Rajalakshmi Engineering College

Name: Baviri Setty Sai Deevan
Email: 241901016@rajalakshmi.edu.in
Roll no: 241901016
Phone: 8838291380
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 5_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 13

Section 1 : MCQ

1. Find the preorder traversal of the given binary search tree.

Answer

9, 2, 1, 6, 4, 7, 10, 14

Status : Correct ***Marks : 1/1***

2. Find the post-order traversal of the given binary search tree.

Answer

10, 17, 20, 18, 15, 32, 21

Status : Correct ***Marks : 1/1***

3. While inserting the elements 5, 4, 2, 8, 7, 10, 12 in a binary search tree, the element at the lowest level is ________.

Answer

12

Status : Correct *Marks : 1/1*

4. Find the pre-order traversal of the given binary search tree.

Answer

13, 2, 1, 4, 14, 18

Status : Correct *Marks : 1/1*

5. Find the in-order traversal of the given binary search tree.

Answer

1, 2, 4, 13, 14, 18

Status : Correct *Marks : 1/1*

6. Which of the following is a valid preorder traversal of the binary search tree with nodes: 18, 28, 12, 11, 16, 14, 17?

Answer

18, 12, 11, 16, 14, 17, 28

Status : Correct *Marks : 1/1*

7. While inserting the elements 71, 65, 84, 69, 67, 83 in an empty binary search tree (BST) in the sequence shown, the element in the lowest level is _______.

Answer

67

Status : Correct ***Marks : 1/1***

8. Find the postorder traversal of the given binary search tree.

Answer

1, 4, 2, 18, 14, 13

Status : Correct ***Marks : 1/1***

9. Which of the following is the correct post-order traversal of a binary search tree with nodes: 50, 30, 20, 55, 32, 52, 57?

Answer

20, 32, 30, 52, 57, 55, 50

Status : Correct ***Marks : 1/1***

10. Which of the following operations can be used to traverse a Binary Search Tree (BST) in ascending order?

Answer

Inorder traversal

Status : Correct ***Marks : 1/1***

11. The preorder traversal of a binary search tree is 15, 10, 12, 11, 20, 18, 16, 19. Which one of the following is the postorder traversal of the tree?

Answer

11, 12, 10, 16, 19, 18, 20, 15

Status : Correct ***Marks : 1/1***

12. Which of the following is the correct in-order traversal of a binary search tree with nodes: 9, 3, 5, 11, 8, 4, 2?

Answer

3, 4 8, 5, 2, 9, 11

Status : Wrong ***Marks : 0/1***

13. How many distinct binary search trees can be created out of 4 distinct keys?

Answer

5

Status : Wrong ***Marks : 0/1***

14. Which of the following is the correct pre-order traversal of a binary search tree with nodes: 50, 30, 20, 55, 32, 52, 57?

Answer

50, 30, 20, 32, 55, 52, 57

Status : Correct ***Marks : 1/1***

15. In a binary search tree with nodes 18, 28, 12, 11, 16, 14, 17, what is the value of the left child of the node 16?

Answer

14

Status : Correct ***Marks : 1/1***

Rajalakshmi Engineering College

Name: Baviri Setty Sai Deevan
Email: 241901016@rajalakshmi.edu.in
Roll no: 241901016
Phone: 8838291380
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are the lead developer of a text-processing application that assists writers in organizing their thoughts. One crucial feature is a character-sorting service that helps users highlight the most critical elements of their text.

To achieve this, you decide to enhance the service to sort characters in descending order using the Quick-Sort algorithm. Implement the algorithm to efficiently rearrange the characters, ensuring that it is sorted in descending order.

Input Format

The first line of the input consists of a positive integer value N, representing the number of characters to be sorted.

The second line of input consists of N space-separated lowercase alphabetical characters.

