**Institute of Information Technology and Management**

**BCA – 174 (Data Structures using C Lab)**

**BCA SEM – II (M2)**

**External Practical Exam**

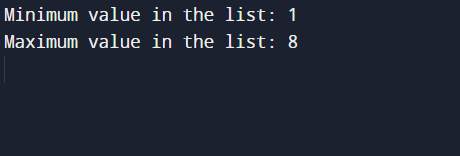
**SET A**

**Q1. Write a program in c language to create a linked list and show the maximum and minimum value in the list.**

**Ans.** CODE:

1. #include <stdio.h>
2. struct node{
3. int data;
4. struct node \*next;
5. };
6. struct node \*head, \*tail = NULL;
7. void addNode(int data) {
8. struct node \*newNode = (struct node\*)malloc(sizeof(struct node));
9. newNode->data = data;
10. newNode->next = NULL;
11. if(head == NULL) {
12. head = newNode;
13. tail = newNode;
14. }
15. else {
16. tail->next = newNode;
17. tail = newNode;
18. }
19. }
20. void minNode() {
21. struct node \*current = head;
22. int min;
23. if(head == NULL) {
24. printf("List is empty \n");
25. }
26. else {
27. min = head->data;
29. while(current != NULL){
30. if(min > current->data) {
31. min = current->data;
32. }
33. current= current->next;
34. }
35. printf("Minimum value in the list: %d\n", min);
36. }
37. }
38. void maxNode() {
39. struct node \*current = head;
40. int max;
42. if(head == NULL) {
43. printf("List is empty \n");
44. }
45. else {
46. max = head->data;
48. while(current != NULL){
49. if(max < current->data) {
50. max = current->data;
51. }
52. current = current->next;
53. }
54. printf("Maximum value in the list: %d\n", max);
55. }
56. }
58. int main()
59. {
60. addNode(5);
61. addNode(8);
62. addNode(1);
63. addNode(6);
64. minNode();
65. maxNode();
67. return 0;
68. }

OUTPUT:



**Q2. Write a program to delete elements from an array from beginning and end.**

**Ans.** CODE:

#include <stdio.h>

int main()

{

   int array[100], position, c, n;

   printf("Enter number of elements in array\n");

   scanf("%d", &n);

   printf("Enter %d elements\n", n);

   for (c = 0; c < n; c++)

      scanf("%d", &array[c]);

   printf("Enter the location where you wish to delete element\n");

   scanf("%d", &position);

   if (position >= n+1)

      printf("Deletion not possible.\n");

   else

   {

      for (c = position - 1; c < n - 1; c++)

         array[c] = array[c+1];

      printf("Resultant array:\n");

      for (c = 0; c < n - 1; c++)

         printf("%d\n", array[c]);

   }

   return 0;

}

OUTPUT:

