: Check if the OPEN list is empty or not, if the list is empty then return failure and stops.

Step 3: Select the node from the OPEN list which has the smallest value of evaluation function (g+h), if node n is goal node then return success and stop, otherwise

Step 4: Expand node n and generate all of its successors, and put n into the closed list. For each successor n', check whether n' is already in the OPEN or CLOSED list, if not then compute evaluation function for n' and place into Open list.

Step 5: Else if node n' is already in OPEN and CLOSED, then it should be attached to the back pointer which reflects the lowest g(n') value.

Step 6: Return to **Step 2**.

Advantages:

1. A* search algorithm is the best algorithm than other search algorithms.

2. A* search algorithm is optimal and complete.
3. This algorithm can solve very complex problems.

Disadvantages:

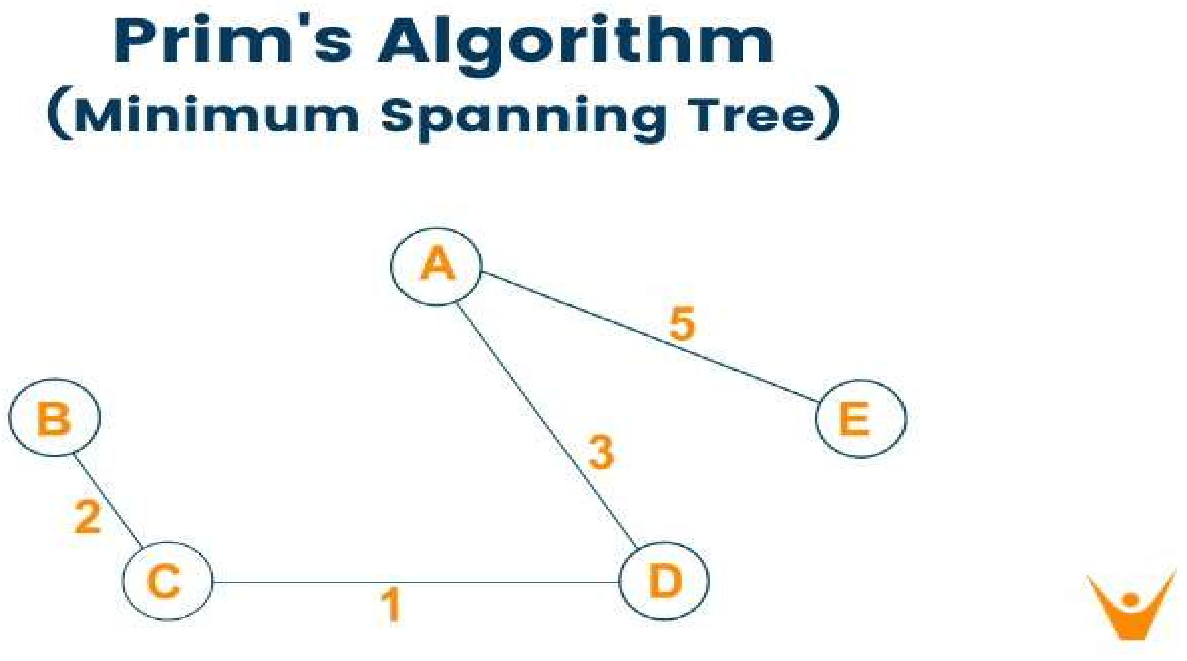
1. It does not always produce the shortest path as it mostly based on heuristics and approximation.
2. A* search algorithm has some complexity issues.
3. The main drawback of A* is memory requirement as it keeps all generated nodes in the memory, so it is not practical for various large-scale problems.

Assignment No : 3	**Date:**
Title: Implement Greedy search algorithm for any 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**	
Sign:	**Remark:**

3. Implement Greedy search algorithm for any of the following application:

- **Prim's Minimal Spanning Tree Algorithm**

Prim's Algorithm | Minimum Spanning Tree (Python Code)



We will study what is the minimum spanning tree and how to convert a graph into a minimum spanning tree using Prim’s Algorithm. We will learn the algorithm and python code for prim’s algorithm and an example for better understanding. Lastly, we will study the running time complexity and applications of prim's algorithm in real life.

What is a Minimum Spanning Tree?

As we all know, the graph which does not have edges pointing to any direction in a graph is called an undirected graph and the graph always has a path from a vertex to any other vertex. A spanning tree is a subgraph of the undirected connected graph where it includes all the nodes of the graph with the minimum possible number of edges. Remember, the subgraph should contain each and every node of the original graph. If any node is missed out then it is not a spanning tree and also, the spanning tree doesn’t contain cycles. If the graph has n number of nodes, then the total number of spanning trees created from a complete graph is equal to n^(n2). In a spanning tree, the edges may or may not have weights associated with them. Therefore, the spanning tree in which the sum of edges is minimum as possible then that spanning tree is called the minimum spanning tree. One graph can have multiple spanning-tree but it can have only one unique minimum spanning tree. There are two different ways to find out the minimum spanning tree from the complete graph i.e Kruskal's algorithm and Prim’s algorithm. Let us study prim’s algorithm in detail below:

What is Prim’s Algorithm?

Prim’s algorithm is a minimum spanning tree algorithm which helps to find out the edges of the graph to form the tree including every node with the minimum sum of weights to form the minimum spanning tree. Prim’s algorithm starts with the single source node and later explore all the adjacent nodes of the source node with all the connecting edges. While we are exploring the graphs, we will choose the edges with the minimum weight and those which cannot cause the cycles in the graph.

Prim's Algorithm for Minimum Spanning Tree

Prim's algorithm basically follows the greedy algorithm approach to find the optimal solution. To find the minimum spanning tree using prim's algorithm, we will choose a source node and keep adding the edges with the lowest weight.

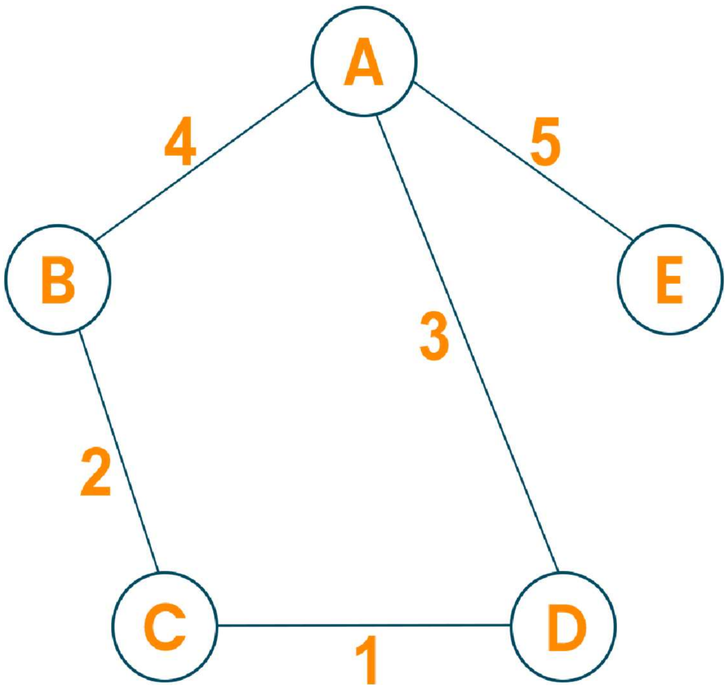
The algorithm is as given below:

- Initialize the algorithm by choosing the source vertex
- Find the minimum weight edge connected to the source node and another node and add it to the tree
- Keep repeating this process until we find the minimum spanning tree

Pseudocode

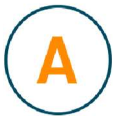
```
T = ;
M = { 1 };
while (M ≠ N)
    let (m, n) be the lowest cost edge such that m ∈ M and n ∈ N - M;
T = T ∪ {(m, n)}
    M    =    ∪ {n}
M
```

Here we create two sets of nodes i.e M and M-N. M set contains the list of nodes that have been visited and the M-N set contains the nodes that haven’t been visited. Later, we will move each node from M to M-N after each step by



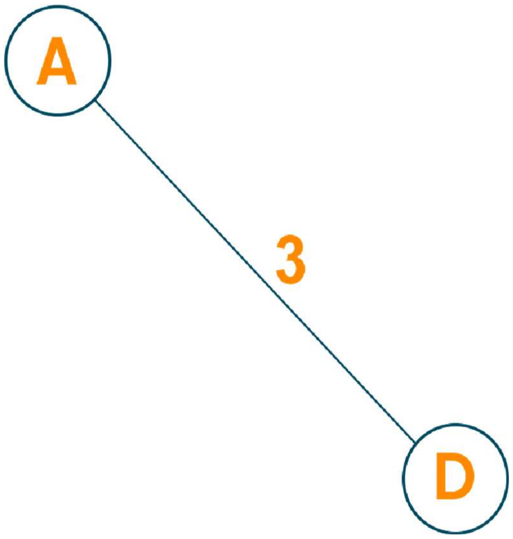
connecting the least weight edge.

Example



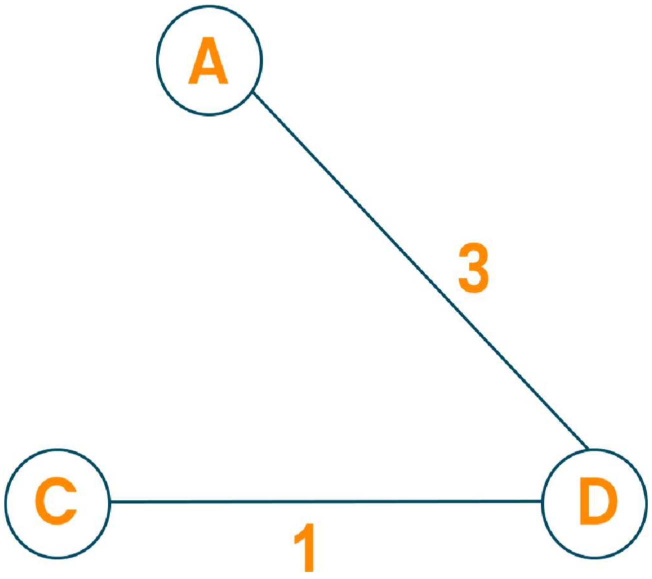
Let us consider the below-weighted graph

Later we will consider the source vertex to initialize the algorithm

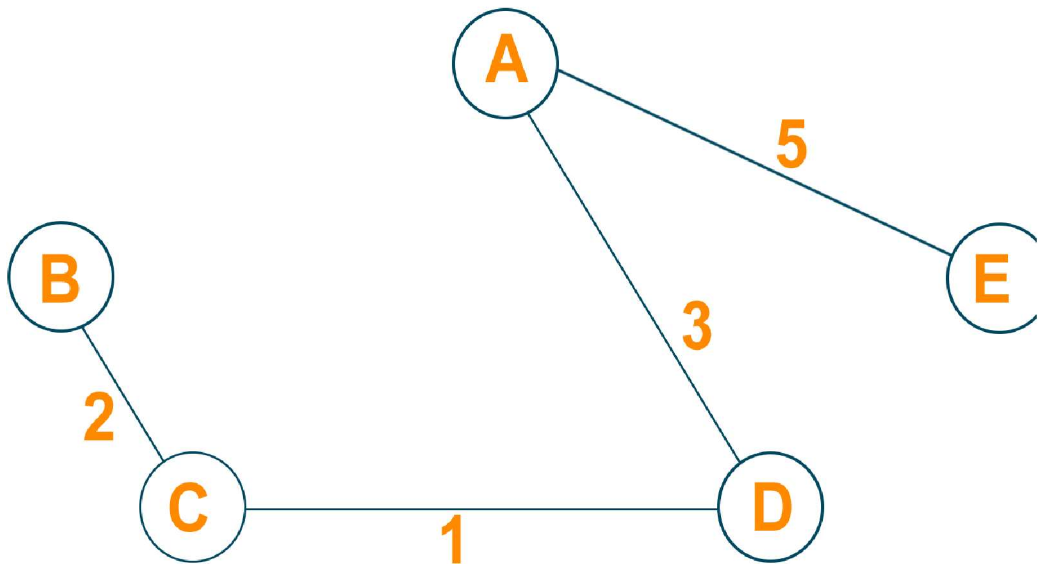


Now, we will choose the shortest weight edge from the source vertex and add it to finding the spanning tree.

