***Name: Muhammad Uzair***

***Roll No. L1F18BSCS0311***

***Section-F***

**Every Question should be implemented as a function with input array and size [if given] as parameter, no main and any other work is needed.**

**Question # 1**

You have an array A of size N populated with three different integer values in increasing sorting order. Every integer value is there for at-least one time. You have to design a function that will find the middle value of the three and then get the count of that middle value in this array.   
  
**You can’t count the integer values in this array.  
You can’t copy integer values to another array.**   
**You have to design generic solution.   
  
void find\_value\_count(int A[], int N);**

**A: {1, 1, 1, 1, 7, 7, 7, 8, 8, 8, 8, 8}  
middle value is : 7  
Its count is : 3  
  
A: {11, 27, 27, 27, 27, 27, 27, 27, 8}  
middle value is : 27  
Its count is : 7  
  
A: {3, 3, 3, 3, 3, 3, 3, 3, 3, 14, 33, 33, 33, 33, 33, 33}  
middle value is : 14  
Its count is : 1**

**Solution**

**void find\_value\_count(int arr[], int size)**

**{**

**int start\_index = find\_starting\_index(arr, 0, 1, arr[0]);**

**int last\_index = find\_last\_index(arr, size - 1, size - 2, arr[size - 1]);**

**cout << "Middle Value is: " << arr[start\_index] << endl;**

**int count = last\_index - start\_index + 1;**

**cout << "Its count is: " << count << endl;**

**}**

**int find\_starting\_index(int arr[], int start, int end, int val)**

**{**

**if (arr[end - 1] != val)**

**{**

**return find\_starting\_index(arr, start, (start + end) / 2, val);**

**}**

**else if (arr[end] == val)**

**{**

**return find\_starting\_index(arr, end, end \* 2, val);**

**}**

**else**

**{**

**return end;**

**}**

**}**

**int find\_last\_index(int arr[], int start, int end, int val)**

**{**

**if (arr[end + 1] != val)**

**{**

**return find\_last\_index(arr, start, (start + end) / 2, val);**

**}**

**else if (arr[end] == val)**

**{**

**return find\_last\_index(arr, end, end / 2, val);**

**}**

**else**

**{**

**return end;**

**}**

**}**

**Question # 2**You have an array A of size N populated with four different integer values [at-least every value is there for once]. Your task is to design a function that will find all the four integer values present in this array.   
  
**You can access this array for once.  
You have to design generic solution.   
  
void find\_values(int A[], int N);**

**Solution**

**void find\_value(int A[], int N)**

**{**

**int diff\_values[4] = {0,0,0,0};**

**int i = 0;**

**for (int index = 0; index < N && i != 4; index++)**

**{**

**if (i == 1)**

**{**

**if (diff\_values[0] != A[index])**

**{**

**diff\_values[i] = A[index];**

**i++;**

**}**

**}**

**else if (i == 2)**

**{**

**if (diff\_values[0] != A[index] && diff\_values[1] != A[index])**

**{**

**diff\_values[i] = A[index];**

**i++;**

**}**

**}**

**else if (i == 3)**

**{**

**if (diff\_values[0] != A[index] && diff\_values[1] != A[index] && diff\_values[2] != A[index])**

**{**

**diff\_values[i] = A[index];**

**i++;**

**}**

**}**

**else**

**{**

**diff\_values[i] = A[i];**

**i++;**

**}**

**}**

**for (int j = 0; j < i; j++)**

**{**

**cout << diff\_values[j] << " ";**

**}**

**}**

**Question # 3**

You have an array A of size N populated with integer values from the set {1, 2, 3 and 4}. Your task is to design a function that will sort the array in increasing order.   
  
**You can’t count the integer values in this array.  
You can’t copy integer values to another array.**   
**You have to design generic solution.   
  
void sort\_values(int A[], int N);**

**Solution**

**void sort\_values(int A[], int N)**

**{**

**int temp = 0;**

**for(int index = 0; index < N; index++)**

**{**

**if (A[index] == 1)**

**{**

**swap(A[index], A[temp]);**

**temp++;**

**}**

**}**

**for(int index = 0; index < N; index++)**

**{**

**if (A[index] == 2)**

**{**

**swap(A[index], A[temp]);**

**temp++;**

**}**

**}**

**for(int index = 0; index < N; index++)**

**{**

**if (A[index] == 3)**

**{**

**swap(A[index], A[temp]);**

**temp++;**

**}**

**}**

**for (int i = 0; i < N; i++)**

**{**

**cout << A[i] << " ";**

**}**

**}**