

```

// Q1. Find the missing element from the array
#include <iostream>
#include <cstdio>
using namespace std;
int Solution(int arr[], int n)
{
    // Sorting the array first
    int temp;
    for (int i = 0; i < n; ++i)
    {
        temp = 0;
        for (int j = 0; j < n - i - 1; ++j)
        {
            if (arr[j] > arr[j + 1])
            {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
    cout << "\nThe Sorted array is:\n";
    for (int i = 0; i < n; ++i)
    {
        cout << arr[i] << " ";
    }
    // Comparing & finding missing element
    for (int i = 1; i <= n; ++i)
    {
        if (i != arr[i])
        {
            return i;
        }
    }
}
int main()
{
    int n;
    cout << "Enter a size of array: ";
    cin >> n;
    // Array of Natural Numbers
    int array[n];

```

```

    cout << "Enter the elements of the array\n";
    for (int i = 0; i < n; ++i)
    {
        cout << i + 1 << ": ";
        cin >> array[i];
    }
    cout << "\nThe array is:\n";
    for (int i = 0; i < n; ++i)
    {
        cout << array[i] << " ";
    }
    int result = Solution(array, n);
    cout << "\nThe missing element in your array is:: ";
    cout << result << endl;

    return 0;
}

```

OUTPUT:

```

Enter a size of array: 5
Enter the elements of the array
1: 2
2: 3
3: 4
4: 6
5: 5

```

The array is:

```
2 3 4 6 5
```

The Sorted array is:

```
2 3 4 5 6
```

The missing element in your array is:: 1

// Q2. Write a function rotate(arr[], d, n) that rotates arr[] of size n by d elements

```

#include <iostream>
#include <cstdio>
using namespace std;
int Rotate(int arr[], int n, int r)
{
    int temp;

```

```

    for(int i=0;i<r;i++)
    {
        temp=arr[0];
        for(int j=0;j<n-1;++j)
        {
            arr[j]=arr[j+1];
        }
        arr[n-1]=temp;
    }
}
int main()
{
    int n;
    cout << "Enter a size of array: ";
    cin >> n;
    int array[n];
    cout << "Enter the elements of the array\n";
    for (int i = 0; i < n; ++i)
    {
        cout << i + 1 << ": ";
        cin >> array[i];
    }
    cout << "\nThe array is:\n";
    for (int i = 0; i < n; ++i)
    {
        cout << array[i] << " ";
    }
    cout << "\nEnter the number of rotations: ";
    int rotate;
    cin>>rotate;
    Rotate(array,n,rotate);
    cout << "\nThe Rotated array is:\n";
    for (int i = 0; i < n; ++i)
    {
        cout << array[i] << " ";
    }

    return 0; }

```

OUTPUT:

Enter a size of array: 5  
Enter the elements of the array  
1: 1  
2: 2  
3: 3  
4: 4  
5: 5

The array is:

1 2 3 4 5

Enter the number of rotations: 2

The Rotated array is:

3 4 5 1 2

```
#include <iostream>
#include <cstdio>
using namespace std;
```

```
// Q3. Write a function rotate(arr[], d, n) that rotates
arr[] of size n by d elements using recursion
```

```
static int k = 0;
int Rotate(int arr[], int n, int r)
{
    int temp;
    if (k < r)
    {
        temp = arr[0];
        for (int j = 0; j < n - 1; ++j)
        {
            arr[j] = arr[j + 1];
        }
        arr[n - 1] = temp;
    }
    else{
        k++;
        return Rotate(arr,n,r);
    }
}
```

```

    }
}
int main()
{
    int n;
    cout << "Enter a size of array: ";
    cin >> n;
    int array[n];
    cout << "Enter the elements of the array\n";
    for (int i = 0; i < n; ++i)
    {
        cout << i + 1 << ": ";
        cin >> array[i];
    }
    cout << "\nThe array is:\n";
    for (int i = 0; i < n; ++i)
    {
        cout << array[i] << " ";
    }
    cout << "\nEnter the number of rotations: ";
    int rotate;
    cin >> rotate;
    Rotate(array, n, rotate);
    cout << "\nThe Rotated array is:\n";
    for (int i = 0; i < n; ++i)
    {
        cout << array[i] << " ";
    }

    return 0;
}

```

OUTPUT :

Enter a size of array: 5

Enter the elements of the array

1: 1

2: 2

3: 3

4: 4

5: 5

The array is:

1 2 3 4 5

Enter the number of rotations: 1

The Rotated array is:

2 3 4 5 1

// Q4.WAP Segregate 0s and 1s in an array

```
#include <iostream>
using namespace std;
```

```
void Filter(int arr[], int n)
{
    int count = 0;

    for (int i = 0; i < n; i++)
        if (arr[i] == 0)
            count++;
    for (int i = 0; i < count; i++)
        arr[i] = 0;
    for (int i = count; i < n; i++)
        arr[i] = 1;
}

void print(int arr[], int n)
{
    cout << "Array after Filter is ";

    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
}

int main()
{
    int arr[] = {0, 1, 0, 1, 1, 1};
    int n = sizeof(arr) / sizeof(arr[0]);
    cout<<"The array is: "<<endl;
```

```

        for (int i = 0; i < n; i++)
            cout << arr[i] << " ";

        Filter(arr, n);
        print(arr, n);
        return 0;
    }

```

OUTPUT:

The array is:

0 1 0 1 1 1

Array after Filter is

0 0 1 1 1 1

```

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

```

// Q5. Binary Search of a Sorted Array

```

int BinarySearch(int arr[], int size, int element)
{
    int low = 0, high = size - 1, mid = (low + high) / 2;
    while (low <= high)
    {
        mid = (low + high) / 2;
        if (arr[mid] == element)
        {
            return mid;
        }
        if (arr[mid] < element)
        {
            low = mid + 1;
        }
        else
        {
            high = mid - 1;
        }
    }
    return -1;
}

```

// Sorting the array

```

void Sort(int arr[], int size)
{
    int temp;
    for (int i = 0; i < size; i++)
    {
        temp = 0;
        for (int j = 0; j < size - i - 1; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

void Display(int arr[], int size_arr)
{
    for (int i = 0; i < size_arr; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main()
{
    int array[] = {12, 3, 5, 4, 67, 14, 32, 15, 42};
    int size = sizeof(array) / sizeof(int), element;
    printf("The array is: ");
    Display(array, size);

    printf("The Sorted array is: ");
    Sort(array, size);
    Display(array, size);

    printf("Enter a element to search: ");
    scanf("%d", &element);
    int x = BinarySearch(array, size, element);
    if (x == -1)

```



```
{  
    printf("Element Not Found\n");  
}  
else  
{  
    printf("The element %d found at index %d\n", element, x);  
}  
return 0;  
}
```

OUTPUT:

The Sorted array is: 3 4 5 12 14 15 32 42 67

Enter a element to search: 14

The element 14 found at index 4

The array is: 12 3 5 4 67 14 32 15 42

The Sorted array is: 3 4 5 12 14 15 32 42 67

Enter a element to search: 22

Element Not Found