Then, choose the next nearest node connected with the minimum edge and add it to the solution.



Continue the steps until all nodes are included and we find the minimum spanning tree.



Time Complexity:

The running time for prim's algorithm is O(VlogV + ElogV) which is equal to O(ElogV) because every insertion of a node in the solution takes logarithmic time. Here, E is the number of edges and V is the number of vertices/nodes. However, we can improve the running time complexity to O(E + logV) of prim's algorithm using Fibonacci Heaps.

Applications

- Prim's algorithm is used in network design
- It is used in network cycles and rail tracks connecting all the cities
- Prim's algorithm is used in laying cables of electrical wiring
- Prim's algorithm is used in irrigation channels and placing microwave towers
- It is used in cluster analysis
- Prim's algorithm is used in gaming development and cognitive science
- Pathfinding algorithms in artificial intelligence and traveling salesman problems make use of prim's algorithm.

Conclusion

As we studied, the minimum spanning tree has its own importance in the real world, it is important to learn the prim's algorithm which leads us to find the solution to many problems. When it comes to finding the minimum spanning tree for the dense graphs, prim's algorithm is the first choice.

Assignment No : 4	**Date:**
Title: Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem.	
Sign:	**Remark:**

Group-B

4. Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem

8 Queens Problem using Branch and Bound

The N-Queens problem is a puzzle of placing exactly N queens on an NxN chessboard, such that no two queens can attack each other in that configuration. Thus, no two queens can lie in the same row, column or diagonal.

The branch and bound solution is somehow different, it generates a partial solution until it figures that there's no point going deeper as we would ultimately lead to a dead end.

In the backtracking approach, we maintain an 8x8 binary matrix for keeping track of safe cells (by eliminating the unsafe cells, those that are likely to be attacked) and update it each time we place a new queen. However, it required O(n^2) time to check safe cell and update the queen.

In the 8 queens problem, we ensure the following:

1. no two queens share a row
2. no two queens share a column
3. no two queens share the same left diagonal
4. no two queens share the same right diagonal we already ensure that the queens do not share the same column by the way we fill out our auxiliary matrix (column by column). Hence, only the left out 3 conditions are left out to be satisfied.

Applying the branch and bound approach :

The branch and bound approach suggest that we create a partial solution and use it to ascertain whether we need to continue in a particular direction or not. For this problem, we create 3 arrays to check for conditions 1,3 and 4. The Boolean arrays tell which rows and diagonals are already occupied. To achieve this, we need a numbering system to specify which queen is placed.

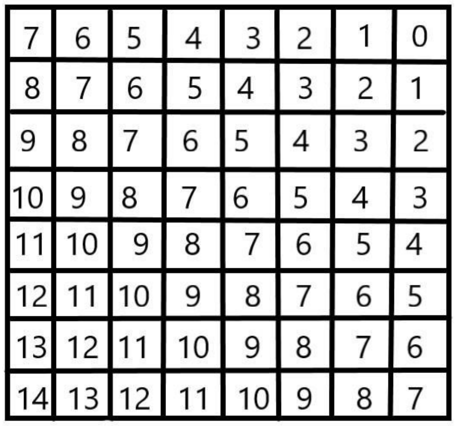
The indexes on these arrays would help us know which queen we are analyzing.

Preprocessing - create two NxN matrices, one for top-left to bottom right diagonal, and other for top-right to bottom-left diagonal. We need to fill these in such a way that two queens sharing same topleft_bottom-right diagonal will have same value in slashDiagonal and two queens sharing same top-right_bottom-left diagonal will have same value in backSlashDiagonal.

slashDiagnol(row)(col) = row + col

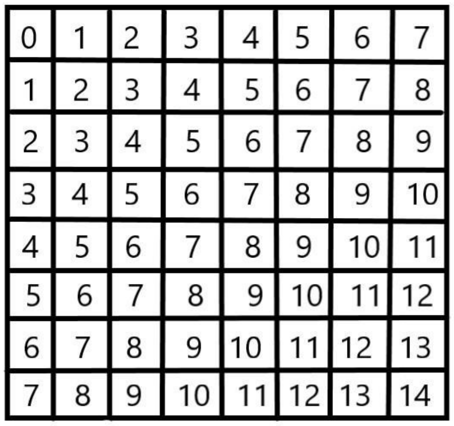
backSlashDiagnol(row)(col) = row - col + (N-1) { N = 8 }

{ we added (N-1) as we do not need negative values in backSlashDiagnol }



7	6	5	4	3	2	1	0
8	7	6	5	4	3	2	1
9	8	7	6	5	4	3	2
10	9	8	7	6	5	4	3
11	10	9	8	7	6	5	4
12	11	10	9	8	7	6	5
13	12	11	10	9	8	7	6
14	13	12	11	10	9	8	7

slash diagnol[row][col] = row + col



0	1	2	3	4	5	6	7
1	2	3	4	5	6	7	8
2	3	4	5	6	7	8	9
3	4	5	6	7	8	9	10
4	5	6	7	8	9	10	11
5	6	7	8	9	10	11	12
6	7	8	9	10	11	12	13
7	8	9	10	11	12	13	14

backslash diagnol[row][col] = row-col+(N-1)

For placing a queen *i* on row *j*, check the following :

1. whether row 'j' is used or not
2. whether slashDiagnol 'i+j' is used or not
3. whether backSlashDiagnol 'i-j+7' is used or not

If the answer to any one of the following is true, we try another location for queen **i** on row **j**, mark the row and diagonals; and recur for queen **i+1**.

Graph coloring problem’s solution using backtracking algorithm

Graph coloring

The **graph coloring problem** is to discover whether the nodes of the graph G can be covered in such a way, that no two adjacent nodes have the same color yet only m colors are used. This graph coloring problem is also known as Mcolorability decision problem.

The M – colorability optimization problem deals with the smallest integer m for which the graph G can be colored. The integer is known as a chromatic number of the graph.

Here, it can also be noticed that if d is the degree of the given graph, then it can be colored with d+ 1 color.

A graph is also known to be planar if and only if it can be drawn in a planar in such a way that no two edges cross each other. A special case is the 4 - colors problem for planar graphs. The problem is to color the region in a map in such a way that no two adjacent regions have the same color. Yet only four colors are needed. This is a problem for which graphs are very useful because a map can be easily transformed into a graph. Each region of the map becomes the node, and if two regions are adjacent, they are joined by an edge.

Graph coloring problem can also be solved using a state space tree, whereby applying a backtracking method required results are obtained.

For solving the **graph coloring problem**, we suppose that the graph is represented by its adjacency matrix G[1:n, 1:n], where, G[i, j]= 1 if (i, j) is an edge of G, and G[i, j] = 0 otherwise.

The colors are represented by the integers 1, 2, ..., m and the solutions are given by the n-tuple (x1, x2, x3, ..., xn), where x1 is the color of node i.

Algorithm for finding the m - colorings of a graph

```
1.      Algorithm mcoloring ( k )
2.      // this algorithm is formed using the recursive backtracking
3.      // schema. The graph is represented by its Boolean adjacency
4.      // matrix G [1: n, 1: n]. All assignments of 1, 2, …, m to the
5.      // vertices of the graph such that adjacent vertices are
6.      // assigned distinct are printed. K is the index
7.      // of the next vertex to color.
```

```
8.      {
9.      Repeat
10.     {
11.     // generate all legal assignments for x[k], 12.       Next value (k); //
assign to x[k] a legal color.
13.     If ( x[k] = 0 ) then return; // no new color possible
14.     If (k = n) then // at most m colors have been used to color the n vertices.
15.     Write (x[1 : n ]);
16.     Else mcoloring (k + 1);
17.     }
18.     Until (false);
19.     }
```

This algorithm uses the recursive backtracking schema. In this algorithm colors to be assigned are to determine from the range (0, m), i.e., m colors are available.

The total time required by the above algorithm is **O (nm^n)**.

Assignment No : 5	**Date:**
Title: Develop an elementary chatbot for any suitable customer interaction application.	
Sign:	**Remark:**

5. Develop an elementary chatbot for any suitable customer interaction application.

What is a chatbot?

A chatbot is a computer program designed to have a conversation with human beings over the internet. It's also known as conversational agents, which communicate and collaborate with human users, through text messaging, in order to accomplish a specific task. Basically, there are two types of chatbots. The one that uses **Artificial Intelligence**, and another one is based on multiple choice scripts.

Both types of chatbots aim to create a more personalized content experience for the users, whether that's while watching a video, reading articles or buying new shoes.

These Chatbots hold the promise of being the next generation of technology that people use to interact online with business enterprises. These Chatbots offer a lot of advantages, one of which is that, because Chatbots communicate using a natural language, users don't need to learn yet another new website interface, to get comfortable with the unavoidable quirks.

Chatbots are capable to interpret human speech, and decide which information is being sought. Artificial intelligence is getting smarter each day, and brands that are integrating Chatbots with the artificial intelligence, can deliver one-to-one individualized experiences to consumers.

Why chatbot?

Chatbots can be useful in many aspects of the customer experience, including providing customer service, presenting product recommendations and engaging customers through targeted marketing campaigns. If a customer has an issue with a product, she can connect with a chatbot to explain the situation and the chatbot can input that information to provide a recommendation of how to fix the product. On the recommendation side, chatbots can be used to share popular products with customers that they might find useful and can act as a sort of personal shopper or concierge service to find the perfect gift, meal or night out for a customer with just a few basic questions. Brands are also using chatbots to connect their customers with thought leaders and add personality to their products. In all cases, brands seem to be having great success and experiencing increased engagement and revenue.

Chatbots are easy to use and many customers prefer them over calling a representative on the phone because it tends to be faster and less invasive. They can also save money for companies and are easy to set up.

Chatbots are relatively new and most companies haven't implemented them yet, it's only natural that users are interested in them. Hence, people want to discover what chatbots can and cannot do.

