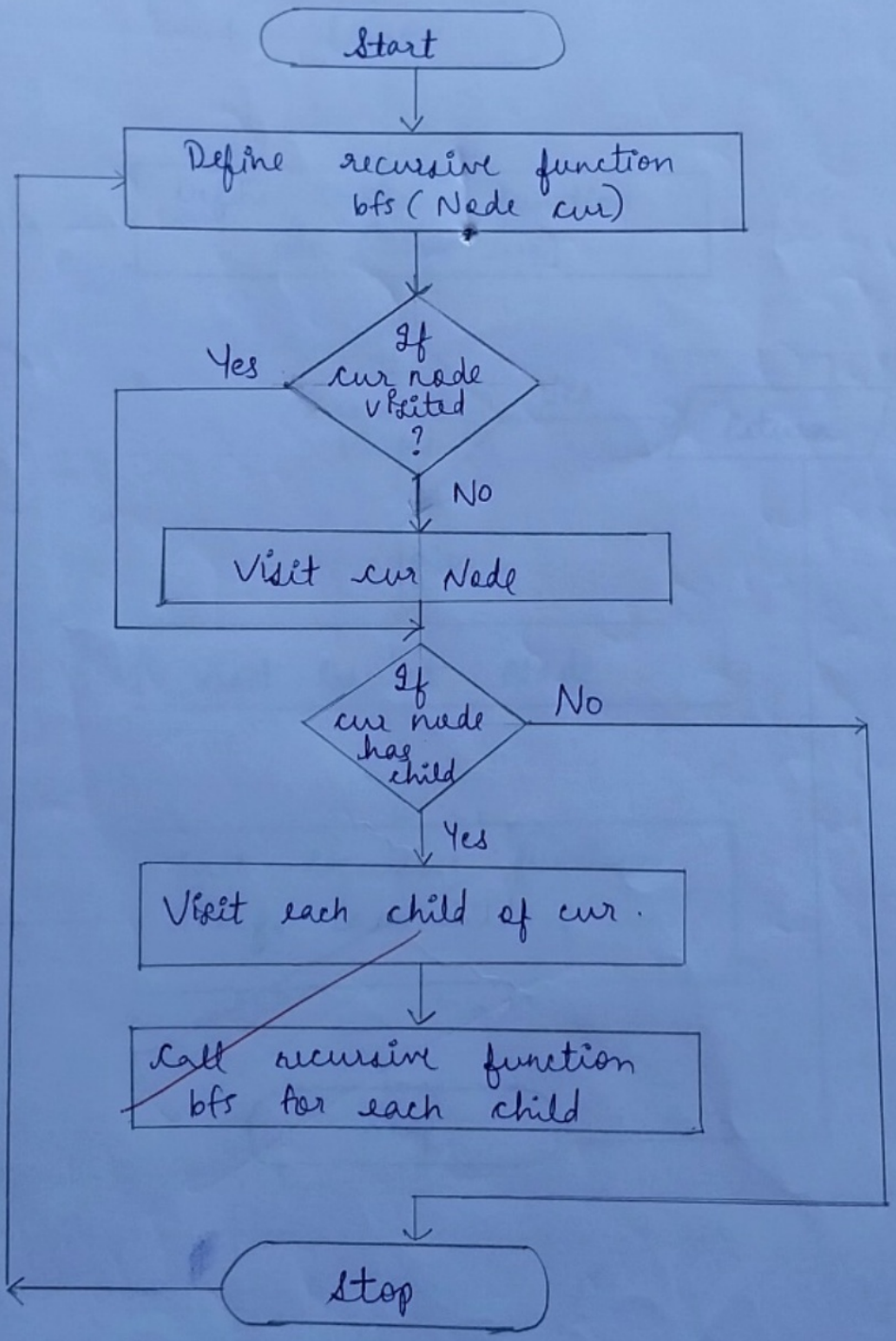
Algorithm :

Dfs (Recursive):

