C program to perform linear search using an array\

```c
#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

binarySearch.c > main()

```c
#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);
    int low = 0, high = n;
    int mid;
    while (low <= high)
    {
        mid = (low + high) / 2;
        if (arr[mid] == x)
        {
            printf("\nElement is found at %d", mid);

            return 1;
        }
        else if (arr[mid] > x)
            high = mid - 1;
        else
            low = mid + 1;
    }
    printf("Element not found ");
    return 0;
}
```

```
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

insertIntoAnArray.c > main()

```c
#include <stdio.h>
int main()
{
    printf("Enter the number of elements ");
    int n;
    scanf("%d", &n);
    printf("\nEnter the elements : ");
    int arr[n + 1], x, p;
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    printf("\nEnter the element to insert : ");
    scanf("%d", &x);
    printf("\nEnter the position : ");
    scanf("%d", &p);
    for (int i = n; i >= p; i--)
    {
        arr[i + 1] = arr[i];
    }
    n++;
    arr[p] = x;
    printf("\nnew array is ");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    return 0;
}
```

```
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

deleteFromAnArray.c > main()

```c
#include <stdio.h>
int main()
{
    printf("Enter the number of elements ");
    int n;
    scanf("%d", &n);
    printf("\nEnter the elements : ");
    int arr[n + 1], x;
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    printf("\nEnter the element to delete : ");
    scanf("%d", &x);
    for (int i = n - 1; arr[i] != x && i > 0; i--)
    {
        arr[i - 1] = arr[i];
    }
    n--;
    printf("\nnew array is ");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    return 0;
}
```

```
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

mergeArrays.c > main()

```c
#include <stdio.h>
int main()
{
    printf("Enter the number of elements of 1st array: ");
    int n;
    scanf("%d", &n);

    printf("\nEnter the number of elements of 2nd array ");
    int m;
    scanf("%d", &m);
    printf("\nEnter the elements of 1st array: ");
    int arr[n + m];
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    printf("\nEnter the elements of 2nd array : ");
    int b[n];
    for (int i = 0; i < m; i++)
        scanf("%d", &b[i]);
    for (int i = 0; i < m; i++)
        arr[n + i] = b[i];

    for (int i = 0; i < m + n; i++)
        printf("%d ", arr[i]);
    return 0;
}
```

```
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
```

C program to insert an element in a linked list

C insertionInSinglyLinkedList.c > node

```c
#include <stdio.h>

struct node
{
    int ele;
    struct node *next;
};

typedef struct node *LIST;
typedef LIST L;

L ListInsert(int num, L p)
{
    L temp;
    temp = (LIST)malloc(sizeof(LIST));
    temp->ele = num;
    if (p->next != NULL)
        temp->next = p->next;
    else
        temp->next = NULL;
    p->next = temp;
    return p;
}
L findInsert(int num, L h)
{
    L temp;
    temp = h;
    while (temp->next != NULL && temp->ele != num)
    {
        temp = temp->next;
    }
    return temp;
}

int main()
{
    printf("Enter the number of elements ");
    int n;
    scanf("%d", &n);
    printf("\nEnter the elements : ");
    int arr[n], x;
    L head;
    head = (LIST)malloc(sizeof(LIST));
    head->ele = 0;
    head->next = NULL;
```

```
    L p = head;
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &x);
        p = ListInsert(x, p);
        p = p->next;
    }

    p = head;
    p = p->next;
    while (p != NULL)
    {
        printf("%d ", p->ele);
        p = p->next;
    }

    int y;
    printf("\nEnter the element to be inserted : ");
    scanf("%d", &x);
    printf("\nAfter which element: ");
    scanf("%d", &y);

    p = findInsert(y, head);
    p = ListInsert(x, p);
    p = head->next;
    printf("\nAfter insertion : ");
    while (p != NULL)
    {
        printf("%d ", p->ele);
        p = p->next;
    }
    return 0;
}
```

```
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
```