The number of businesses using chatbots has grown exponentially. Chatbots have increased from

30,000 in 2016 to over 100,000 today. Every major company has announced their own chatbot and 60% of the youth population uses them daily.

These statistics prove that chatbots are the new-gen tech. No more waiting for the right time to incorporate them into your business. The time is now. By the year 2020, nearly 80% of businesses will have their own chatbot.

Billions of people are already using chatbots, so it's time your business did too.

Benefits of chatbot?

Chatbots are being made to ease the pain that the industries are facing today. The purpose of chat bots is to support and scale business teams in their relations with customers.

Chatbots may sound like a futuristic notion, but according to Global Web Index statistics, it is said that 75% of internet users are adopting one or more messenger platforms. Although research shows us that each user makes use of an average of 24 apps a month, wherein 80% of the time would be in just 5 apps. This means you can hardly shoot ahead with an app, but you still have high chances to integrate your chatbot with one of these platforms.

Now lets go through some of the benefits that chatbots provide:

1. Available 24*7:

I'm sure most of you have experienced listening to the boring music playing while you're kept on hold by a customer care agent. On an average people spend 7 minutes until they are assigned to an agent. Gone are the days of waiting for the next available operative. Bots are replacing live chat and other forms of contact such as emails and phone calls.

Since chat bots are basically virtual robots they never get tired and continue to obey your command. They will continue to operate every day throughout the year without requiring to take a break. This improves your customer satisfaction and helps you rank highly in your sector.

2. Handling Customers:

We humans are restricted to the number of things we can do at the same time. A study suggests that humans can only concentrate on 3–4 things at the same time. If it goes beyond that you are bound to meet errors.

Chatbots on the other hand can simultaneously have conversations with thousands of people. No matter what time of the day it is or how many people are contacting you, every single one of them will be answered instantly. Companies like Taco Bell and Domino's are already using chatbots to arrange delivery of parcels.

3. Helps you Save Money:

If you are a business owner you are bound have a lot of employees who need to be paid for the work they do. And these expenses just keep adding up as business grows. Chatbots are a one time investment which helps businesses reduce down on staff required.

You could integrate a customer support chatbot in your business to cater to simple queries of customers and pass on only the complex queries to customer support agents.

4. Provides 100% satisfaction to customers:

Humans react to others based on their mood and emotions. If a agent is having a good attitude or is in good mood he will most probably talk to customers in a good way. In contrary to this the customer will not be satisfied.

Whereas chatbots are bound by some rules and obey them as long as they’re programmed to. They always treat a customer in the most polite and perfect way no matter how rough the person is. Also, in the travel and hospitality industry where travelers do not speak the same language, a bot can be trained to communicate in the language of the traveler.

5. Automation of repetitive work:

Lets be honest, no one likes doing the same work again and again over brief period of time. In the case of humans, such tasks are prone to errors. Chatbots now help automate tasks which are to be done frequently and at the right time.

Also, now there are numerous slack bots which automate repetitive tasks. This helps people save time and increase productivity. For example, there are new items bought from your eCommerce site or there is a bug reported then it sends a short summary to a slack channel.

6. Personal Assistant:

People could use Bots as a fashion advisor for clothing recommendations, or ask trading tips from a finance bot, suggest places to visit from a travel bot and so forth. This would help the users get a more personal touch from the chatbot. Also, the chatbot will remember all your choices and provide you with relevant choices the next time you visit it.

How chatbot can drive revenue for you?

Below we have compiled reasons why chatbots are important for your business and how can they help in increasing revenues:

a. **Higher user customer engagement**

Most businesses these days have a web presence. But with being on the internet, boundaries of day and night, availability and unavailability have changed, so have user expectations. This is probably the biggest reason to use them. Bots give the user an interactive experience. It makes customers feel they are working with someone to help resolve their issue. If done right, bots can help customers find what they are looking for and make them more likely to return.

Customer Engagement

- Clearance Sale : Notify users about on-going clearance sale of products relevant to the users at their nearest outlets.
- Product Finder : Enable consultative selling without the need of a call center
- It offer Notification : Notify users about offers, product launches on products/ services they've shown interest in, and products that's back in stock

b. **Mobile-ready and immediate availability**

Along with a web presence, it has also become increasingly important for brands to have a mobile presence - mobile apps, mobile-optimized websites. Considering how chat has been around on the mobile for ages, most chatbot implementations don't need you to work on tweaking their UI, they are ready to implement and so available to your customers immediately

You might argue that you have an app for that. Having an app for your brand is great, but having users discover that app, download it and use it to stay engaged is not an easy deal. Instead, implementing a chatbot - which works on the mobile browser or a messaging-app which the user regularly uses - makes it all the more reason for a customer to be engaged with the brand c. **It can drive sales**

Chatbots can be intelligent. Depending on a user's preferences or purchases, it can send products to customers which are more likely to convert into sales. Or it can send coupons to users for in-store purchases/discounts. Bots can also be used to link the user to your mCommerce site/app so they can buy the product directly from the convenience of their phones

Sell Intelligently

 Product Recommendations : Push proactive recommendations to users based on their preferences and search and order history. Enable order booking over chat.

d. **Minimal cost - Maximum return**

The best part about bots is they are cheap. Chatbot provide the necessary infrastructure and APIs for creating these bots. They require minimal maintenance and since it is automated, there is no labor-intensive work that goes in there.

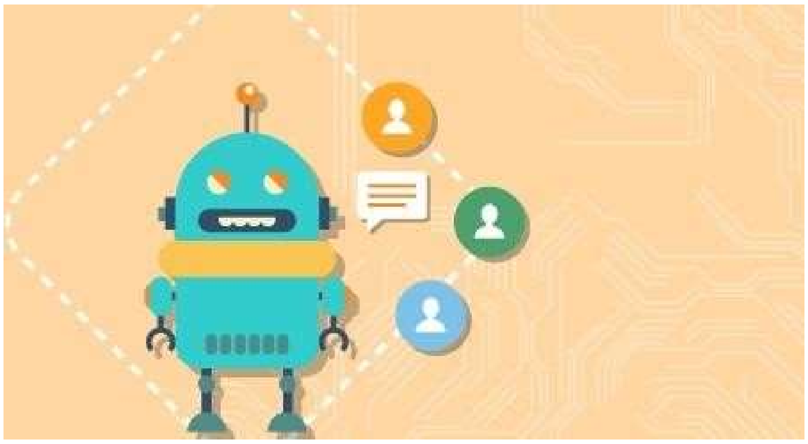
e. **Customer Service**

- Track Order : Keep users up to date with order status. Schedule or reschedule delivery to a provided address or request to pick it up at any other Best Buy outlet.
- Stock outs : Notify users when desired product is available and place order over a chat.
- Returns and Replacements : No waiting time to reach customer care. Customers can instantly place request to replace or return an order.
- Seek Reviews : Reach out to users to seek reviews on the products recently bought

Gift Recommendations

- Recommend relevant gifting options to users, accessing calendar events and understanding the likes and style of beneficiary.
- Opportunity to upsell gift cards for the users for every occasion.

Application across Industries



According to a new survey, 80% of businesses want to integrate chatbots in their business model by 2020. So which industries can reap the greatest benefits by implementing consumer-facing chatbots? According to a chatbot, these major areas of direct-to-consumer engagement are prime:

Chatbots in Restaurant and Retail Industries

Famous restaurant chains like Burger King and Taco bell has introduced their Chatbots to stand out of competitors of the Industry as well as treat their customers quickly. Customers of these restaurants are greeted by the resident Chatbots, and are offered the menu options- like a counter order, the Buyer chooses their pickup location, pays, and gets told when they can head over to grab their food. Chatbots also works to accept table reservations, take special requests and go take the extra step to make the evening special for your guests.

Chatbots are not only good for the restaurant staff in reducing work and pain but can provide a better user experience for the customers.

Chatbots in Hospitality and Travel

For hoteliers, automation has been held up as a solution for all difficulties related to productivity issues, labour costs, a way to ensure consistently, streamlined production processes across the system. Accurate and immediate delivery of information to customers is a major factor in running a successful online Business, especially in the price sensitive and competitive Travel and Hospitality industry. Chatbots particularly have gotten a lot of attention from the hospitality industry in recent months.

Chatbots can help hotels in a number of areas, including time management, guest services and cost reduction. They can assist guests with elementary questions and requests. Thus, freeing up hotel staff to devote more of their time and attention to time-sensitive, critical, and complicated tasks. They are often more cost effective and faster than their human counterparts. They can be programmed to speak to guests in different languages, making it easier for the guests to speak in their local language to communicate.

Chatbots in Health Industry

Chatbots are a much better fit for patient engagement than Standalone apps. Through these Health-Bots, users can ask health related questions and receive immediate responses. These responses are either original or based on responses to similar questions in the database. The impersonal nature of a bot could act as a benefit in certain situations, where an actual Doctor is not needed.

Chatbots ease the access to healthcare and industry has favorable chances to serve their customers with personalized health tips. It can be a good example of the success of Chatbots and Service Industry combo.

Chatbots in E-Commerce

Mobile messengers- connected with Chatbots and the E-commerce business can open a new channel for selling the products online. E-commerce Shopping destination “Spring” was the early adopter. E-commerce future is where brands have their own Chatbots which can interact with their customers through their apps.

Chatbots in Fashion Industry

Chatbots, AI and Machine Learning pave a new domain of possibilities in the Fashion industry, from Data Analytics to Personal Chatbot Stylists. Fashion is such an industry where luxury goods can only be bought in a few physical boutiques and one to one customer service is essential. The Internet changed this dramatically, by giving the customers a seamless but a very impersonal experience of shopping. This particular problem can be solved by Chatbots. Customers can be treated personally with bots, which can exchange messages, give required suggestions and information. Famous fashion brands like Burberry, Tommy Hilfiger have recently launched Chatbots for the London and New York Fashion Week respectively. Sephora a famous cosmetics brand and H&M– a fashion clothing brand have also launched their Chatbots.

Chatbots in Finance

Chatbots have already stepped in Finance Industry. Chatbots can be programmed to assists the customers as Financial Advisor, Expense Saving Bot, Banking Bots, Tax bots, etc. Banks and Fintech have ample opportunities in developing bots for reducing their costs as well as human errors. Chatbots can work for customer's convenience, managing multiple accounts, directly checking their bank balance and expenses on particular things. Further about Finance and Chatbots have been discussed in our earlier blog: Chatbots as your Personal Finance Assistant.

Chatbots in Fitness Industry

Chat based health and fitness companies using Chatbot, to help their customers get personalised health and fitness tips. Tech based fitness companies can have a huge opportunity by developing their own Chatbots offering huge customer base with personalised services. Engage with your fans like never before with news, highlights, game-day info, roster and more.

Chatbots and Service Industry together have a wide range of opportunities and small to big all size of companies using chatbots to reduce their work and help their customers better.

Chatbots in Media

Big publisher or small agency, our suite of tools can help your audience chatbot experience rich and frictionless. Famous News and Media companies like The Wall Street Journal, CNN, Fox news, etc have launched their bots to help you receive the latest news on the go.

Chatbot in Celebrity:

With a chatbot you can now have one-on-one conversation with millions of fans.

