AIM Implementation of Linked List Data Structure

**CODE-**

#include <stdio.h> #include <stdlib.h>

struct node

{

int value;

struct node \*next;

};

struct node \*head, \*tail = NULL;

void addNode()

{

int value;

printf("Write the number to be added.\n"); scanf("%d",&value);

struct node \*newNode = (struct node\*)malloc(sizeof(struct node)); newNode->value = value;

newNode->next = NULL; if(head == NULL)

{

head = newNode; tail = newNode;

}

else

{

tail->next = newNode; tail = newNode;

}

printf("Successully added.\n");

}

int countNodes()

{

int count = 0;

struct node \*current = head;

while(current != NULL)

{

count++;

current = current->next;

}

return count;

}

void display()

{

struct node \*current = head; if(head == NULL)

{

printf("Empty List\n"); return;

}

printf("Nodes of linked list are: \n"); while(current != NULL)

{

printf("%d ", current->value); current = current->next;

}

printf("\n");

}

int main()

{

int opt; while(1)

{

printf("\n Choose any of the below options."); printf("\n 1 Adding a node.");

printf("\n 2 Counting number of nodes.");

printf("\n 3 Displaying all nodes.");

printf("\n 4 Close the program.");

printf("\n");

scanf("%d",&opt);

printf("\n");

switch(opt)

{

case 1:

addNode(); break;

case 2:

countNodes();

printf("NUmber of Node are: %d",countNodes()); break;

case 3:

display(); break;

case 4: exit(1); break;

default:

printf("Wrong Choice.\n");

}

}

}

## ALGORITHM

Step 1 - Include all the header files which are used in the program.

Step 2 - Declare all the user defined functions.

Step 3 - Define a Node structure with two members data and next

Step 4 - Define a Node pointer 'head' and set it to NULL.

Step 5 - Implement the main method by displaying operations menu and make suitable function calls in the main method to perform user selected operation.

OUTPUT-