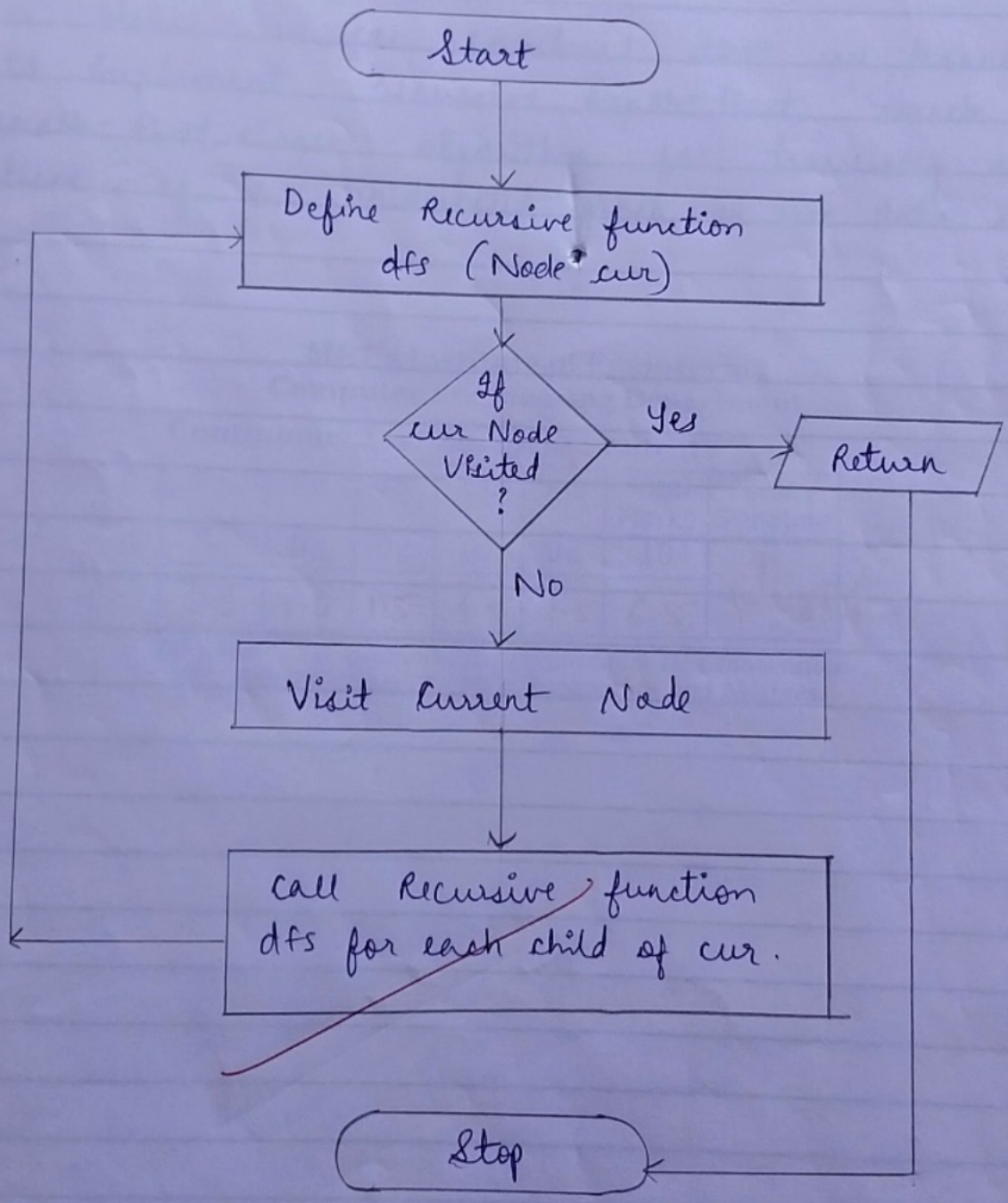
Step 1 : Start
Step 2 : Define function dfs (Node cur) which takes current node as an argument.
If current node is not null & not visited
Then (i) Print current node.
(ii) Call dfs function recursively for each child of current node.
Step 3 : Call function dfs for root node (starting vertex).
Step 4 : Stop.