Chatbot in Marketing

SMS Marketing

- Why promote just a coupon code that the customer does not know how to use?
- Improve conversions from your existing SMS campaigns.
- Talk to your customers when they want to using "Talk to an Agent" feature.

Email Marketing

- So your eMail has made a solid elevator pitch about your product.
- As a next step, is making customers fill an online form the most exciting way to engage with your customers? ▯ It’s time to rethink the landing page.
- Instantly engage in a conversation with your customers.
- Address their concerns and queries

Social Media Triage

- How effectively are you addressing the negative sentiment around your brand on social media?
- Addressing queries instantly and effectively can convert even an angry customer into a loyal fan.
- Leverage a chatbot as your first response strategy and comfort that customer.

Process

Stage #1: Chatty Bot welcomes you
Teach your assistant to introduce itself in the console.
Stage #2: Print your name
Introduce yourself to the bot.
Stage #3: Guess the age
Use your knowledge of strings and numbers to make the assistant guess your age.
Stage #4: Learning numbers
Your assistant is old enough to learn how to count. And you are experienced enough to apply a for loop at this stage!
Stage #5: Multiple Choice
At this point, the assistant will be able to check your knowledge and ask multiple-choice questions. Add some functions to your code and make the stage even better.

How To Run The Project?

To run this project, you must have installed Python on your PC. After downloading the project, follow the steps below:
Step1: Extract/Unzip the file
Step2: Go inside the project folder, open cmd then type bot.py and enter to start the system.
OR
Step2: Simply, double-click the bot.py file and you are ready to go.

Assignment No : 6	**Date:**
Title: 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**	
Sign:	**Remark:**

Group-C

6. Implement any one of the following Expert System

VII. Information management
VIII. Hospitals and medical facilities
IX. Help desks management
X. Employee performance evaluation
XI. Stock market trading
XII. Airline scheduling and cargo schedules

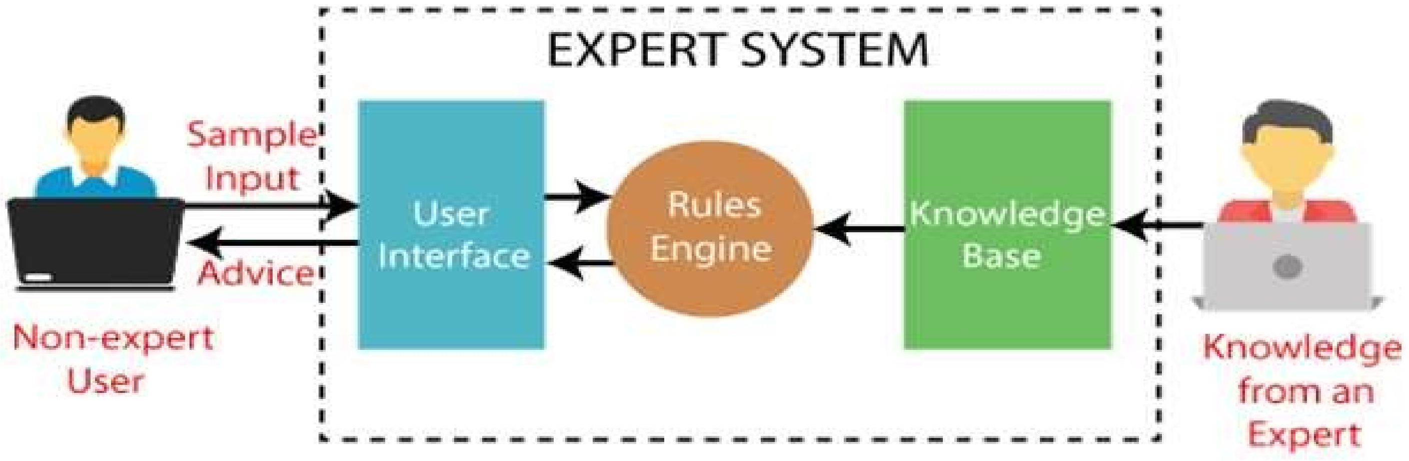
What is an Expert System?

An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert. It performs this by extracting knowledge from its knowledge base using the reasoning and inference rules according to the user queries.

The expert system is a part of AI, and the first ES was developed in the year 1970, which was the first successful approach of artificial intelligence. It solves the most complex issue as an expert by extracting the knowledge stored in its knowledge base. The system helps in decision making for complex problems using **both facts and heuristics like a human expert**. It is called so because it contains the expert knowledge of a specific domain and can solve any complex problem of that particular domain. These systems are designed for a specific domain, such as **medicine, science,** etc.

The performance of an expert system is based on the expert's knowledge stored in its knowledge base. The more knowledge stored in the KB, the more that system improves its performance. One of the common examples of an ES is a suggestion of spelling errors while typing in the Google search box.

Below is the block diagram that represents the working of an expert system:



Below are some popular examples of the Expert System:

- **DENDRAL:** It was an artificial intelligence project that was made as a chemical analysis expert system. It was used in organic chemistry to detect unknown organic molecules with the help of their mass spectra and knowledge base of chemistry.
- **MYCIN:** It was one of the earliest backward chaining expert systems that was designed to find the bacteria causing infections like bacteremia and meningitis. It was also used for the recommendation of antibiotics and the diagnosis of blood clotting diseases.
- **PXDES:** It is an expert system that is used to determine the type and level of lung cancer. To determine the disease, it takes a picture from the upper body, which looks like the shadow. This shadow identifies the type and degree of harm.
- **CaDeT:** The CaDet expert system is a diagnostic support system that can detect cancer at early stages.

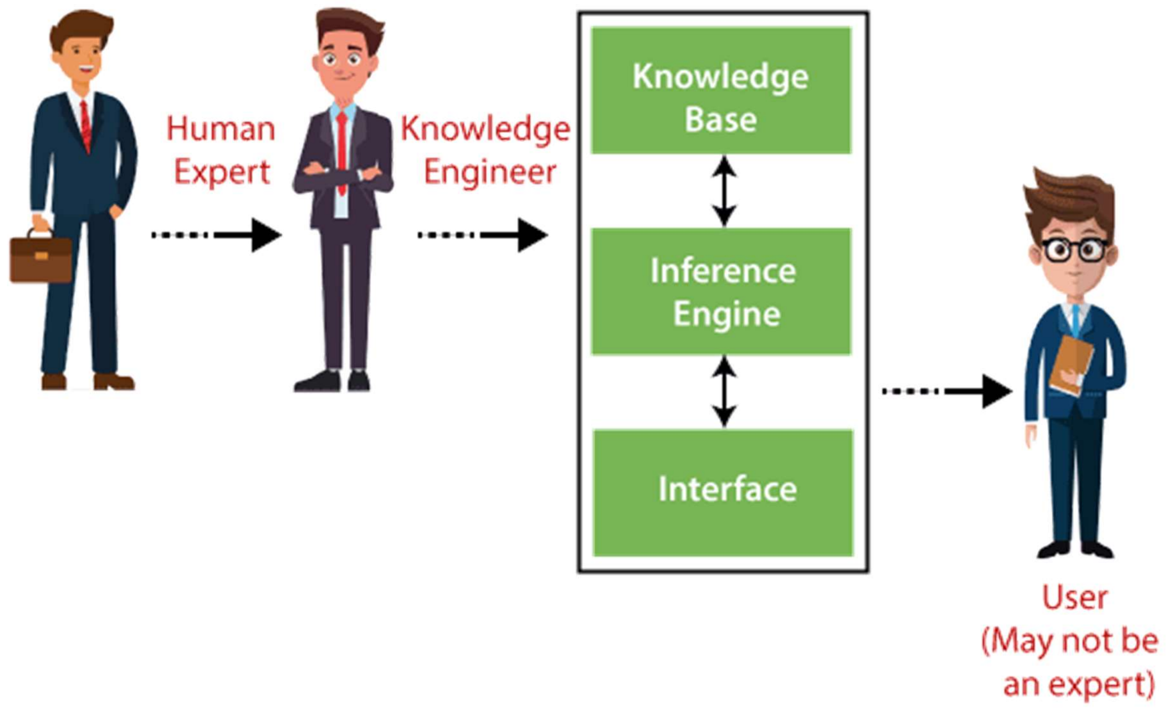
Characteristics of Expert System

- **High Performance:** The expert system provides high performance for solving any type of complex problem of a specific domain with high efficiency and accuracy.
- **Understandable:** It responds in a way that can be easily understandable by the user. It can take input in human language and provides the output in the same way.
- **Reliable:** It is much reliable for generating an efficient and accurate output.
- **Highly responsive:** ES provides the result for any complex query within a very short period of time.

Components of Expert System

An expert system mainly consists of three components:

- **User Interface**
- **Inference Engine**
- **Knowledge Base**



Participants in the development of Expert System

There are three primary participants in the building of Expert System:

1. **Expert:** The success of an ES much depends on the knowledge provided by human experts. These experts are those persons who are specialized in that specific domain.
2. **Knowledge Engineer:** Knowledge engineer is the person who gathers the knowledge from the domain experts and then codifies that knowledge to the system according to the formalism.
3. **End-User:** This is a particular person or a group of people who may not be experts, and working on the expert system needs the solution or advice for his queries, which are complex.

Advantages of Expert System

- These systems are highly reproducible.
- They can be used for risky places where the human presence is not safe.
- Error possibilities are less if the KB contains correct knowledge.
- The performance of these systems remains steady as it is not affected by emotions, tension, or fatigue.
- They provide a very high speed to respond to a particular query.

Limitations of Expert System

- The response of the expert system may get wrong if the knowledge base contains the wrong information.
- Like a human being, it cannot produce a creative output for different scenarios.
- Its maintenance and development costs are very high.
- Knowledge acquisition for designing is much difficult.
- For each domain, we require a specific ES, which is one of the big limitations.
- It cannot learn from itself and hence requires manual updates.

Applications of Expert System

- **In designing and manufacturing domain**
  It can be broadly used for designing and manufacturing physical devices such as camera lenses and automobiles.
- **In the knowledge domain**
  These systems are primarily used for publishing the relevant knowledge to the users. The two popular ES used for this domain is an advisor and a tax advisor.
- **In the finance domain**
  In the finance industries, it is used to detect any type of possible fraud, suspicious activity, and advise bankers that if they should provide loans for business or not.
- **In the diagnosis and troubleshooting of devices**
  In medical diagnosis, the ES system is used, and it was the first area where these systems were used.
- **Planning and Scheduling**
  The expert systems can also be used for planning and scheduling some particular tasks for achieving the goal of that task.

Assignment No : 7	**Date:**
Title: 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.	
Sign:	**Remark:**

AIM: Write a C 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.

