Rajbahadur Meena
2022BCSE078

Week 1
26-02-2024 (B-Section)

Arrays and Structures

Aim

Use the following concepts from C/C++ to solve the given programming problems.

- ***Arrays*:** An array is a homogenous collection of elements of the same data type, arranged in contiguous memory locations. It is used to store multiple values of the same data type under a single name.
- ***Structures*:** A structure is a user-defined data type that allows you to group together variables of different data types (heterogenous collection) under a single name. It is used to represent a record or a collection of related data items.

Problem Statements

1. Find the 'nth' largest and smallest element in an array.

```cpp
#include <iostream>
using namespace std;
int nthSmallest(int arr[], int size, int n) {
    sort(arr, arr + size);
    return arr[n - 1];
}
int nthLargest(int arr[], int size, int n) {
    sort(arr, arr + size, greater<int>());
    return arr[n - 1];
}
int main() {
    int arr[] = {12, 45, 23, 51, 19, 8};
    int size = sizeof(arr) / sizeof(arr[0]);
    int n = 4;
    cout << "Original Array: ";
    for (int i = 0; i < size; ++i)
        cout << arr[i] << " ";
    cout << endl;
    cout << n << "rd smallest element: " << nthSmallest(arr, size, n) << endl;
    cout << n << "rd largest element: " << nthLargest(arr, size, n) << endl;
    return 0;
```

Output :

```
Original Array: 12 45 23 51 19 8
4rd smallest element: 23
4rd largest element: 19
```

2. Implement left shift with given number of steps.

```cpp
#include <iostream>
using namespace std;
void leftRotate(int arr[], int n, int d)
{
    d = d % n;
    int temp[d];
    for (int i = 0; i < d; ++i)
        temp[i] = arr[i];
    for (int i = d; i < n; ++i)
        arr[i - d] = arr[i];
    for (int i = 0; i < d; ++i)
        arr[n - d + i] = temp[i];
}
void printArray(int arr[], int n)
{
    for (int i = 0; i < n; ++i)
        cout << arr[i] << " ";
    cout << endl;
}
int main()
{
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    int steps = 3;
    cout << "Original Array: ";
    printArray(arr, n);
    leftRotate(arr, n, steps);
    cout << "Array after " << steps << " left rotations: ";
    printArray(arr, n);
    return 0;
}
```

Output:

```
Original Array: 1 2 3 4 5
Array after 3 left rotations: 4 5 1 2 3
```

3. Implement cyclic right rotate with given number of steps.

```cpp
#include <iostream>
using namespace std;
void cyclicRightRotate(int arr[], int n, int d)
{
    d = d % n;
    int temp[d];
    for (int i = n - d; i < n; ++i)
        temp[i - (n - d)] = arr[i];
    for (int i = n - 1; i >= d; --i)
        arr[i] = arr[i - d];
    for (int i = 0; i < d; ++i)
        arr[i] = temp[i];
}
void printArray(int arr[], int n)
{
    for (int i = 0; i < n; ++i)
        cout << arr[i] << " ";
    cout << endl;
}
int main()
{
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    int steps = 2;
    cyclicRightRotate(arr, n, steps);
    printArray(arr, n);
    return 0;
}

```

Output:

4. Implement Intersection of two arrays.

```cpp
#include <iostream>
using namespace std;
void intersection(int arr1[], int arr2[], int m, int n) {
    int i = 0, j = 0;
    while (i < m && j < n) {
        if (arr1[i] < arr2[j]) {
            i++;
        } else if (arr2[j] < arr1[i]) {
            j++;
        } else {
            cout << arr2[j] << " ";
            i++;
            j++;
        }
    }
}
int main() {
    int arr1[] = {1, 2, 2, 3, 4};
    int arr2[] = {2, 2, 4, 6};
    int m = sizeof(arr1) / sizeof(arr1[0]);
    int n = sizeof(arr2) / sizeof(arr2[0]);
    cout << "Intersection: ";
    intersection(arr1, arr2, m, n);
    cout << endl;
    return 0;
}
```

Click to collapse the range.

Output :

```
Intersection: 2 2 4
```

5. Reverse a given array.

```cpp
#include<iostream>
using namespace std;
void reverse(int arr[] , int n)
{
    int start=0;
    int end=n-1;
    while(start<=end)
    {
        swap(arr[start] , arr[end]);
        start++;
        end--;
    }
}
void printArray(int arr[], int n)
{
    for(int i=0;i<n;i++)
    {
        cout<<arr[i]<<" ";
    }
    cout<<endl;
}
int main()
{
    int arr[6]={1,2,3,4,5,6};
    reverse (arr,6);
    printArray(arr,6);

}
```

Output :

```
6 5 4 3 2 1
```