Bfs (Recursive) :
Step 1 : Start
Step 2 : Define function bfs (Node cur) which takes current node as an argument.
(i) If current node is not visited
Then visit it (print it)
(ii) Visit every child of current node.
(iii) Call function bfs for each child of cur -ent node.
Step 3 : Call bfs function for starting node of graph or tree.
Step 4 : Stop.

Flowchart : BFS

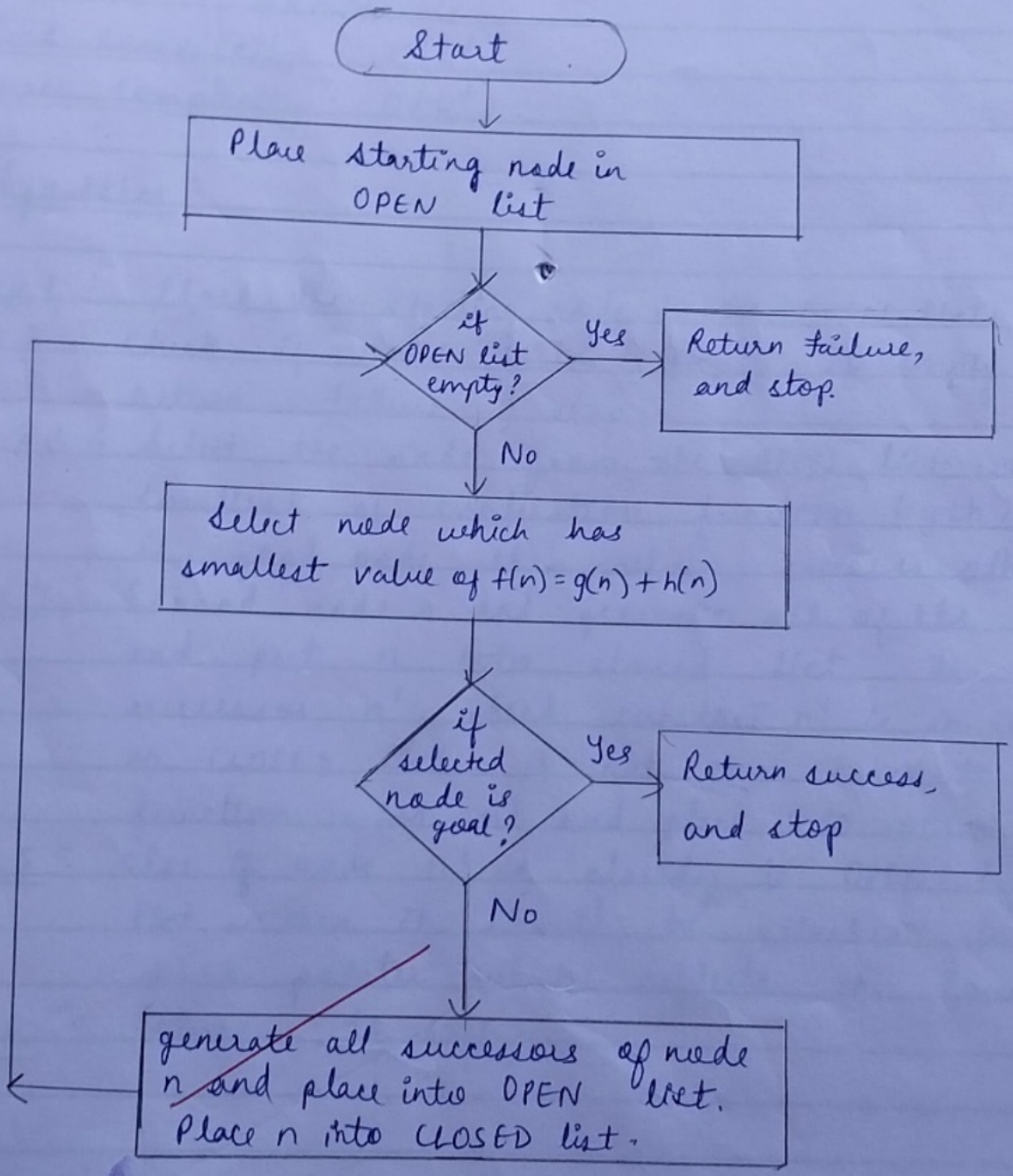


Flowchart : DFS



Date : 1/5/23

Flowchart: For A* Algorithm



Optimality : A* Search is complete. if h(n) is admissible.
Time complexity : $O(b^d)$
Space complexity : $O(b^d)$.

Algorithm :

Step 1 : Place the starting node in the OPEN list.
Step 2 : Check if OPEN list is empty, if empty then return failure & stop.
Step 3 : Select the node from the OPEN list which has smallest of evaluation function (g+h). if node n is goal node then return success otherwise,
Step 4 : Expand node n and generate all of its successor, and put n into closed list. for each successor n', check whether n' is 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 list, CLOSED list, then it should be attached to the back pointer which reflects the lowest g(n')
Step 6 : Return to step 2.

4) The feasible is the subset that satisfies the criteria particular problem & on the quantity of heuristic.

Selection Sort:

1) Selection Sort is a comparison sorting algorithm that is used to sort random list of item. While the second list that is used in ascending order.
2) The list is divided into two partitions the first contain sorted item while the second list contains unsorted items.
3) By default, the sorted list is empty & the unsorted list contain all the elements.
4) The unsorted list is when scanned for the minimum value which is when placed in the sorted list.

- Time Complexity - $O(n^2)$
- Space Complexity - $O(1)$

Algorithm:

Step1 : Get the value of n which is the total size of the array.

Step2 : Portion the list into sorted and unsorted section the sorted section is initially empty. While the unsorted selection contains the entire list.

Date :

Step 3 : Pick the minimum value from the unsorted selection and place it into the sorted selection.

Step 4 : Repeat the process (n-1) times until all of the elements in the list have been sorted.

Advantages

- It perform very well on small list.
- It is an in place algorithm, it does not Require a lot of space for sorting only the extra space is required for holding the temporal variable.
- It performs well on items that have already been sorted.

Disadvantages :

- It performs poorly working on big list.
- The number of iterations made having sorting is n- required where n is the number of elements in the list.
- Other algorithm such as quick sort have better performance compared to the Selection Sort.