Output Format

The output displays the set of alphabetical characters, sorted in descending order.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5
a d g j k

Output: k j g d a

Answer

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

// You are using GCC
void swap(char* a, char* b) {
    char temp = *a;
    *a = *b;
    *b = temp;
}

int partition(char arr[], int low, int high) {
    int pivot = arr[low];
    int start = low;
    int end = high;
    while (start < end){
        while (arr[start] >= pivot) start++;
        while (arr[end] < pivot) end--;
        if (start < end) swap(&arr[start], &arr[end]);
    }
    swap(&arr[low], &arr[end]);
    return end;
}
```

```
void quicksort(char arr[], int low, int high) {
   if (low < high){
      int loc = partition(arr, low, high);
      quicksort(arr, low, loc - 1);
      quicksort(arr, loc + 1, high);
   }
}

int main() {
   int n;
   scanf("%d", &n);

   char characters[n];

   for (int i = 0; i < n; i++) {
      char input;
      scanf(" %c", &input);
      characters[i] = input;
   }

   quicksort(characters, 0, n - 1);

   for (int i = 0; i < n; i++) {
      printf("%c ", characters[i]);
   }

   return 0;
}
```

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

Rajalakshmi Engineering College

Name: Baviri Setty Sai Deevan
Email: 241901016@rajalakshmi.edu.in
Roll no: 241901016
Phone: 8838291380
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_COD_Question 2

Attempt : 2
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Nandhini asked her students to arrange a set of numbers in ascending order. She asked the students to arrange the elements using insertion sort, which involves taking each element and placing it in its appropriate position within the sorted portion of the array.

Assist them in the task.

Input Format

The first line of input consists of the value of n, representing the number of array elements.

The second line consists of n elements, separated by a space.

Output Format

The output prints the sorted array, separated by a space.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
67 28 92 37 59

Output: 28 37 59 67 92

Answer

```
#include <stdio.h>

// You are using GCC

void insertionSort(int a[], int n) {
    for (int i = 1 ; i < n ; i++){
        int temp = a[i];
        int j = i - 1;
        while (j >= 0 && a[j] > temp){
            a[j + 1] = a[j];
            j--;
        }
        a[j + 1] = temp;
    }
}

void printArray(int arr[], int n) {
    for (int i = 0 ; i < n ; i++){
        printf("%d ",arr[i]);
    }
    printf("\n");
}

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

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

Rajalakshmi Engineering College

Name: Baviri Setty Sai Deevan
Email: 241901016@rajalakshmi.edu.in
Roll no: 241901016
Phone: 8838291380
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_COD_Question 1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

John and Mary are collaborating on a project that involves data analysis. They each have a set of age data, one sorted in ascending order and the other in descending order. However, their analysis requires the data to be in ascending order.

Write a program to help them merge the two sets of age data into a single sorted array in ascending order using merge sort.

Input Format

The first line of input consists of an integer N, representing the number of age values in each dataset.

The second line consists of N space-separated integers, representing the ages of participants in John's dataset (in ascending order).

The third line consists of N space-separated integers, representing the ages of participants in Mary's dataset (in descending order).

Output Format

The output prints a single line containing space-separated integers, which represents the merged dataset of ages sorted in ascending order.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
1 3 5 7 9
10 8 6 4 2

Output: 1 2 3 4 5 6 7 8 9 10

Answer

```
#include <stdio.h>


void merge(int arr[], int left[], int right[], int left_size, int right_size) {
    int i = 0, j = 0, k = 0;

    while (i < left_size && j < right_size) {
        if (left[i] <= right[j]) {
            arr[k] = left[i];
            i++;
        } else {
            arr[k] = right[j];
            j++;
        }
        k++;
    }

    while (i < left_size) {
        arr[k] = left[i];
        i++;
        k++;
    }
    while (j < right_size) {
        arr[k] = right[j];
        j++;
        k++;
    }
}

void mergeSort(int arr[], int size) {
    if (size > 1) {
        int mid = size / 2;
        int left[mid];
        int right[size - mid];

        for (int i = 0; i < mid; i++) {
            left[i] = arr[i];
        }

        for (int i = mid; i < size; i++) {
            right[i - mid] = arr[i];
        }

        mergeSort(left, mid);
        mergeSort(right, size - mid);
        merge(arr, left, right, mid, size - mid);
    }
}

int main() {
    int n, m;
    scanf("%d", &n);
    int arr1[n], arr2[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr1[i]);
    }
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr2[i]);
    }
    int merged[n + n];
    mergeSort(arr1, n);
    mergeSort(arr2, n);
    merge(merged, arr1, arr2, n, n);
    for (int i = 0; i < n + n; i++) {
        printf("%d ", merged[i]);
    }
    return 0;
}
```

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

Rajalakshmi Engineering College

Name: Baviri Setty Sai Deevan
Email: 241901016@rajalakshmi.edu.in
Roll no: 241901016
Phone: 8838291380
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_COD_Question 5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Jose has an array of N fractional values, represented as double-point numbers. He needs to sort these fractions in increasing order and seeks your help.