HARDWARE AND SOFTWARE REQUIREMENT:

a. Intel based Desktop PC: - RAM of 512 MB

b. Turbo C / Borland C

THEORY:

i. String

The string is the one-dimensional array of characters terminated by a null ('\0'). Each and every character in the array consumes one byte of memory, and the last character must always be 0. The termination character ('\0') is used to identify where the string ends. In C language string declaration can be done in two ways

1. By char array
2. By string literal

Let's see the example of declaring **string by char array** in C language.

```
1.  char ch[17]={'o', 'n', 'l', 'i', 'n', 'e', 's', 'm', 'a', 'r', 't', 't', 'r', 'a',
'i', 'n', 'e', 'r', '\0'};
```

As we know, array index starts from 0, so it will be represented as in the figure given below.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
o	n	l	i	n	e	s	m	a	r	t	t	r	a	i	n	e	r	\0

While declaring string, size is not mandatory. So we can write the above code as given below:

```
char ch[]={'o', 'n', 'l', 'i', 'n', 'e', 's', 'm', 'a', 'r', 't', 't', 'r', 'a',
'i', 'n', 'e', 'r', '\0'};
```

We can also define the **string by the string literal** in C language. For example:

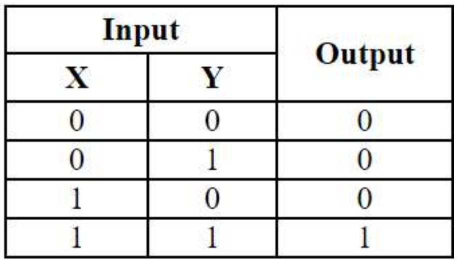
```
2.  char str[]="onlinesmarttrainer";
```

In such case, '\0' will be appended at the end of the string by the compiler.

ii. AND Operation

There are two inputs and one output in binary **AND** operation. The inputs and result to a binary **AND** operation can only be **0** or **1.**The binary **AND** operation will always produce a **1** output if both inputs are **1** and will produce a **0** output if both inputs are **0**.For two different inputs, the output will be **0.**

AND Truth Table



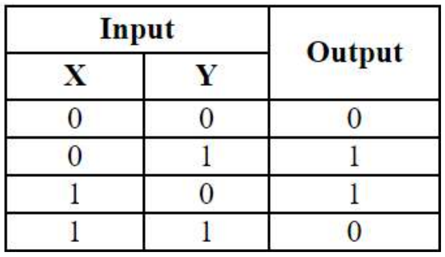
Input		Output
X	Y	
0	0	0
0	1	0
1	0	0
1	1	1

iii. XOR Operation

There are two inputs and one output in binary **XOR** (exclusive **OR**) operation. It is similar to **ADD** operation which takes two inputs and produces one result i.e. one output.

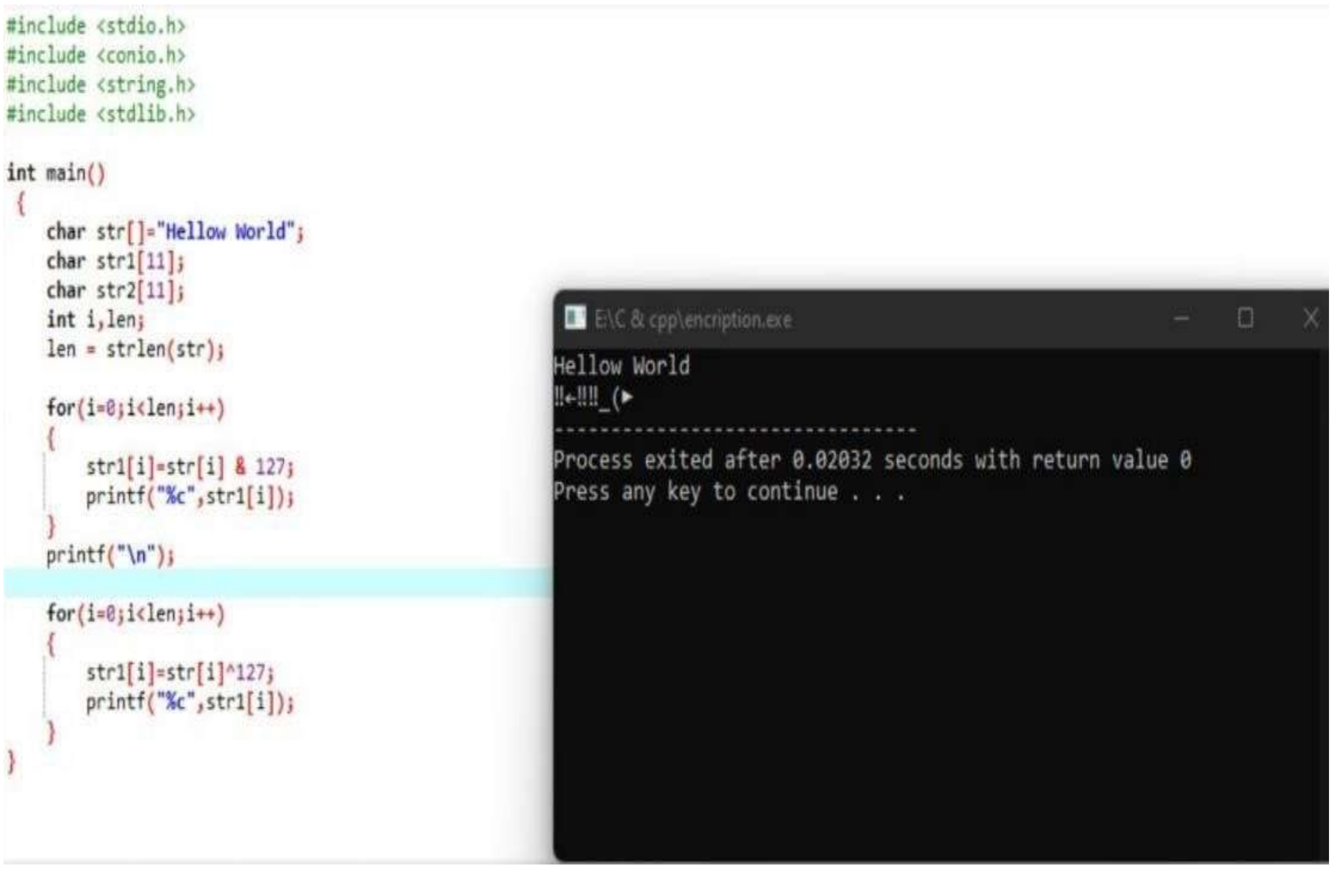
The inputs and result to a binary **XOR** operation can only be **0** or **1.**The binary **XOR** operation will always produce a **1** output if either of its inputs is **1** and will produce a **0** output if both of its inputs are **0** or **1**.

XOR Truth Table



Input		Output
X	Y	
0	0	0
0	1	1
1	0	1
1	1	0

SOURCE CODE & OUTPUT:



```c
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>

int main()
 {
    char str[]="Hellow World";
    char str1[11];
    char str2[11];
    int i,len;
    len = strlen(str);

    for(i=0;i<len;i++)
    {
        str1[i]=str[i] & 127;
        printf("%c",str1[i]);
    }
    printf("\n");

    for(i=0;i<len;i++)
    {
        str1[i]=str[i]^127;
        printf("%c",str1[i]);
    }
}
```

E:\C & cpp\encription.exe

```
Hellow World
‼←‼‼_(►
--------------------------------
Process exited after 0.02032 seconds with return value 0
Press any key to continue . . .
```

Assignment No : 8	**Date:**
Title: Write a Java/C/C++/Python program to perform encryption and decryption using the method of Transposition technique.	
Sign:	**Remark:**

Aim: Write a Java/C/C++/Python program to perform encryption and decryptionusing the method of Transposition technique**.**

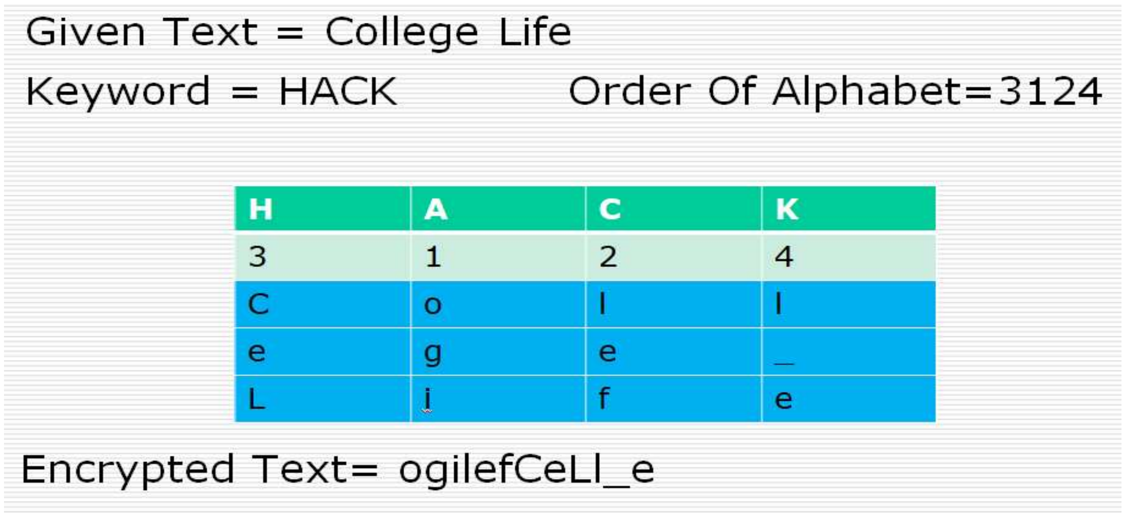
Theory:

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 constitutesa permutation of the plaintext.

1. Encryption

In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.

a) The message is written out in rows of a fixed length, and then read out againcolumn by column, and the columns are chosen in some scrambled order.
b) Width of the rows and the permutation of the columns are usually defined by a keyword.
c) For example, the word HACK is of length 4 (so the rows are of length 4), and the permutation is defined by the alphabetical order of the letters in the keyword. In thiscase, the order would be “3 1 2 4”.
d) Any spare spaces are filled with nulls or left blank or placed by a character (Example: _).
e) Finally, the message is read off in columns, in the order specified by the keyword.



Given Text = College Life

Keyword = HACK Order Of Alphabet=3124

H	A	C	K
3	1	2	4
C	o	l	l
e	g	e	_
L	i	f	e

Encrypted Text= ogilefCeLl_e

2. Decryption

a) To decipher it, the recipient has to work out the column lengths by dividing themessage length by the key length.

b) Then, write the message out in columns again, then re-order the columns byreforming the key word.

Question

The following message was encrypted with a columnar transposition cipher using a full rectangular array and keyword mathematics. Decrypt it.

RIUGS IPNCT MSPAL AUNCY SOOCH UEYSA RTE

Assignment No :9	**Date:**
Title: Implementation of DES.	
Sign:	**Remark:**

Title: Implementation of DES

Problem Definition: Write a Java / Python program for performs the various operations on DES

Prerequisite:

- Basic concepts of Java /Python/C++

Tools/Framework/Language Used: Java / Python/C++

Learning Objectives: Understand the implementation of DES key generation passing keys, encryption and decryption.

Outcomes: After completion of this assignment students will understand how Encrypt and Decrepit work.

Theory:

Concepts:

DES, developed by Professor Edward Schaefer of Santa Clara University [SCHA96], is an educational rather than a secure encryption algorithm. It has similar properties and structure to DES with much smaller parameters. The reader might find it useful to work through an example byhand while following the discussion in this Appendix.