Example -

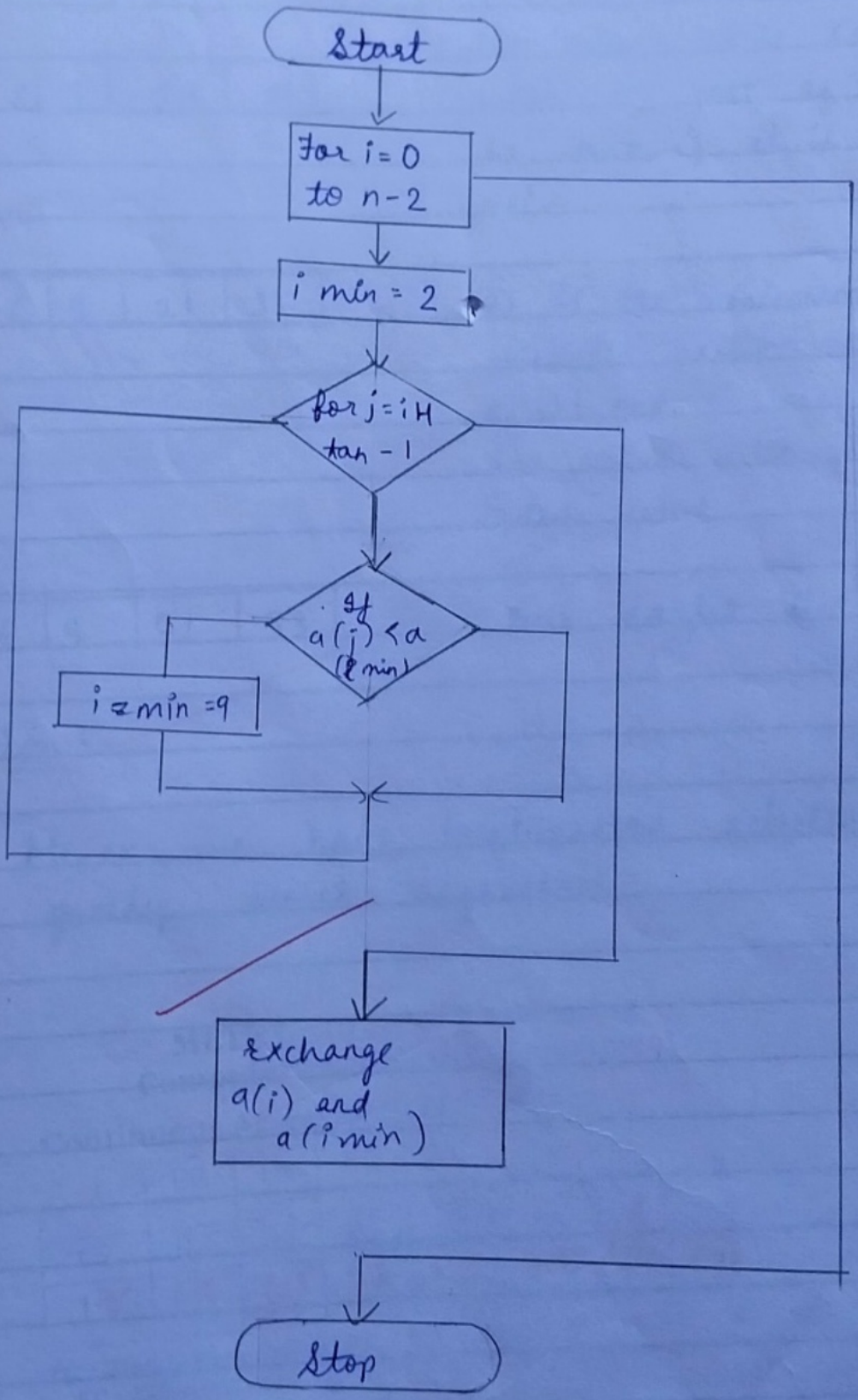
Unsorted list

21	6	9	33	3

Adgaon, Nashik - 422 003.

Flowchart : For Selection Sort

Date :



Algorithm:

Step 1 : Arrange the vertices of the graph in some order.

Step 2 : Choose the first vertex & color it with the first color.

Step 3 : Choose the next vertex and color it with the lowest numbered color that has not been colored on any vertices adjacent to it. If all the adjacent vertices are colored with this color, assign a new color to it. Repeat this step until all the vertices are colored.

Example : Algorithm

Step 1 : Greeting the customer.

Step 2 : Offer the service [Eg. main service
sub service]

Step 3 : Let customer select the service.

Step 4 : Check fact and apply the rule & provide the relevant answer.

Step 5 : Ask user whether to continue or exit.

Step 6 : If Yes repeat from step 2 to step 5.

Step 7 : Thankyou Message.

Step 8 : Exit.

Conclusion : We have studied the working of chatbot & implemented it for customer interaction application.