Write a program to help Jose sort the array using the merge sort algorithm.

Input Format

The first line of input consists of an integer N, representing the number of fractions to be sorted.

The second line consists of N double-point numbers, separated by spaces, representing the fractions array.

Output Format

The output prints N double-point numbers, sorted in increasing order, and rounded to three decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4
0.123 0.543 0.321 0.789

Output: 0.123 0.321 0.543 0.789

Answer

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

// You are using GCC
int compare(double a, double b) {
    if (a < b) return -1;
    else if (a > b) return 1;
    else return 0;
}
void merge(double arr[], int l, int m, int r) {
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;
    double L[n1], R[n2];
    for (i = 0 ; i < n1 ; i++) L[i] = arr[l + i];
    for (j = 0 ; j < n2 ; j++) R[j] = arr[m + 1 + j];
    i = 0, j = 0, k = l;
    while (i < n1 && j < n2){
        if (compare(L[i], R[j]) <= 0){
            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(double 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;
    scanf("%d", &n);
    double fractions[n];
    for (int i = 0; i < n; i++) {
        scanf("%lf", &fractions[i]);
    }
    mergeSort(fractions, 0, n - 1);
    for (int i = 0; i < n; i++) {
        printf("%.3f ", fractions[i]);
    }
    return 0;
}
```

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

Rajalakshmi Engineering College

Name: Baviri Setty Sai Deevan
Email: 241901016@rajalakshmi.edu.in
Roll no: 241901016
Phone: 8838291380
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Kavya, a software developer, is analyzing data trends. She has a list of integers and wants to identify the nth largest number in the list after sorting the array using QuickSort.

To optimize performance, Kavya is required to use QuickSort to sort the list before finding the nth largest number.

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 nums.

The third line consists of an integer k, representing the position of the largest number you need to print after sorting the array.

Output Format

The output prints the k-th largest number in the sorted array (sorted in ascending order).

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 6
-1 0 1 2 -1 -4
3
Output: 0

Answer

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

void swap(int *a, int*b){
   int temp = *a;
   *a = *b;
   *b = temp;
}

int partition(int arr[], int low, int high) {
   int pivot = arr[low];
   int start = low, end = high;
   while (start < end){
      while (arr[start] <= pivot) start++;
      while (arr[end] > pivot) end--;
      if (start < end) swap(&arr[start], &arr[end]);
   }
   swap(&arr[low], &arr[end]);
   return end;
}

void quickSort(int arr[], int low, int high) {
   if (low < high){
      int loc = partition(arr, low, high);
```

```
        quickSort(arr, low, loc - 1);
        quickSort(arr, loc + 1, high);
    }
}

void findNthLargest(int* nums, int n, int k) {
    quickSort(nums, 0, n - 1);
    printf("%d",nums[n - k]);
}

int main() {
    int n, k;
    scanf("%d", &n);
    int* nums = (int*)malloc(n * sizeof(int));
    for (int i = 0; i < n; i++) {
        scanf("%d", &nums[i]);
    }
    scanf("%d", &k);
    findNthLargest(nums, n, k);
    free(nums);
    return 0;
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Baviri Setty Sai Deevan
Email: 241901016@rajalakshmi.edu.in
Roll no: 241901016
Phone: 8838291380
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 5_CY_Updated

Attempt : 1
Total Mark : 30
Marks Obtained : 23.5

Section 1 : Coding

1. Problem Statement

Arun is working on a Binary Search Tree (BST) data structure. His goal is to implement a program that reads a series of integers and inserts them into a BST. Once the integers are inserted, he needs to add a given integer value to each node in the tree and find the maximum value in the BST.

Your task is to help Arun implement this program.

Input Format

The first line of input consists of an integer N, representing the number of elements to be inserted into the BST.

The second line consists of N space-separated integers, each representing an element to be inserted into the BST.

The third line consists of an integer add, representing the value to be added to each node in the BST.

Output Format

The output prints the maximum value in the BST after adding the add value.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
10 5 15 20 25
5
Output: 30

Answer

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

struct node{
   int data;
   struct node* left;
   struct node* right;
};

struct node* createnode(int data){
   struct node* newnode = (struct node*)malloc(sizeof(struct node));
   newnode -> data = data;
   newnode -> left = newnode -> right = NULL;
   return newnode;
}

struct node* insert(struct node* root, int data){
   if (root == NULL) return createnode(data);
   if (root -> data > data) root -> left = insert(root -> left, data);
   else root -> right = insert(root -> right, data);
   return root;
}

int findmax(struct node* root){
   while (root && root -> right != NULL) root = root -> right;
   return root -> data;
}

int main(){
   struct node* root = NULL;
   int n;
   scanf("%d",&n);
   int js;
   for (int i = 0 ; i < n ; i++){
       scanf("%d ",&js);
       root = insert(root, js);
   }
   int max = findmax(root);
   int addi;
   scanf("%d",&addi);
   printf("%d",addi + max);
   return 0;
}
```