OVERVIEW

The S-DES encryption algorithm takes an 8-bit block of plaintext (example: 10111101) and a 10-bitkey as input and produces an 8-bit block of ciphertext as output. The S-DES decryption algorithm takes an 8-bit block of ciphertext and the same 10-bit key used to produce that ciphertext as input and produces the original 8-bit block of plaintext.

The encryption algorithm involves five functions: an initial permutation (IP); a complex function labeled fK, which involves both permutation and substitution operations and depends on a key input; a simple permutation function that switches (SW) the two halves of the data; the function fK again; and finally a permutation function that is the inverse of the initial permutation (IP^{-1}). As was mentioned in Chapter 2, the use of multiple stages of permutation and substitution results in a more complex algorithm, which increases the difficulty of cryptanalysis.

The function fK takes as input not only the data passing through the encryption algorithm, but also an 8-bit key. The algorithm could have been designed to work with a 16-bit key, consisting of two 8-bit subkeys, one used for each occurrence of fK. Alternatively, a single 8-bit key could have been used, with the same key used twice in the algorithm. A compromise is to use a 10-bit key from which two 8-bit subkeys are generated, as depicted in Figure G.1. In this case, the key is first subjected to a permutation (P10). Then a shift operation is performed. The output of the shift operation then passes through a permutation function that produces an 8-bit output (P8) for the first subkey (K_1). The output of the shift operation also feeds into another shift and another instance of P8 to produce the second subkey (K_2).

We can concisely express the encryption algorithm as a composition[1] of functions:

$$\mathrm{IP}^{-1} \,!\, \mathrm{f}K_2 \,!\, \mathrm{SW} \,!\, \mathrm{f}K_1 \,!\, \mathrm{IP}$$

which can also be written as:

$$\text{ciphertext} = \mathrm{IP}^{-1}\Big(\mathrm{f}K_2\big(\mathrm{SW}\big(\mathrm{f}K_1\big(\mathrm{IP}(\text{plaintext})\big)\big)\big)\Big)$$

where

$$K_1 = \mathrm{P8}\big(\mathrm{Shift}\big(\mathrm{P10}(\text{key})\big)\big)$$

$$K_2 = \mathrm{P8}\big(\mathrm{Shift}\big(\mathrm{Shift}\big(\mathrm{P10}(\text{key})\big)\big)\big)$$

Decryption is also shown in Figure G.1 and is essentially the reverse of encryption:

$$\text{plaintext} = \mathrm{IP}^{-1}\Big(\mathrm{f}K_1\big(\mathrm{SW}\big(\mathrm{f}K_2\big(\mathrm{IP}(\text{ciphertext})\big)\big)\big)\Big)$$

We now examine the elements of S-DES in more detail.

DES KEY GENERATION

S-DES depends on the use of a 10-bit key shared between sender and receiver. From this key, two 8-bit subkeys are produced for use in particular stages of the encryption and decryption algorithm.Figure G.2 depicts the stages followed to produce the subkeys.

First, permute the key in the following fashion. Let the 10-bit key be designated as (k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}). Then the permutation P10 is defined as:

$$P10(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) = (k_3, k_5, k_2, k_7, k_4, k_{10}, k_1, k_9, k_8, k_6)$$

4. **Definition:** If f and g are two functions, then the function F with the equation y = F(x) = g[f(x)] is called the **composition** of f and g and is denoted as F = g ! f .

P10 can be concisely defined by the display:

P10									
3	5	2	7	4	10	1	9	8	6

This table is read from left to right; each position in the table gives the identity of the input bit that produces the output bit in that position. So the first output bit is bit 3 of the input; the second output bit is bit 5 of the input, and so on. For example, the key (1010000010) is permuted to (1000001100). Next, perform a circular left shift (LS-1), or rotation, separately on the first five bits and the second five bits. In our example, the result is (00001 11000).

Next we apply P8, which picks out and permutes 8 of the 10 bits according to the following rule:

P8							
6	3	7	4	8	5	10	9

The result is subkey 1 (K_1). In our example, this yields (10100100)

We then go back to the pair of 5-bit strings produced by the two LS-1 functions and perform a circular left shift of 2 bit positions on each string. In our example, the value (00001 11000) becomes (00100 00011). Finally, P8 is applied again to produce K_2. In our example, the result is (01000011).

DES ENCRYPTION

Figure G.3 shows the S-DES encryption algorithm in greater detail. As was mentioned, encryption involves the sequential application of five functions. We examine each of these.

Initial and Final Permutations

The input to the algorithm is an 8-bit block of plaintext, which we first permute using the IP function:

IP							
2	6	3	1	4	8	5	7

This retains all 8 bits of the plaintext but mixes them up. At the end of the algorithm, the inverse permutation is used:

IP^{-1}							
4	1	3	5	7	2	8	6

It is easy to show by example that the second permutation is indeed the reverse of the first; that is, $IP^{-1}(IP(X)) = X$.

The Function f$_K$

The most complex component of S-DES is the function f_K, which consists of a combination of permutation and substitution functions. The functions can be expressed as follows. Let L and R be the leftmost 4 bits and rightmost 4 bits of the 8-bit input to f_K, and let F be a mapping (not necessarily one to one) from 4-bit strings to 4-bit strings. Then we let

$$f_K(L, R) = (L\ !\ F(R, SK), R)$$

where SK is a subkey and ! is the bit-by-bit exclusive-OR function. For example, suppose the output of the IP stage in Figure G.3 is (10111101) and F(1101, SK) = (1110) for some key SK. Then f_K(10111101) = (01011101) because (1011) ! (1110) = (0101).

We now describe the mapping F. The input is a 4-bit number ($n_1n_2n_3n_4$). The first operation is an expansion/permutation operation:

E/P

4 1 2 3 2 3 4 1

For what follows, it is clearer to depict the result in this fashion:

$$\begin{array}{c|cc|c} n_4 & n_1 & n_2 & n_3 \\ n_2 & n_3 & n_4 & n_1 \end{array}$$

The 8-bit subkey $K_1 = (k_{11}, k_{12}, k_{13}, k_{14}, k_{15}, k_{16}, k_{17}, k_{18})$ is added to this value using exclusive-OR:

$$\begin{array}{c|cc|c} n_4 ! k_{11} & n_1 ! k_{12} & n_2 ! k_{13} & n_3 ! k_{14} \\ n_2 ! k_{15} & n_3 ! k_{16} & n_4 ! k_{17} & n_1 ! k_{18} \end{array}$$

Let us rename these 8 bits:

$$\begin{array}{c|cc|c} p_{0,0} & p_{0,1} & p_{0,2} & p_{0,3} \\ p_{1,0} & p_{1,1} & p_{1,2} & p_{1,3} \end{array}$$

The first 4 bits (first row of the preceding matrix) are fed into the S-box S0 to produce a 2-bit output, and the remaining 4 bits (second row) are fed into S1 to produce another 2-bit output. These two boxes are defined as follows:

$$S0 = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 0 & 3 & 2 \\ 1 & 3 & 2 & 1 & 0 \\ 2 & 0 & 2 & 1 & 3 \\ 3 & 3 & 1 & 3 & 2 \end{array} \qquad S1 = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 1 & 2 & 3 \\ 1 & 2 & 0 & 1 & 3 \\ 2 & 3 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 & 3 \end{array}$$

The S-boxes operate as follows. The first and fourth input bits are treated as a 2-bit number that specify a row of the S-box, and the second and third input bits specify a column of the S-box. The entry in that row and column, in base 2, is the 2-bit output. For example, if $(p_{0,0}p_{0,3})$ = (0) and $(p_{0,1}p_{0,2})$ = (10), then the output is from row 0, column 2 of S0, which is 3, or (11) in binary. Similarly, $(p_{1,0}p_{1,3})$ and $(p_{1,1}p_{1,2})$ are used to index into a row and column of S1 to produce an additional 2 bits.

Next, the 4 bits produced by S0 and S1 undergo a further permutation as follows:

P4			
2	4	3	1

The output of P4 is the output of the function F.

The Switch Function

The function f_K only alters the leftmost 4 bits of the input. The switch function (SW) interchanges the left and right 4 bits so that the second instance of f_K operates on a different 4 bits. In this second instance, the E/P, S0, S1, and P4 functions are the same. The key input is K_2.

ANALYSIS OF DES

A brute-force attack on simplified DES is certainly feasible. With a 10-bit key, there are only 2^{10} = 1024 possibilities. Given a ciphertext, an attacker can try each possibility and analyze the result to determine if it is reasonable plaintext.

What about cryptanalysis? Let us consider a known plaintext attack in which a single plaintext (p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8) and its ciphertext output (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8) are known and the key (k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) is unknown. Then each c_i is a polynomial function g_i of the p_j's and k_j's. We can therefore express the encryption algorithm as 8 nonlinear equations in 10 unknowns. There are a number of possible solutions, but each of these could be calculated and then analyzed. Each of the permutations and additions in the algorithm is a linear mapping. The nonlinearity comes from the S-boxes. It is useful to write down the equations for these boxes. For clarity, rename ($p_{0,0}$, $p_{0,1}$,$p_{0,2}$, $p_{0,3}$) = (a, b, c, d) and ($p_{1,0}$, $p_{1,1}$,$p_{1,2}$, $p_{1,3}$) = (w, x, y,z), and let the 4-bit output be (q, r, s, t) Then the operation of the S0 is defined by the following equations:

$$q = abcd + ab + ac + b + d$$
$$r = abcd + abd + ab + ac + ad + a + c + 1$$

where all additions are modulo 2. Similar equations define S1. Alternating linear maps with these nonlinear maps results in very complex polynomial expressions for the ciphertext bits, making cryptanalysis difficult. To visualize the scale of the problem, note that a polynomial equation in 10 unknowns in binary arithmetic can have 2^{10} possible terms. On average, we might therefore expect each of the 8 equations to have 2^9 terms. The interested reader might try to find these equations witha symbolic processor. Either the reader or the software will give up before much progress is made.

