Rajalakshmi Engineering College

Name: Sarvesh S
Email: 240701479@rajalakshmi.edu.in
Roll no: 240701479
Phone: 9361488694
Branch: REC
Department: I CSE FE
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

Reshma is passionate about sorting algorithms and has recently learned about the merge sort algorithm. She wants to implement a program that utilizes the merge sort algorithm to sort an array of integers, both positive and negative, in ascending order.

Help her in implementing the program.

Input Format

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

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

Output Format

The output prints N space-separated integers, representing the array elements sorted in ascending order.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 9
5 -3 0 12 7 -8 2 1 6

Output: -8 -3 0 1 2 5 6 7 12

Answer

```
#include<stdio.h>
#include<stdlib.h>

void merge(int arr[],int left,int center,int right){
   int a[25],b[25],n1,n2,aptr,bptr,cptr,i,j;
   n1=center-left+1;
   n2=right-center;
   for(int i=0;i<n1;i++){
      a[i]=arr[left+i];
   }
   for(int j=0;j<n2;j++){
      b[j]=arr[center+1+j];
   }
   aptr=0;
   bptr=0;
   cptr=left;
   while(aptr<n1 && bptr<n2){
      if(a[aptr]<=b[bptr]){
         arr[cptr]=a[aptr];
         aptr++;
      }
      else{
         arr[cptr]=b[bptr];
         bptr++;
      }
      cptr++;
```

```
    }
    while(aptr<n1){
        arr[cptr]=a[aptr];
        aptr++;
        cptr++;
    }
    while(bptr<n2){
        arr[cptr]=b[bptr];
        bptr++;
        cptr++;
    }
}

void mergesort(int arr[],int left,int right){
    int center;
    if(left<right){
        center=(left+right)/2;
        mergesort(arr,left,center);
        mergesort(arr,center+1,right);
        merge(arr,left,center,right);
    }
}

int main(){
    int n,e;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++){
        scanf("%d",&e);
        arr[i]=e;
    }
    int left=0,right=n-1;
    mergesort(arr,left,right);
    for(int i=0;i<n;i++){
        printf("%d ",arr[i]);
    }
    return 0;
}
```

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

2. 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

```
#include<stdio.h>
#include<stdlib.h>

int comp(const void *a,const void *b){
   return (*(int*)a-*(int*)b);
}

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

```
        scanf("%d",&e);
        arr[i]=e;
    }
    qsort(arr,n,sizeof(arr[0]),comp);
    int max_gap=0;
    for(int i=1;i<n;i++){
        int gap=arr[i]-arr[i-1];
        if(gap>max_gap){
            max_gap=gap;
        }
    }
    printf("Maximum gap: %d",max_gap);
    return 0;
}
```

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

3. Problem Statement

Aryan is participating in a coding competition where he needs to sort a list of numbers using an efficient sorting algorithm. He decides to use Merge Sort, a divide-and-conquer algorithm, to achieve this. Given a list of n elements, Aryan must implement merge sort to arrange the numbers in ascending order.

Help Aryan by implementing the merge sort algorithm to correctly sort the given list of numbers.

Input Format

The first line of input contains an integer n, the number of elements in the list.

The second line contains n space-separated integers representing the elements of the list.

Output Format

The output prints the sorted list of numbers in ascending order, separated by a space.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
80 40 20 50 30

Output: 20 30 40 50 80

Answer

```
#include<stdio.h>
#include<stdlib.h>

void merge(int arr[],int left,int center,int right){
   int a[25],b[25],n1,n2,aptr,bptr,cptr,i,j;
   n1=center-left+1;
   n2=right-center;
   for(i=0;i<n1;i++){
      a[i]=arr[left+i];
   }
   for(j=0;j<n2;j++){
      b[j]=arr[center+1+j];
   }
   aptr=0;
   bptr=0;
   cptr=left;
   while(aptr<n1 && bptr<n2){
      if(a[aptr]<=b[bptr]){
         arr[cptr]=a[aptr];
         aptr++;
      }
      else{
         arr[cptr]=b[bptr];
         bptr++;
      }
      cptr++;
   }
   while(aptr<n1){
      arr[cptr]=a[aptr];
      aptr++;
      cptr++;
   }
   while(bptr<n2){
```

```
        arr[cptr]=b[bptr];
        bptr++;
        cptr++;
    }
}

void mergesort(int arr[],int left,int right){
    int center;
    if(left<right){
        center=(left+right)/2;
        mergesort(arr,left,center);
        mergesort(arr,center+1,right);
        merge(arr,left,center,right);
    }
}

int main(){
    int n,e,left=0;
    scanf("%d",&n);
    int arr[n],right=n-1;
    for(int i=0;i<n;i++){
        scanf("%d",&e);
        arr[i]=e;
    }
    mergesort(arr,left,right);
    for(int i=0;i<n;i++){
        printf("%d ",arr[i]);
    }
    return 0;
}
```

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