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

2. Problem Statement

John is building a system to store and manage integers using a binary search tree (BST). He needs to add a feature that allows users to search for a specific integer key in the BST using recursion.

Implement functions to create the BST and perform a recursive search for an integer.

Input Format

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

The second line consists of integers representing, the values of nodes, separated by space.

The third line consists of an integer representing, the key to be searched.

Output Format

The output prints whether the given key is present in the binary search tree or not.

Refer to the sample output for the exact format.

Sample Test Case

Input: 7
10 5 15 3 7 12 20
12
Output: The key 12 is found in the binary search tree

Answer

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

struct node{
   int data;
   struct node* left;
   struct node* right;
};

struct node* createnode(int data){
   struct node* newnode = (struct node*)malloc(sizeof(struct node));
   newnode -> data = data;
   newnode -> left = newnode -> right = NULL;
   return newnode;
}

struct node* insert(struct node* root, int data){
   if (root == NULL) return createnode(data);
   if (root -> data > data) root -> left = insert(root -> left, data);
   else root -> right = insert(root -> right, data);
   return root;
}

struct node* search(struct node* root, int data){
   if (root == NULL) return NULL;
   else if (data < root -> data) return search(root -> left, data);
   else if (data > root -> data) return search(root -> right, data);
```

```
    else return root;
}

int main(){
    struct node* root = NULL;
    int n;
    scanf("%d",&n);
    int ruby;
    for (int i = 0 ; i < n ; i++){
        scanf("%d ",&ruby);
        root = insert(root, ruby);
    }
    int searchele;
    scanf("%d",&searchele);
    if (search(root, searchele)) printf("The key %d is found in the binary search tree
\n",searchele);
    else printf("The key %d is not found in the binary search tree\n",searchele);
    return 0;
}
```

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

3. Problem Statement

Dhruv is working on a project where he needs to implement a Binary Search Tree (BST) data structure and perform various operations on it.

He wants to create a program that allows him to build a BST, traverse it in different orders (inorder, preorder, postorder), and exit the program when needed.

Help Dhruv by designing a program that fulfils his requirements.

Input Format

The first input consists of the choice.

If the choice is 1, enter the number of elements N and the elements inserted into the tree, separated by a space in a new line.

If the choice is 2, print the in-order traversal.

If the choice is 3, print the pre-order traversal.

If the choice is 4, print the post-order traversal.

If the choice is 5, exit.

Output Format

The output prints the results based on the choice.

For choice 1, print "BST with N nodes is ready to use" where N is the number of nodes inserted.

For choice 2, print the in-order traversal of the BST.

For choice 3, print the pre-order traversal of the BST.

For choice 4, print the post-order traversal of the BST.

For choice 5, the program exits.

If the choice is greater than 5, print "Wrong choice".

Refer to the sample output for the formatting specifications.

Sample Test Case

```
Input: 1
5
12 78 96 34 55
2
3
4
5
Output: BST with 5 nodes is ready to use
BST Traversal in INORDER
```

