Rajalakshmi Engineering College

Name: mahati sathish
Email: 240701302@rajalakshmi.edu.in
Roll no: 240701302
Phone: 9176485888
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_CY_Updated

Attempt : 1
Total Mark : 30
Marks Obtained : 30

Section 1 : Coding

1. Problem Statement

Priya, a data analyst, is working on a dataset of integers. She needs to find the maximum difference between two successive elements in the sorted version of the dataset. The dataset may contain a large number of integers, so Priya decides to use QuickSort to sort the array before finding the difference. Can you help Priya solve this efficiently?

Input Format

The first line of input consists of an integer n, representing the size of the array.

The second line consists of n space-separated integers, representing the elements of the array.

Output Format

The output prints a single integer, representing the maximum difference between two successive elements in the sorted form of the array.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
10
Output: Maximum gap: 0

Answer

```
// You are using GCC
#include <stdio.h>

void quicksort(int arr[], int low, int high)

{


   if (low >= high) return;
   int pivot = arr[high];
   int i = low - 1;
   for (int j = low; j < high; j++)

{


      if (arr[j] < pivot)

{


         i++;
         int t = arr[i];
         arr[i] = arr[j];
         arr[j] = t;


}
```

```
}
    int t = arr[i + 1];
    arr[i + 1] = arr[high];
    arr[high] = t;
    quicksort(arr, low, i);
    quicksort(arr, i + 2, high);

}

int main()

{



    int n;
    scanf("%d", &n);
    int arr[10];
    for (int i = 0; i < n; i++)

{



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



}
    quicksort(arr, 0, n - 1);
    int max_gap = 0;
    for (int i = 1; i < n; i++)

{



        int diff = arr[i] - arr[i - 1];
        if (diff > max_gap)

{



            max_gap = diff;
}

}
    printf("Maximum gap: %d\n", max_gap);
    return 0;

}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Marie, the teacher, wants her students to implement the ascending order of numbers while also exploring the concept of prime numbers.

Students need to write a program that sorts an array of integers using the merge sort algorithm while counting and returning the number of prime integers in the array. Help them to complete the program.

Input Format

The first line of input consists of an integer N, representing the number of array elements.

The second line consists of N space-separated integers, representing the array elements.

Output Format

The first line of output prints the sorted array of integers in ascending order.

The second line prints the number of prime integers in the array.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 7

5 3 6 8 9 7 4
Output: Sorted array: 3 4 5 6 7 8 9
Number of prime integers: 3

Answer

```
// You are using GCC
#include <stdio.h>
#include <math.h>

int isPrime(int num)

{


   if (num < 2) return 0;
   for (int i = 2; i <= sqrt(num); i++)

{


      if (num % i == 0) return 0;


}
   return 1;

}

void merge(int arr[], int l, int m, int r)

{


   int n1 = m - l + 1;
   int n2 = r - m;
   int L[10], R[10];
   for (int i = 0; i < n1; i++) L[i] = arr[l + i];
   for (int j = 0; j < n2; j++) R[j] = arr[m + 1 + j];
   int i = 0, j = 0, k = l;
   while (i < n1 && j < n2)

{
        if (L[i] <= R[j]) arr[k++] = L[i++];
        else arr[k++] = R[j++];


}
    while (i < n1) arr[k++] = L[i++];
    while (j < n2) arr[k++] = R[j++];

}

void mergeSort(int arr[], int l, int r)

{


    if (l < r)

{


        int m = l + (r - l) / 2;
        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);
        merge(arr, l, m, r);


}

}

int main()

{


    int n, count = 0;
    scanf("%d", &n);
    int arr[10];
    for (int i = 0; i < n; i++)
```

```
{
        scanf("%d", &arr[i]);
        if (isPrime(arr[i])) count++;
}
    mergeSort(arr, 0, n - 1);
    printf("Sorted array: ");
    for (int i = 0; i < n; i++)
{
        printf("%d ", arr[i]);
}
    printf("\nNumber of prime integers: %d\n", count);
    return 0;
}
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

Meera is organizing her art supplies, which are represented as a list of integers: red (0), white (1), and blue (2). She needs to sort these supplies so that all items of the same color are adjacent, in the order red, white, and blue. To achieve this efficiently, Meera decides to use QuickSort to sort the items. Can you help Meera arrange her supplies in the desired order?

Input Format

The first line of input consists of an integer n, representing the number of items in the list.

The second line consists of n space-separated integers, where each integer is either 0 (red), 1 (white), or 2 (blue).

Output Format

The output prints the sorted list of integers in a single line, where integers are arranged in the order red (0), white (1), and blue (2).

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 6
2 0 2 1 1 0
Output: Sorted colors:
0 0 1 1 2 2

Answer

```
// You are using GCC
#include <stdio.h>

void swap(int *a, int *b)

{


    int t = *a;
    *a = *b;
    *b = t;

}

int partition(int arr[], int low, int high)

{


    int pivot = arr[high];
    int i = low - 1;
    for (int j = low; j < high; j++)

{
    if (arr[j] < pivot)
{

        i++;
        swap(&arr[i], &arr[j]);

}

}
  swap(&arr[i + 1], &arr[high]);
  return i + 1;
}
void quickSort(int arr[], int low, int high)
{

  if (low < high)
{

    int pi = partition(arr, low, high);
    quickSort(arr, low, pi - 1);
    quickSort(arr, pi + 1, high);

}
}
int main()
{
    int n;
    scanf("%d", &n);
    int arr[100];
    for (int i = 0; i < n; i++)

{


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


}
    quickSort(arr, 0, n - 1);
    printf("Sorted colors:\n");
    for (int i = 0; i < n; i++)

{


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


}
    printf("\n");
    return 0;

}
```

Status : Correct ***Marks : 10/10***