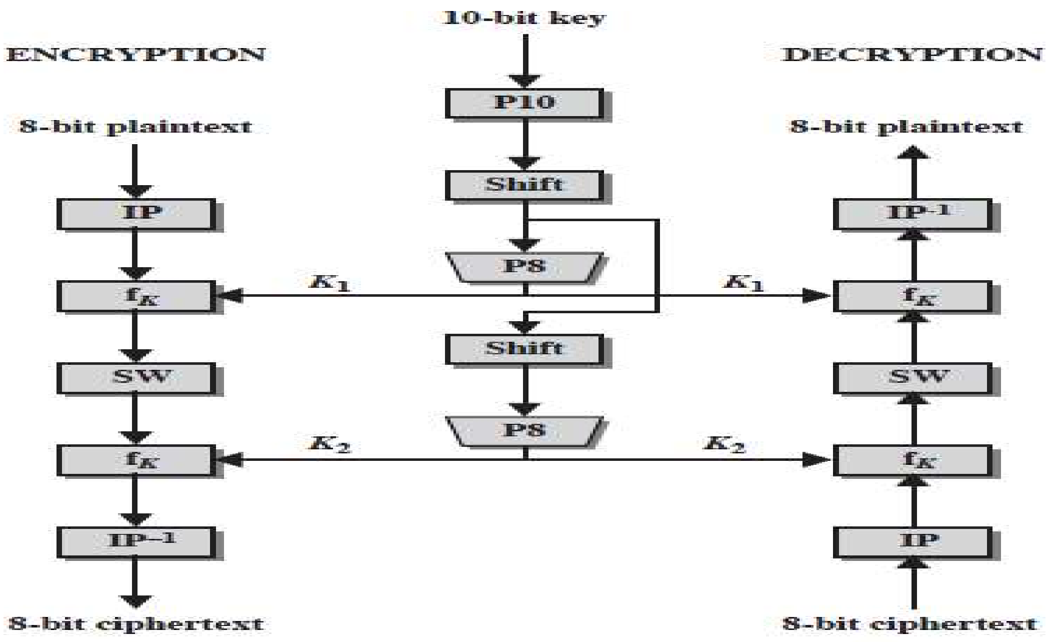
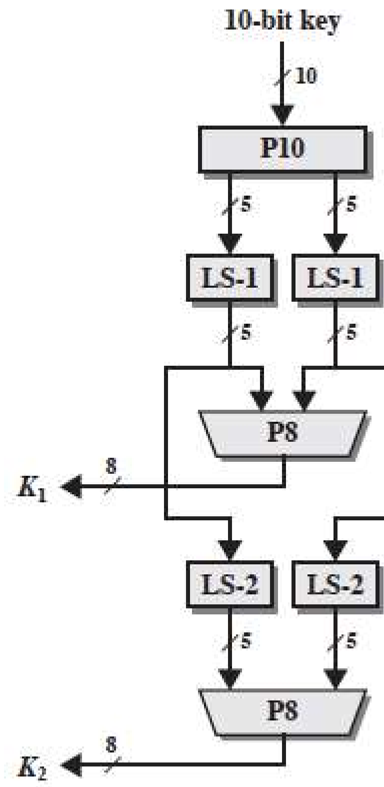


Figure G.1 Simplified DES Scheme

.



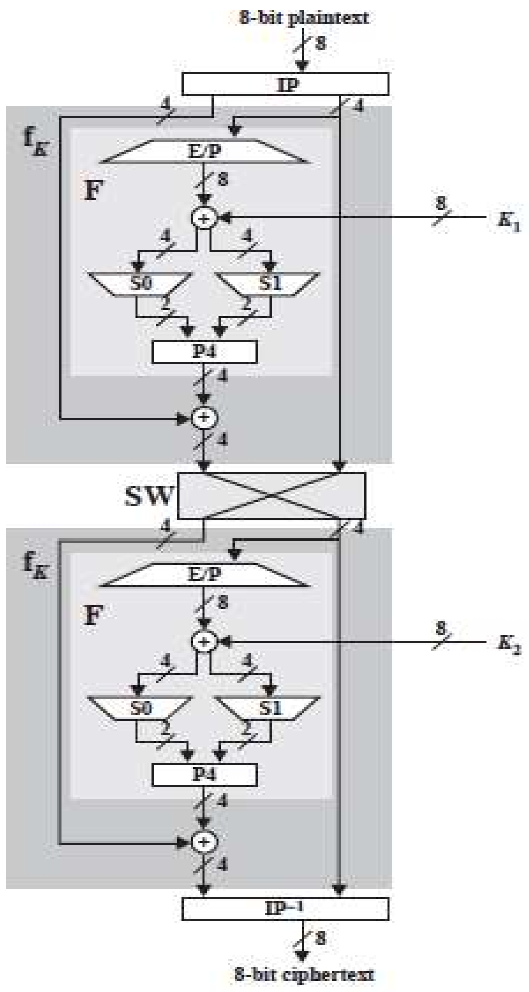


Figure G.2 Key Generation for Simplified DES

Figure G.3 Simplified DES Encryption Detail

Conclusion:

Thus we have implemented SDES and study the encryption and decryption process with key generation technique. DES encryption with two keys instead of one key already will increase the efficiency of cryptography.

Assignment No : 10	**Date:**
Title: Implementation of AES.	
Sign:	**Remark:**

Title: Implementation of AES

Problem Definition: Write a Java / Python program for performs the various operations on AES

Prerequisite:

- Basic concepts of Java /Python/C++

Tools/Framework/Language Used: Java / Python/C++

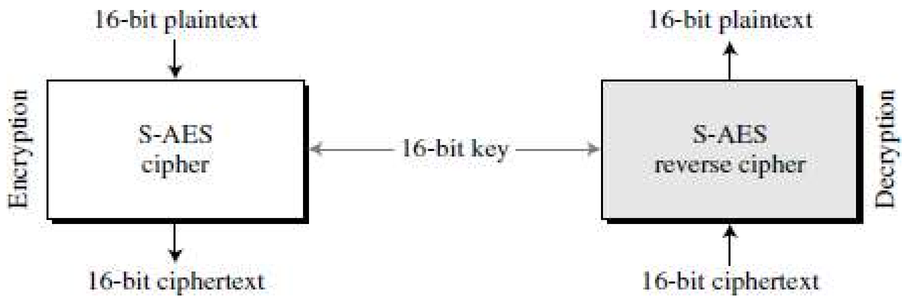
Learning Objectives: Understand the implementation of AES key generation passing keys, encryption and decryption.

Outcomes: After completion of this assignment students will understand how Encrypt and Decrepit work.

Theory:

Concepts:

AES is developed by Professor Edward Schaefer of Santa Clara University,
is an educational tool designed to help students learn the structure of AES using smaller blocks and keys.



AES is a block cipher, as shown in Figure 1.

Rounds

S-AES is a non-Feistel cipher that encrypts and decrypts a data block of 16 bits. It uses one pre-round transformation and two rounds. The cipher key is also 16 bits.

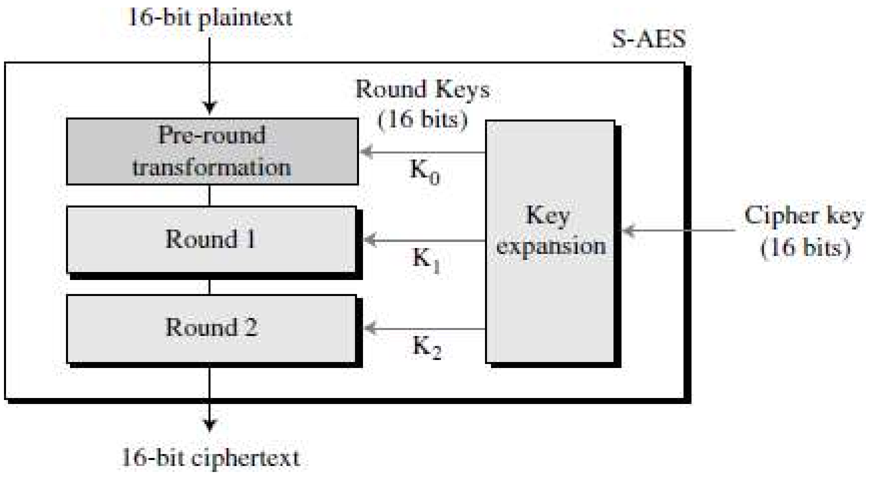
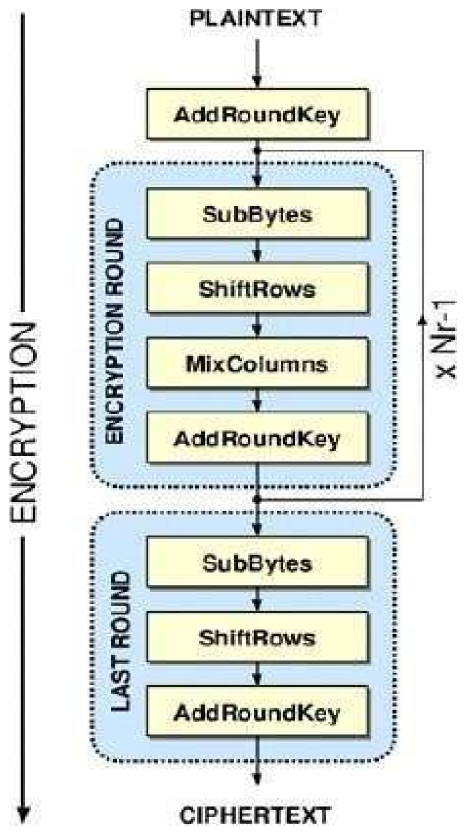


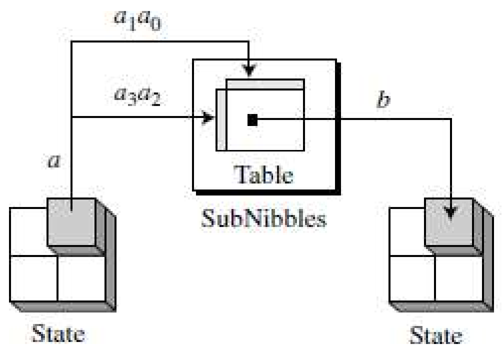
Figure P.2 Structure of AES

The above Fig. shows the general design for the encryption algorithm (called the cipher); thedecryption algorithm (called the inverse cipher) is similar, but the round keys are applied in the reverse order. In Figure P.2, the round keys, which are created by the key-expansion algorithm,
are always 16 bits, the same size as the plaintext or ciphertext block. In S-AES, there are three round keys, K0, K1, and K2.



Substitution
Substitution is done for each nibble (4-bit data unit). Only one table is used for transformations of every nibble, which means that if two nibbles are the same, the transformation is also the same. Inthis appendix, transformation is defined by a table lookup process.

SubNibbles
The first transformation, **SubNibbles,** is used at the encryption site. To substitute a nibble,we interpret the nibble as 4 bits. The left 2 bits define the row and the right 2 bits define the column of the substitution table. The hexadecimal digit at the junction of the row and the column is the new nibble.
In the SubNibbles transformation, the state is treated as a 2 ×2 matrix of nibbles.
Transformation is done one nibble at a time. The contents of each nibble is changed, but thearrangement of the nibbles in the matrix remains the same. In the process, each nibble is transformed independently: There are four distinct nibble-to-nibble transformations.



a_1a_0

a_3a_2	00	01	10	11
00	9	4	A	B
01	D	1	8	5
10	6	2	0	3
11	C	E	F	7

SubNibbles table

a_1a_0

a_3a_2	00	01	10	11
00	A	5	9	B
01	1	7	8	F
10	6	0	2	3
11	C	4	D	E

InvSubNibbles table

Figure P.7
SubNibbles transformations

Permutation
Another transformation found in a round is shifting, which permutes the nibbles. Shifting transformation in S-AES is done at the nibble level; the order of the bits in the nibble is not changed.

ShiftRows
In the encryption, the transformation is called *ShiftRows* and the shifting is to the left. The number of shifts depends on the row number (0, 1) of the state matrix. This means row 0 is not shifted at all and row 1 is shifted 1 nibble. Figure P.9 shows the shifting transformation. Note that the ShiftRows transformation operates one row at a time.