```
12 34 55 78 96
BST Traversal in PREORDER
12 78 34 55 96
BST Traversal in POSTORDER
55 34 96 78 12
```

Answer

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

struct node{
   int data;
   struct node* left;
   struct node* right;
};

struct node* createnode(int data){
   struct node* newnode = (struct node*)malloc(sizeof(struct node));
   newnode -> data = data;
   newnode -> left = newnode -> right = NULL;
   return newnode;
}

struct node* insert(struct node* root, int data){
   if (root == NULL) return createnode(data);
   if (data > root -> data) root -> right = insert(root -> right, data);
   else root -> left = insert(root -> left, data);
   return root;
}

void inorder(struct node* root){
   if (root != NULL){
      inorder(root -> left);
      printf("%d ",root -> data);
      inorder(root -> right);
   }
}

void preorder(struct node* root){
   if (root != NULL){
      printf("%d ",root -> data);
      preorder(root -> left);
```

```
        preorder(root -> right);
    }
}

void postorder(struct node* root){
    if (root != NULL){
        postorder(root -> left);
        postorder(root -> right);
        printf("%d ",root -> data);
    }
}

void freetree(struct node* root){
    if (root == NULL) return;
    freetree(root -> left);
    freetree(root -> right);
    free(root);
}

int main(){
    struct node* root = NULL;
    int choice;
    int flutter;
    int n;
    do{
        scanf("%d",&choice);
        switch(choice){
            case 1:
                freetree(root);
                scanf("%d",&n);
                for (int i = 0 ; i < n ; i++){
                    scanf("%d ",&flutter);
                    root = insert(root, flutter);
                }
                printf("BST with %d nodes is ready to use\n",n);
                break;
            case 2:
                printf("BST Traversal in INORDER\n");
                inorder(root);
                printf("\n");
                break;
            case 3:
                printf("BST Traversal in PREORDER\n");
                preorder(root);
                printf("\n");
                break;
            case 4:
                printf("BST Traversal in POSTORDER\n");
                postorder(root);
                printf("\n");
                break;
            case 5:
                exit(0);
                break;
            default:
                printf("Wrong choice\n");
                break;
        }
    }while(choice != 5);
    return 0;
}
```

Status : Partially correct ***Marks : 3.5/10***

Rajalakshmi Engineering College

Name: Baviri Setty Sai Deevan
Email: 241901016@rajalakshmi.edu.in
Roll no: 241901016
Phone: 8838291380
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 5_PAH_Updated

Attempt : 1
Total Mark : 50
Marks Obtained : 30

Section 1 : Coding

1. Problem Statement

Yogi is working on a program to manage a binary search tree (BST) containing integer values. He wants to implement a function that removes nodes from the tree that fall outside a specified range defined by a minimum and maximum value.

Help Yogi by writing a function that achieves this.

Input Format

The first line of input consists of an integer N, representing the number of elements to be inserted into the BST.

The second line consists of N space-separated integers, representing the elements to be inserted into the BST.

The third line consists of two space-separated integers min and max, representing the minimum value and the maximum value of the range.

Output Format

The output prints the remaining elements of the BST in an in-order traversal, after removing nodes that fall outside the specified range.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
10 5 15 20 12
5 15
Output: 5 10 12 15

Answer

-

Status : Skipped ***Marks : 0/10***

2. Problem Statement

Aishu is participating in a coding challenge where she needs to reconstruct a Binary Search Tree (BST) from given preorder traversal data and then print the in-order traversal of the reconstructed BST.

Since Aishu is just learning about tree data structures, she needs your help to write a program that does this efficiently.

Input Format

The first line consists of an integer n, representing the number of nodes in the BST.

The second line of input contains n integers separated by spaces, which represent the preorder traversal of the BST.

Output Format

The output displays n space-separated integers, representing the in-order traversal of the reconstructed BST.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 6
10 5 1 7 40 50

Output: 1 5 7 10 40 50

Answer

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

struct node{
   int data;
   struct node* left;
   struct node* right;
};

struct node* createnode(int data){
   struct node* newnode = (struct node*)malloc(sizeof(struct node));
   newnode -> data = data;
   newnode -> left = newnode -> right = NULL;
   return newnode;
}

struct node* insert(struct node* root, int data){
   struct node* newnode = createnode(data);
   if (root == NULL){
      root = newnode;
   }else if (root -> data > data){
      root -> left = insert(root -> left, data);
   }else if (root -> data < data){
      root -> right = insert(root -> right, data);
   }
   return root;
}
```

```
void inorder(struct node* root){
   if (root != NULL){
      inorder(root -> left);
      printf("%d ",root -> data);
      inorder(root -> right);
   }
}

