Assignment - 15 (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 greatestArrayElement(int [], int);

int greatestArrayElement(int arr[], int size)

{

int i, largest;

largest = arr[0];

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

if(arr[i] > largest)

largest = arr[i];

return largest;

}

int main()

{

int arr[10] = {400, 90, 80, 70, 60, 50, 40, 30, 20, 10};

printf("Greatest element in array is %d.", greatestArrayElement(arr, 10));

return 0;

}

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

#include<stdio.h>

int smallestArrayElement(int [], int);

int smallestArrayElement(int arr[], int size)

{

int i, smallest;

smallest = arr[0];

for(i = 1; i < 10; i++)

if(arr[i] < smallest)

smallest = arr[i];

return smallest;

}

int main()

{

int arr[10] = {400, 90, 80, 70, 60, 50, 40, 30, 20, 10};

printf("Smallest element in array is %d.", smallestArrayElement(arr, 10));

return 0;

}

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

#include<stdio.h>

void sortArray(int arr[], int size)

{

int i, count = -1;

while(count)

{

count = 0;

for(i = 0; i < size - 1; i++)

if(arr[i] > arr[i + 1])

{

arr[i] = arr[i] + arr[i + 1];

arr[i + 1] = arr[i] - arr[i + 1];

arr[i] = arr[i] - arr[i + 1];

count++;

}

}

}

int main()

{

int arr[10] = {49, 28, 17, 61, 58, 88, 62, 12, 36, 5}, i;

printf("Array before sorting is:\n");

for(i = 0; i < 10; i++)

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

sortArray(arr, 10);

printf("\nArray after sorting is now:\n");

for(i = 0; i < 10; i++)

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

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 A[], int size, int n, char dir[]);

void rotateArray(int A[], int size, int n, char dir[])

{

int i, valHold;

if(dir == "right")

{

if(n < size)

while(n)

{

valHold = A[size - 1];

for(i = size - 2; i >= 0; i--)

A[i + 1] = A[i];

A[0] = valHold;

n--;

}

else if(n > size)

printf("Rotation of Array by %d elements is not possible in array of size %d.", n, size);

}

else if(dir == "left")

{

if(n < size)

while(n)

{

valHold = A[0];

for(i = 0; i < size - 1; i++)

A[i] = A[i + 1];

A[size - 1] = valHold;

n--;

}

else if(n > size)

printf("Rotation of Array by %d elements is not possible in array of size %d.", n, size);

}

else

printf("Invalid Direction passed in function.");

}

void printArray(int arr[], int size)

{

int i;

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

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

}

int main()

{

//int A[10] = {7, 8, 6, 2, 4, 9, 7, 1, 8, 5};

int A[5] = {32, 29, 40, 12, 70};

printArray(A, 5);

printf("\n");

rotateArray(A, 5, 2, "left");

printArray(A, 5);

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 firstOccurenceOfAdjacentDuplicateValuesInArray(int arr[], int size)

{

int i;

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

{

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

return arr[i];

}

return -1;

}

int main()

{

int arr[10] = {10, 20, 30, 40, 50 , 60, 70, 70, 80, 90};

if(firstOccurenceOfAdjacentDuplicateValuesInArray(arr, 10) != -1)

printf("first occurence of adjacent duplicate values in array is %d.", firstOccurenceOfAdjacentDuplicateValuesInArray(arr, 10));

else

printf("No adjacent duplicate values are present in array.");

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 displayArrayValuesInReverseOrder()

{

int n, arr[n], i;

printf("Enter number of elements to be stored in array: ");

scanf("%d", &n);

printf("Enter %d values to be stored in array: ", n);

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

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

printf("Array values in reverse order are:\n");

for(i = n - 1; i >= 0; i--)

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

}

int main()

{

displayArrayValuesInReverseOrder();

return 0;

}

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

#include<stdio.h>

int totalNumberOfDuplicateElementsInArray(int arr[], int size)

{

int i, j, k = 1, uniqueElements[10] = {arr[0]};

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

{

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

{

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

{

break;

}

}

if(j >= i)

{

uniqueElements[k] = arr[i];

k++;

}

}

return size - k;

}

int main()

{

int arr[10] = {9, 0, 1, 5, 1, 7, 1, 0, 4, 6};

printf("Total Number of duplicate elements in array are %d.", totalNumberOfDuplicateElementsInArray(arr, 10));

return 0;

}

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

#include<stdio.h>

void printAllUniqueElementsInArray(int arr[], int size)

{

int i, j, k = 1, uniqueElements[10] = {arr[0]};

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

{

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

{

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

{

break;

}

}

if(j >= i)

{

uniqueElements[k] = arr[i];

k++;

}

}

printf("Unique Elements in Array are:\n");

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

{

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

}

printf("\b\b ");

}

int main()

{

int arr[10] = {9, 0, 1, 5, 1, 7, 1, 0, 4, 6};

printAllUniqueElementsInArray(arr, 10);

return 0;

}

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

#include<stdio.h>

void mergeTwoArraysOfSameSizeSortedInDescendingOrder(int arr1[], int arr2[], int mergedArray[], int size)

{

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

while(i < size && j < size)

{

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

{

mergedArray[k] = arr1[i];

i++;

k++;

}

else

{

mergedArray[k] = arr2[j];

j++;

k++;

}

}

while (i < size)

{

mergedArray[k] = arr1[i];

k++;

i++;

}

while (j < size)

{

mergedArray[k] = arr2[j];

k++;

j++;

}

}

int main()

{

int arr1[6] = {91, 68, 49, 37, 19, 7}, arr2[6] = {108, 98, 87, 45, 30, 11}, mergedArray[12], i;

mergeTwoArraysOfSameSizeSortedInDescendingOrder(arr1, arr2, mergedArray, 6);

for(i = 0; i < 12; i++)

{

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

}

return 0;

}

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

#include<stdio.h>

void printFrequencyOfEachElementOfArray(int arr[], int size)

{

int i, j, k = 1, uniqueElements[10] = {arr[0]}, elementFrequency;

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

{

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

{

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

{

break;

}

}

if(j >= i)

{

uniqueElements[k] = arr[i];

k++;

}

}

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

{

elementFrequency = 0;

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

{

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

{

elementFrequency++;

}

}

printf("Frequency of %d is %d.\n", uniqueElements[i], elementFrequency);

}

}

int main()

{

int arr[10] = {7, 0, 4, 2, 1, 7, 6, 2, 7, 2};

printFrequencyOfEachElementOfArray(arr, 10);

return 0;

}