**Assignment - 15 A Job Ready Bootcamp in C++, DSA and IOT MySirG**

**Array and Functions in C Language**

1. Write a function to find the greatest number from the given array of any size. (TSRS)

#include <stdio.h>

int findGreatest(int arr[], int size) {

int i, max = arr[0];

for (i = 1; i < size; i++) {

if (arr[i] > max) {

max = arr[i];

}

}

return max;

}

int main() {

int arr[] = {3, 5, 2, 8, 1, 9, 4};

int size = sizeof(arr) / sizeof(arr[0]);

int greatest = findGreatest(arr, size);

printf("The greatest number is: %d", greatest);

return 0;

2. Write a function to find the smallest number from the given array of any size. (TSRS)

#include <stdio.h>

int findSmallest(int arr[], int size) {

int smallest = arr[0];

for (int i = 1; i < size; i++) {

if (arr[i] < smallest) {

smallest = arr[i];

}

}

return smallest;

}

int main() {

int arr[] = {5, 1, 9, 3, 7, 2};

int size = sizeof(arr) / sizeof(arr[0]);

int smallest = findSmallest(arr, size);

printf("The smallest number is: %d\n", smallest);

}

3. Write a function to sort an array of any size. (TSRS)

printf("%d ", arr[i]); // Print each element

}

printf("\n");

return 0;

}

4. Write a function to rotate an array by n position in d direction. The d is an indicative

value for left or right. (For example, if array of size 5 is [32, 29, 40, 12, 70]; n is 2 and

d is left, then the resulting array after left rotation 2 times is [40, 12, 70, 32, 29] ).

#include <stdio.h>

void rotateArray(int arr[], int size, int n, char d) {

int temp;

int i, j;

if (d == 'L') {

for (i = 0; i < n; i++) {

temp = arr[0];

for (j = 0; j < size - 1; j++) {

arr[j] = arr[j + 1];

}

arr[size - 1] = temp;

}

} else if (d == 'R') {

for (i = 0; i < n; i++) {

temp = arr[size - 1];

for (j = size - 1; j > 0; j--) {

arr[j] = arr[j - 1];

}

arr[0] = temp;

}

} else {

printf("Invalid direction. Please enter 'L' for left or 'R' for right.\n");

}

}

int main() {

int arr[] = {32, 29, 40, 12, 70};

int size = sizeof(arr) / sizeof(arr[0]);

int n = 2;

char d = 'L';

printf("Original array:\n");

for (int i = 0; i < size; i++) {

printf("%d ", arr[i]);

}

printf("\n");

rotateArray(arr, size, n, d);

printf("Rotated array:\n");

for (int i = 0; i < size; i++) {

printf("%d ", arr[i]);

}

printf("\n");

return 0;

}

5. Write a function to find the first occurrence of adjacent duplicate values in the array.

Function has to return the value of the element.

#include <stdio.h>

int findAdjacentDuplicates(int arr[], int size) {

for (int i = 1; i < size; i++) {

if (arr[i] == arr[i - 1]) {

return arr[i];

}

}

return -1;

}

int main() {

int arr[] = {2, 3, 4, 4, 5, 6, 6, 7, 8};

int size = sizeof(arr) / sizeof(arr[0]);

int adjacentDuplicate = findAdjacentDuplicates(arr, size);

if (adjacentDuplicate != -1) {

printf("The first adjacent duplicate value is %d\n", adjacentDuplicate);

} else {

printf("No adjacent duplicates found\n");

return 0;

}

6. Write a function in C to read n number of values in an array and display it in reverse

order.

#include <stdio.h>

void displayReverse(int arr[], int size) {

printf("Array in reverse order: ");

for (int i = size - 1; i >= 0; i--) {

printf("%d ", arr[i]);

}

printf("\n");

}

int main() {

int arr[100], n;

printf("Enter the number of elements: ");

scanf("%d", &n);

printf("Enter %d elements:\n", n);

for (int i = 0; i < n; i++) {

scanf("%d", &arr[i]);

}

printf("Array in original order: ");

for (int i = 0; i < n; i++) {

printf("%d ", arr[i]);

}

printf("\n");

displayReverse(arr, n);

return 0;

}

7. Write a function in C to count a total number of duplicate elements in an array.

#include <stdio.h>

int duplicateNumber(int arr[], int size)

{

int count = 0;

for (int i = 0; i < size; i++) {

for (int j = i+1; j < size; j++) {

if (arr[i] == arr[j]) {

count++;

}

}

}

return count;

}

int main()

{

int arr[] = {1, 2, 3, 4, 3, 5, 1, 2};

int size = sizeof(arr) / sizeof(arr[0]);

int count = duplicateNumber(arr, size);

printf("%d\n", count);

return 0;

}

8. Write a function in C to print all unique elements in an array.

#include <stdio.h>

int uniqNumber(int arr[], int size)

{

int uniq = 0;

for (int i = 0; i < size; i++) {

int isUnique = 1;

for (int j = i+1; j < size; j++) {

if (arr[i] == arr[j]) {

isUnique = 0;

break;

}

}

if (isUnique) {

uniq++;

}

}

return uniq;

}

int main()

{

int arr[] = {6,2,5,2,2,6,6,9,7};

int size = sizeof(arr) / sizeof(arr[0]);

int uniq = uniqNumber(arr, size);

printf("%d\n", uniq);

return 0;

}

9. Write a function in C to merge two arrays of the same size sorted in descending

order.

#include <stdio.h>

void mergeArrays(int arr1[], int arr2[], int size, int merged[]) {

int i = 0,j = 0,k = 0;

while (i < size && j < size) {

if (arr1[i] > arr2[j]) {

merged[k] = arr1[i];

i++;

} else {

merged[k] = arr2[j];

j++;

}

k++;

}

while (i < size) {

merged[k] = arr1[i];

i++;

k++;

}

while (j < size) {

merged[k] = arr2[j];

j++;

k++;

}

}

int main() {

int arr1[] = {10, 8, 6, 4, 2};

int arr2[] = {9, 7, 5, 3, 1};

int size = sizeof(arr1) / sizeof(arr1[0]);

int merged[size\*2];

mergeArrays(arr1, arr2, size, merged);

printf("Merged array in descending order:\n");

for (int i = 0; i < size\*2; i++) {

printf("%d ", merged[i]);

}

return 0;

}

10. Write a function in C to count the frequency of each element of an array.

#include <stdio.h>

void countFrequency(int arr[], int size)

{

int i, j, count;

int freq[size];

for(i = 0; i < size; i++)

{

freq[i] = -1;

}

for(i = 0; i < size; i++)

{

count = 1;

for(j = i+1; j < size; j++)

{

if(arr[i] == arr[j])

{

count++;

freq[j] = 0;

}

}

if(freq[i] != 0)

{

freq[i] = count;

}

}

printf("Element\t\tFrequency\n");

for(i = 0; i < size; i++)

{

if(freq[i] != 0)

{

printf("%d\t\t%d\n", arr[i], freq[i]);

}

}

}

int main()

{

int arr[] = {1, 2, 3, 2, 3, 1, 2, 2, 3, 3, 3};

int size = sizeof(arr)/sizeof(arr[0]);

countFrequency(arr, size);

return 0;

}