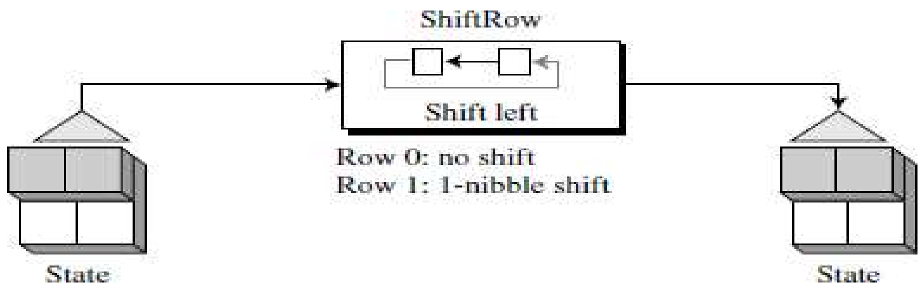
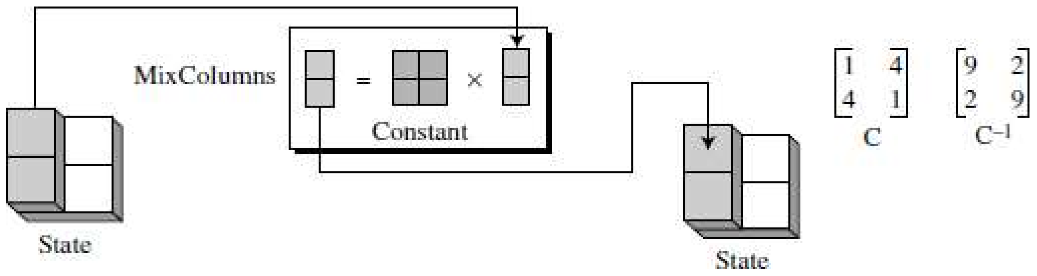


Figure P.9 *ShiftRows transformation*

MixColumns

The *MixColumns* transformation operates at the column level; it transforms each column of the state into a new column. The transformation is actually the matrix multiplication of a state column by a constant square matrix. The nibbles in the state column and constants matrix are interpreted as 4-bit words (or polynomials) with coefficients in GF(2). Multiplication of bytes is done in GF(24) with modulus (*x*4+*x*+ 1) or (10011). Addition is the same as XORing of 4-bit words. Figure P.11 shows the MixColumns transformation.



$$\begin{bmatrix} 1 & 4 \\ 4 & 1 \end{bmatrix} \quad \begin{bmatrix} 9 & 2 \\ 2 & 9 \end{bmatrix}$$

Figure P.11
MixColumns transformation

Key Adding
Probably the most important transformation is the one that includes the cipher key.
All previous transformations use known algorithms that are invertible. If the cipher
key is not added to the state at each round, it is very easy for the adversary to find the plaintext, given the ciphertext. The cipher key is the only secret between Alice and Bob in this case.

ES uses a process called key expansion (discussed later in this appendix)
thatcreates three round keys from the cipher key. Each round key is 16 bits
long

it is treated as two 8-bit words. For the purpose of adding the key to the state, each word is considered as a column matrix.

AddRoundKey

AddRoundKey also proceeds one column at a time. It is similar to MixColumns in this respect. MixColumns multiplies a constant square matrix by each state column AddRoundKey adds a round key word with each state column matrix. The operations in MixColumns are matrix multiplication; the operations in AddRoundKey are matrix addition. The addition is performed in the GF(2 4) field. Because addition and subtraction in this field are the same, the AddRoundKey transformation is the inverse of itself.

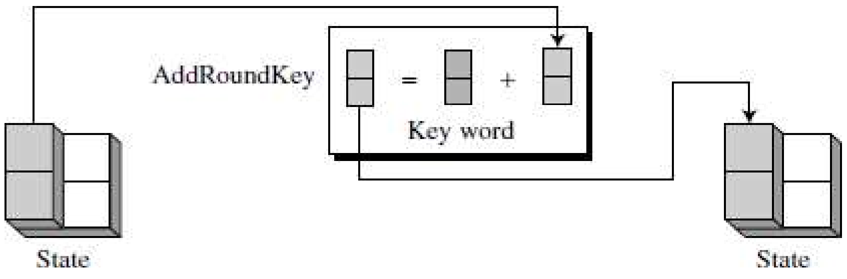


Figure P.13 shows the AddRoundKey transformation.

You take the following aes steps of encryption for a 128-bit block:

1. Derive the set of round keys from the cipher key.
2. Initialize the state array with the block data (plaintext).
3. Add the initial round key to the starting state array.
4. Perform nine rounds of state manipulation.
5. Perform the tenth and final round of state manipulation.
6. Copy the final state array out as the encrypted data (ciphertext)

Conclusion:

Thus we have implemented AES and study the encryption and decryption process.
The purpose was to create an algorithm that was resistant against known attacks, simple, and quick to code.

Assignment No : 11	**Date:**
Title: Implementation of RSA	
Sign:	**Remark:**

Title: Implementation of RSA

Problem Definition: Write a Java / Python program for performs the RSA Algorithm.

Prerequisite:

- Basic concepts of Java /Python

Tools/Framework/Language Used: Java / Python

Learning Objectives: Understand the implementation of RSA.

Outcomes: After completion of this assignment students will understand how RSA Algorithm works. The RSA algorithm's security is based on the difficulty of factoring large numbers into their prime factors.

Theory:

Concepts:

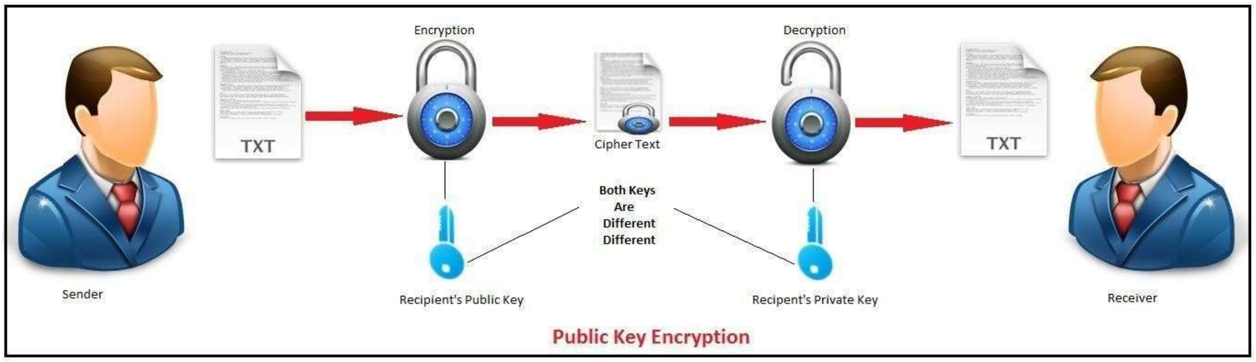
Data security and confidentiality issues are an important aspect of information systems. Because of the confidentiality and security of data, it is essential always to be concerned both for common security, as well as for the privacy of individuals. In this case, it is closely related to how importantthe information and data is sent and received by the interested person. Information is no longer useful if the information is tapped by an irresponsible person. For the data is not known by parties who are not concerned, then each data owner is always trying to do security with some specific techniques.

Cryptography is a science or art of securing a message and done by a cryptographer. While cryptanalysis is a science and art of solving ciphertext to plaintext. The person doing it is called cryptanalysis. The word cryptography comes from the Greek word "kryptos" which means to hide and "graphein" which means to write. Cryptography can be defined as a science that transforms information from understandable forms into incomprehensible forms. Cryptography Algorithm always consists of two parts, namely encryption, and decryption. Encryption is a process done to convert a readable message into an unreadable message (ciphertext). Decryption is the opposite of theencryption process, returning unread messages to unread messages. The encryption and decryption process is governed by one or more cryptographic keys. In a system where there is a cryptographic algorithm, plus all possible plaintext, ciphertext and keys are called cryptosystems.

Public-Key one of the major difficulties of conventional encryption is the need for security to distribute keys used in a secure state. The next generation technique is a modern cryptography. It is a way to remove this weakness with an encryption model that does not require a key to be distributed. This method is known as the "public-key" encryption and was first introduced in 1976. For conventional encryption, the keys used in the encryption and decryption process are the same.However, this is not necessary conditions. However, it is possible to construct an algorithm that uses one key for encryption and decryption. This process requires different keys for encryption and decryption. Furthermore, it is possible to create an algorithm in which the knowledge of the encryption algorithm and the encryption key is insufficient to determine the decryption key

RSA Algorithm is used to encrypt and decrypt data in modern computer systems and other electronic devices. RSA algorithm is an asymmetric cryptographic algorithm as it creates 2 different keys for the purpose of encryption and decryption. It is public key cryptography as one of the keys involved is made public. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman who first publicly described it in 1978.

RSA makes use of prime numbers (arbitrary large numbers) to function. The public key is made available publicly (means to everyone) and only the person having the private key with them can decrypt the original message.



The RSA algorithm involves three steps:

1. Key generation
2. Encryption
3. Decryption.

Key generation

For the RSA cryptosystem, we first start off by generating two large prime numbers, 'p' and 'q', of about the same size in bits. Next, compute 'n' where n = pq, and 'x' such that, x = (p -1)(q-1). We select a small odd integer less than x, which is relatively prime to it i.e. gcd(e,x) = 1. Finally we find out the unique multiplicative inverse of e modulo x, and name it 'd'. In other words, ed = 1 (mod x), and of course, 1 < d < x. Now, the public key is the pair (e,n) and the private key is d.

RSA Encryption

Suppose Bob wishes to send a message (say 'm') to Alice. To encrypt the message using the RSA encryption scheme, Bob must obtain Alice's public key pair (e,n). The message to send must now be encrypted using this pair (e,n). However, the message 'm' must be represented as an integer in the interval [0,n-1]. To encrypt it, Bob simply computes the number 'c' where c = m ^ e mod n. Bob sendsthe ciphertext c to Alice.

RSA Decryption

To decrypt the ciphertext c, Alice needs to use her own private key d (the decryption exponent) and the modulus n. Simply computing the value of c ^ d mod n yields back the decypted message (m). Any article treating the RSA algorithm in considerable depth proves the correctness of the decryption algorithm.

Working of RSA Algorithm

RSA involves use of public and private key for its operation. The keys are generated using the following steps:-

1. Two prime numbers are selected as **p** and **q**
2. **n = pq** which is the modulus of both the keys.
3. Calculate **totient = (p-1)(q-1)**
4. Choose **e** such that **e > 1** and coprime to **totient** which means **gcd (e, totient)** must be equal to **1**, **e** is the public key
5. Choose **d** such that it satisfies the equation **de = 1 + k (totient)**, **d** is the private key not known to everyone.
6. Cipher text is calculated using the equation **c = m^e mod n** where **m** is the message.
7. With the help of **c** and **d** we decrypt message using equation **m = c^d mod n** where **d**is the private key.

CONCLUSION

By using RSA Algorithm, a system capable of ensuring data confidentiality can be established. Froma technical point of view, RSA has easy and simple encryption.