MET's Institute of Engineering
Computer Engineering Department

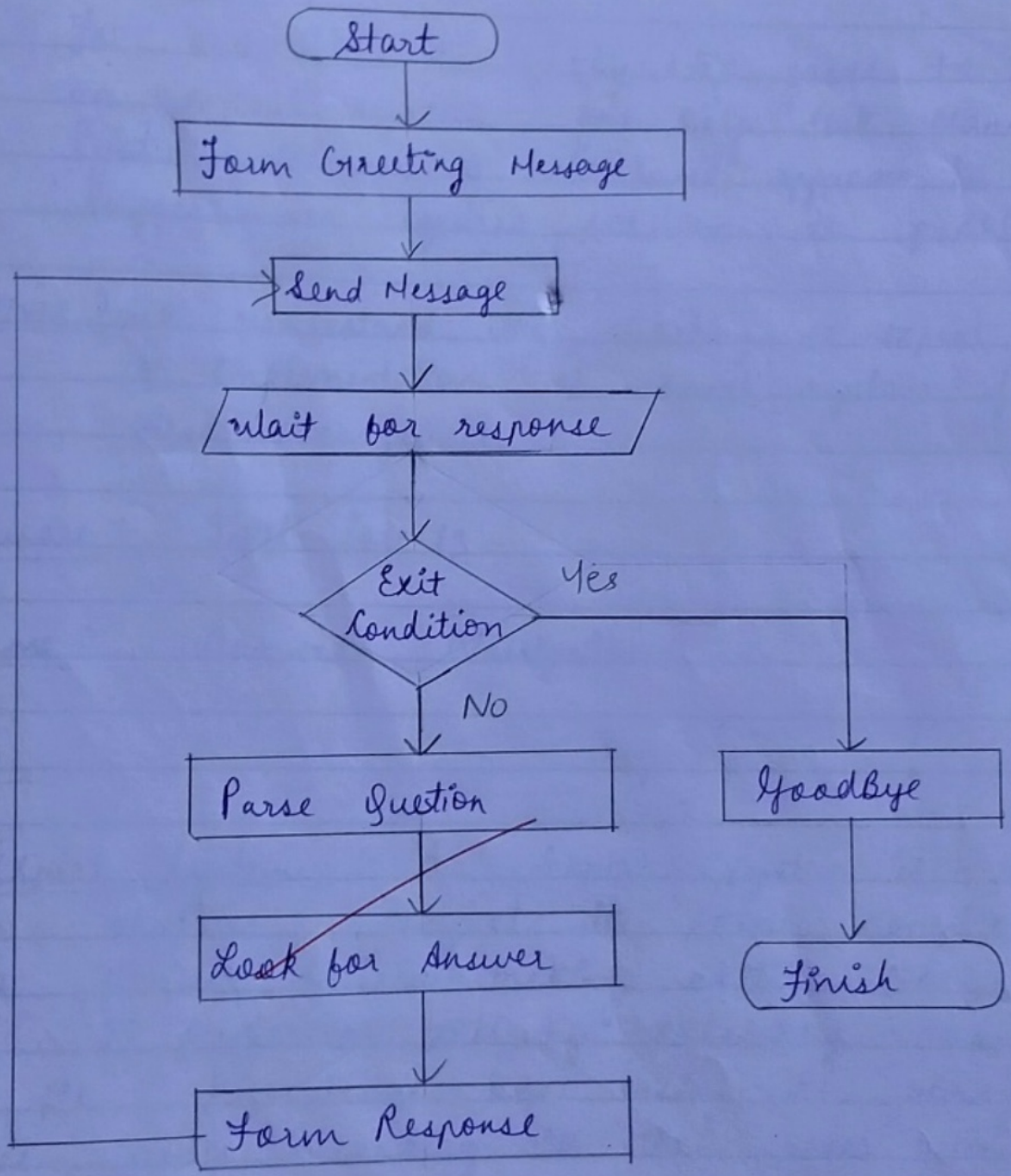
Continuous Assessment of Student :

TC	PR	IN	EC	PN	Total Marks	Faculty Signature
02	02	02	02	02	10	
1.5	1.5	1.5	1.5	1.5	7.5	

TC: Timely Completion, PR: Performance, IN: Innovation,
EC: Efficient Codes, PN: Punctuality and Neatness.

Date :

Flowchart: Greeting Chatbot



Flowchart : For Expert System

Date :