

C program to perform linear search using an array\

```
#include <stdio.h>
int main()
{
    printf("Enter the number of elements ");
    int n;
    scanf("%d", &n);
    printf("\nEnter the elements : ");
    int arr[n], x;
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    printf("\nEnter the element to search : ");
    scanf("%d", &x);

    for (int i = 0; i < n; i++)
        if (arr[i] == x)
        {
            printf("\nElement is found at %d", i);
            return 1;
        }
    printf("Element not found ");
    return 0;
}
```

Enter the number of elements 3

Enter the elements : 1 2 3

Enter the element to search : 2

Element is found at 1

C program to implement binary search using an array

```

C binarySearch.c > main()
1  #include <stdio.h>
2  int main()
3  {
4      printf("Enter the number of elements ");
5      int n;
6      scanf("%d", &n);
7      printf("\nEnter the elements : ");
8      int arr[n], x;
9      for (int i = 0; i < n; i++)
10         scanf("%d", &arr[i]);
11     printf("\nEnter the element to search : ");
12     scanf("%d", &x);
13     int low = 0, high = n;
14     int mid;
15     while (low <= high)
16     {
17         mid = (low + high) / 2;
18         if (arr[mid] == x)
19         {
20             printf("\nElement is found at %d", mid);
21             return 1;
22         }
23         else if (arr[mid] > x)
24             high = mid - 1;
25         else
26             low = mid + 1;
27     }
28     printf("Element not found ");
29     return 0;
30 }
31

```

Enter the number of elements 3

Enter the elements : 1 2 3

Enter the element to search : 3

Element is found at 2

C program to insert element into an array

```

C insertIntoAnArray.c > main()
1  #include <stdio.h>
2  int main()
3  {
4      printf("Enter the number of elements ");
5      int n;
6      scanf("%d", &n);
7      printf("\nEnter the elements : ");
8      int arr[n + 1], x, p;
9      for (int i = 0; i < n; i++)
10         scanf("%d", &arr[i]);
11     printf("\nEnter the element to insert : ");
12     scanf("%d", &x);
13     printf("\nEnter the position : ");
14     scanf("%d", &p);
15     for (int i = n; i >= p; i--)
16     {
17         arr[i + 1] = arr[i];
18     }
19     n++;
20     arr[p] = x;
21     printf("\nnew array is ");
22     for (int i = 0; i < n; i++)
23         printf("%d ", arr[i]);
24     return 0;
25 }

```

Enter the number of elements 3

Enter the elements : 1 2 4

Enter the element to insert : 3

Enter the position : 2

new array is 1 2 3 4

C program to delete an element from an array

```

C deleteFromArray.c > main()
1  #include <stdio.h>
2  int main()
3  {
4      printf("Enter the number of elements ");
5      int n;
6      scanf("%d", &n);
7      printf("\nEnter the elements : ");
8      int arr[n + 1], x;
9      for (int i = 0; i < n; i++)
10         scanf("%d", &arr[i]);
11     printf("\nEnter the element to delete : ");
12     scanf("%d", &x);
13     for (int i = n - 1; arr[i] != x && i > 0; i--)
14     {
15         arr[i - 1] = arr[i];
16     }
17     n--;
18     printf("\nnew array is ");
19     for (int i = 0; i < n; i++)
20         printf("%d ", arr[i]);
21     return 0;
22 }

```

Enter the number of elements 3

Enter the elements : 1 2 3

Enter the element to delete : 3

new array is 1 2

C program to merge two arrays

```

C mergeArrays.c > main()
1  #include <stdio.h>
2  int main()
3  {
4      printf("Enter the number of elements of 1st array: ");
5      int n;
6      scanf("%d", &n);
7
8      printf("\nEnter the number of elements of 2nd array ");
9      int m;
10     scanf("%d", &m);
11     printf("\nEnter the elements of 1st array: ");
12     int arr[n + m];
13     for (int i = 0; i < n; i++)
14         scanf("%d", &arr[i]);
15     printf("\nEnter the elements of 2nd array : ");
16     int b[n];
17     for (int i = 0; i < m; i++)
18         scanf("%d", &b[i]);
19     for (int i = 0; i < m; i++)
20         arr[n + i] = b[i];
21
22     for (int i = 0; i < m + n; i++)
23         printf("%d ", arr[i]);
24     return 0;
25 }

```

Enter the number of elements of 1st array: 3

Enter the number of elements of 2nd array 4

Enter the elements of 1st array: 1 2 3

Enter the elements of 2nd array : 4 5 6 7

1 2 3 4 5 6 7

(khan) PS D:\New

C program to insert an element in a linked list

C insertionInSinglyLinkedList.c > node

```
1  #include <stdio.h>
2
3  struct node
4  {
5      int ele;
6      struct node *next;
7  };
8
9  typedef struct node *LIST;
10 typedef LIST L;
11
12 L ListInsert(int num, L p)
13 {
14     L temp;
15     temp = (LIST)malloc(sizeof(LIST));
16     temp->ele = num;
17     if (p->next != NULL)
18         temp->next = p->next;
19     else
20         temp->next = NULL;
21     p->next = temp;
22     return p;
23 }
24 L findInsert(int num, L h)
25 {
26     L temp;
27     temp = h;
28     while (temp->next != NULL && temp->ele != num)
29     {
30         temp = temp->next;
31     }
32     return temp;
33 }
34
35 int main()
36 {
37     printf("Enter the number of elements ");
38     int n;
39     scanf("%d", &n);
40     printf("\nEnter the elements : ");
41     int arr[n], x;
42     L head;
43     head = (LIST)malloc(sizeof(LIST));
44     head->ele = 0;
45     head->next = NULL;
```

```

46     L p = head;
47     for (int i = 0; i < n; i++)
48     {
49         scanf("%d", &x);
50         p = ListInsert(x, p);
51         p = p->next;
52     }
53
54     p = head;
55     p = p->next;
56     while (p != NULL)
57     {
58         printf("%d ", p->ele);
59         p = p->next;
60     }
61
62     int y;
63     printf("\nEnter the element to be inserted : ");
64     scanf("%d", &x);
65     printf("\nAfter which element: ");
66     scanf("%d", &y);
67
68     p = findInsert(y, head);
69     p = ListInsert(x, p);
70     p = head->next;
71     printf("\nAfter insertion : ");
72     while (p != NULL)
73     {
74         printf("%d ", p->ele);
75         p = p->next;
76     }
77     return 0;
78 }

```

Enter the number of elements 3

Enter the elements : 1 2 3

1 2 3

Enter the element to be inserted : 4

After which element: 2

After insertion : 1 2 4 3