

Rajalakshmi Engineering College

Name: Nikhil Vinayak P
Email: 240701359@rajalakshmi.edu.in
Roll no: 240701359
Phone: 9884558531
Branch: REC
Department: I CSE FD
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>
#include <stdlib.h>
int compare(const void *a,const void *b) {
    return (*(int*)a-*(int*)b);
}
int max_difference(int arr[],int n) {
    qsort(arr,n,sizeof(int),compare);
    int max_diff=0;
    for (int i=0; i<n-1; i++) {
        int diff=arr[i+1]-arr[i];
        if (diff>max_diff) {
            max_diff=diff;
        }
    }
    return max_diff;
}
int main() {
    int n;
    scanf("%d", &n);
    int arr[n];
    for (int i=0; i<n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Maximum gap: %d\n", max_difference(arr,n));
    return 0;
}
```

Status : Correct

Marks : 10/10

2. 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

```
// You are using GCC
#include <stdio.h>
void merge(int arr[],int left,int mid,int right) {
    int i,j,k;
    int n1=mid-left+1;
    int n2=right-mid;
    int L[n1],R[n2];
    for (i=0; i<n1; i++)
```

```

    L[i]=arr[left+i];
    for (j=0; j<n2; j++)
        R[j]=arr[mid+1+j];
    i=0;j=0;k=left;
    while (i<n1 && j<n2) {
        if (L[i]<=R[j]) {
            arr[k]=L[i];
            i++;
        }
        else {
            arr[k]=R[j];
            j++;
        }
        k++;
    }
    while (i<n1) {
        arr[k]=L[i];
        i++;
        k++;
    }
    while (j<n2) {
        arr[k]=R[j];
        j++;
        k++;
    }
}

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

int main() {
    int n;
    scanf("%d", &n);
    int arr[n];
    for (int i=0; i<n; i++)
        scanf("%d", &arr[i]);
    mergeSort(arr,0,n-1);
    for (int i=0; i<n; i++)

```

```
    printf("%d ", arr[i]);  
    return 0;  
}
```

Status : Correct

Marks : 10/10

3. Problem Statement

Sheela wants to distribute cookies to her children, but each child will only be happy if the cookie size meets or exceeds their individual greed factor. She has a limited number of cookies and wants to make as many children happy as possible. Priya decides to sort both the greed factors and cookie sizes using QuickSort to efficiently match cookies with children. Your task is to help Sheela determine the maximum number of children that can be made happy.

Input Format

The first line of input consists of an integer n , representing the number of children.

The second line contains n space-separated integers, where each integer represents the greed factor of a child.

The third line contains an integer m , representing the number of cookies.

The fourth line contains m space-separated integers, where each integer represents the size of a cookie.

Output Format

The output prints a single integer, representing the maximum number of children that can be made happy.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
1 2 3

2

11

Output: The child with greed factor: 1

Answer

```
#include <stdio.h>
#include <stdlib.h>
int compare(const void *a, const void *b) {
    return (*(int*)a - *(int*)b);
}
int main() {
    int n, m;
    scanf("%d", &n);
    int greed[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &greed[i]);
    }
    scanf("%d", &m);
    int cookies[m];
    for (int i = 0; i < m; i++) {
        scanf("%d", &cookies[i]);
    }
    qsort(greed, n, sizeof(int), compare);
    qsort(cookies, m, sizeof(int), compare);
    int child = 0, cookie = 0, happyChildren = 0;
    while (child < n && cookie < m) {
        if (cookies[cookie] >= greed[child]) {
            happyChildren++;
            child++;
        }
        cookie++;
    }
    printf("The child with greed factor: %d\n", happyChildren);
    return 0;
}
```

Status : Correct

Marks : 10/10