

```
#include <stdio.h>
void printArray(int* arr, int n)
{
    int i;
    printf("Array: ");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
int main()
{
    int arr[] = { 2, -1, 5, 6, 0, -3 };
    int n = sizeof(arr) / sizeof(arr[0]);
    printArray(arr, n);
    return 0;
}
```

```
#include <stdio.h>
int findElement(int arr[], int n, int key)
{
    int i;
    for (i = 0; i < n; i++)
        if (arr[i] == key)
            return i;
    return -1;
}
int main()
{
    int arr[] = { 12, 34, 10, 6, 40 };
    int n = sizeof(arr) / sizeof(arr[0]);
    int key = 40;
    int position = findElement(arr, n, key);
    if (position == -1)
        printf("Element not found");
    else
        printf("Element Found at Position: %d",
            position + 1);
    return 0;
}
```

```
#include <stdio.h>
int main()
{
    int arr[100] = { 0 };
    int i, x, pos, n = 10;
    for (i = 0; i < 10; i++)
        arr[i] = i + 1;
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
    x = 50;
    pos = 5;
    n++;
    for (i = n - 1; i >= pos; i--)
        arr[i] = arr[i - 1];
    arr[pos - 1] = x;
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
    return 0;
}
```

```

#include <stdio.h>
int findElement(int arr[], int n, int key);
int deleteElement(int arr[], int n, int key){
    int pos = findElement(arr, n, key);
    if (pos == -1) {
        printf("Element not found");
        return n;
    }
    int i;
    for (i = pos; i < n - 1; i++)
        arr[i] = arr[i + 1];
    return n - 1;
}
int findElement(int arr[], int n, int key)
{
    int i;
    for (i = 0; i < n; i++)
        if (arr[i] == key)
            return i;
    return -1;
}
int main()
{
    int i;
    int arr[] = { 10, 50, 30, 40, 20 };
    int n = sizeof(arr) / sizeof(arr[0]);
    int key = 30;
    printf("Array before deletion\n");
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    n = deleteElement(arr, n, key);
    printf("\nArray after deletion\n");
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    return 0;
}

```

```
#include <stdio.h>
int main()
{
    int arr_int[5];
    char arr_char[5];
    return 0;
}
```

```
#include<stdio.h>
long int multiplyNumbers(int n);
int main() {
    int n;
    printf("Enter a positive integer: ");
    scanf("%d",&n);
    printf("Factorial of %d = %ld", n, multiplyNumbers(n));
    return 0;
}
long int multiplyNumbers(int n) {
    if (n>=1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}
```

```
#include <stdio.h>
int findDuplicate(int arr[])
{
    int slow = arr[0];
    int fast = arr[0];
    do {
        slow = arr[slow];
        fast = arr[arr[fast]];
    } while (slow != fast);

    int ptr1 = arr[0];
    int ptr2 = slow;
    while (ptr1 != ptr2) {
        ptr1 = arr[ptr1];
        ptr2 = arr[ptr2];
    }
    return ptr1;
}
int main()
{
    int arr[] = { 1, 3, 2, 1 };
    printf("%d", findDuplicate(arr));
    return 0;
}
```

```
#include <stdio.h>
#define MAX_SIZE 100
int main()
{
    int arr[MAX_SIZE];
    int i, max, min, size;
    printf("Enter size of the array: ");
    scanf("%d", &size);
    printf("Enter elements in the array: ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }
    max = arr[0];
    min = arr[0];
    for(i=1; i<size; i++)
    {
        if(arr[i] > max)
        {
            max = arr[i];
        }
        if(arr[i] < min)
        {
            min = arr[i];
        }
    }
    printf("Maximum element = %d\n", max);
    printf("Minimum element = %d", min);
    return 0;
}
```



```
#include <stdio.h>
void printFib(int n)
{
    if (n < 1) {
        printf("Invalid Number of terms\n");
        return;
    }
    int prev1 = 1;
    int prev2 = 0;
    for (int i = 1; i <= n; i++) {
        if (i > 2) {
            int num = prev1 + prev2;
            prev2 = prev1;
            prev1 = num;
            printf("%d ", num);
        }
        if (i == 1) {
            printf("%d ", prev2);
        }
        if (i == 2) {
            printf("%d ", prev1);
        }
    }
}

int main()
{
    int n = 9;
    printFib(n);
    return 0;
}
```

```
#include<stdio.h>

int binarySearch(int arr[], int l, int r, int x)
{
    if (r >= l) {
        int mid = l + (r - l) / 2;

        if (arr[mid] == x)
            return mid;

        if (arr[mid] > x) {
            return binarySearch(arr, l, mid - 1, x);
        }

        return binarySearch(arr, mid + 1, r, x);
    }
    return -1;
}

int main(void)
{
    int arr[] = { 2, 3, 4, 10, 40 };
    int size = sizeof(arr) / sizeof(arr[0]);
    int x = 10;
    int index = binarySearch(arr, 0, size - 1, x);

    if (index == -1) {
        printf("Element is not present in array");
    }
    else {
        printf("Element is present at index %d", index);
    }
    return 0;
}
```

```
#include <stdio.h>
int linearSearch(int* arr, int size, int key)
{
    for (int i = 0; i < size; i++) {
        if (arr[i] == key) {
            return i;
        }
    }
    return -1;
}

int main()
{
    int arr[10] = { 3, 4, 1, 7, 5, 8, 11, 42, 3, 13 };
    int size = sizeof(arr) / sizeof(arr[0]);
    int key = 4;
    int index = linearSearch(arr, size, key);
    if (index == -1) {
        printf("The element is not present in the arr.");
    }
    else {
        printf("The element is present at arr[%d].", index);
    }
    return 0;
}
```