int main(){
   struct node* root = NULL;
   int n;
   scanf("%d",&n);
   int react;
   for (int i = 0 ; i < n ; i++){
      scanf("%d ",&react);
      root = insert(root, react);
   }
   inorder(root);
   printf("\n");
   return 0;
}
```

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

3. Problem Statement

Viha, a software developer, is working on a project to automate searching for a target value in a Binary Search Tree (BST). She needs to create a program that takes an integer target value as input and determines if that value is present in the BST or not.

Write a program to assist Viha.

Input Format

The first line of input consists of integers separated by spaces, which represent the elements to be inserted into the BST. The input is terminated by entering -1.

The second line consists of an integer target, which represents the target value to be searched in the BST.

Output Format

If the target value is found in the BST, print "[target] is found in the BST".

Else, print "[target] is not found in the BST"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5 3 7 1 4 6 8 -1
4
Output: 4 is found in the BST

Answer

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

struct node{
   int data;
   struct node* left;
   struct node* right;
};

struct node* createnode(int data){
   struct node* newnode = (struct node*)malloc(sizeof(struct node));
   newnode -> data = data;
   newnode -> left = newnode -> right = NULL;
   return newnode;
}

struct node* insert(struct node* root, int data){
   struct node* newnode = createnode(data);
   if (root == NULL){
      root = newnode;
   }else if (root -> data > data){
      root -> left = insert(root -> left, data);
   }else if (root -> data < data){
      root -> right = insert(root -> right, data);
   }
```

```
    return root;
}

struct node* search(struct node* root, int key){
    if (root == NULL){
        return NULL;
    }else if (root -> data < key){
        return search(root -> right, key);
    }else if (root -> data > key){
        return search(root -> left, key);
    }else{
        return root;
    }
}

int main(){
    struct node* root = NULL;
    int prolog;
    while (1){
        scanf("%d ",&prolog);
        if (prolog == -1)break;
        else root = insert(root, prolog);
    }
    int key;
    scanf("%d",&key);
    if (search(root, key)){
        printf("%d is found in the BST\n",key);
    }else{
        printf("%d is not found in the BST\n",key);
    }
    return 0;
}
```

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

4. Problem Statement

Arun is exploring operations on binary search trees (BST). He wants to write a program with an unsorted distinct integer array that represents the BST keys and construct a height-balanced BST from it.

After constructing, he wants to perform the following operations that can alter the structure of the tree and traverse them using a level-order traversal:

InsertionDeletion

Your task is to assist Arun in completing the program without any errors.

Input Format

The first line of input consists of an integer N, representing the number of initial keys in the BST.

The second line consists of N space-separated integers, representing the initial keys.

The third line consists of an integer X, representing the new key to be inserted into the BST.

The fourth line consists of an integer Y, representing the key to be deleted from the BST.

Output Format

The first line of output prints "Initial BST: " followed by a space-separated list of keys in the initial BST after constructing it in level order traversal.

The second line prints "BST after inserting a new node X: " followed by a space-separated list of keys in the BST after inserting X n level order traversal.

The third line prints "BST after deleting node Y: " followed by a space-separated list of keys in the BST after deleting Y n level order traversal.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
25 14 56 28 12
34
12

Output: Initial BST: 25 14 56 12 28
BST after inserting a new node 34: 25 14 56 12 28 34
BST after deleting node 12: 25 14 56 28 34

Answer

-

Status : Skipped ***Marks : 0/10***

5. Problem Statement

Joseph, a computer science student, is interested in understanding binary search trees (BST) and their node arrangements. He wants to create a program to explore BSTs by inserting elements into a tree and displaying the nodes using post-order traversal of the tree.

Write a program to help Joseph implement the program.

Input Format

The first line of input consists of an integer N, representing the number of elements to insert into the BST.

The second line consists of N space-separated integers data, which is the data to be inserted into the BST.

Output Format

The output prints N space-separated integer values after the post-order traversal.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4
10 15 5 3

Output: 3 5 15 10

Answer

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

struct node{
    int data;
    struct node* left;
    struct node* right;
};

struct node* createnode(int data){
    struct node* newnode = (struct node*)malloc(sizeof(struct node));
    newnode -> data = data;
    newnode -> left = newnode -> right = NULL;
    return newnode;
}

struct node* insert(struct node* root, int data){
    struct node* newnode = createnode(data);
    if (root == NULL){
        root = newnode;
    }else if(root -> data > data){
        root -> left = insert(root -> left, data);
    }else if (root -> data < data){
        root -> right = insert(root -> right, data);
    }
    return root;
}

void postorder(struct node* root){
    if (root != NULL){
        postorder(root -> left);
        postorder(root -> right);
        printf("%d ",root -> data);
    }
}

int main(){
    struct node* root = NULL;
    int n;
    scanf("%d",&n);
    int r;
    for (int i = 0 ; i < n ; i++){
        scanf("%d ",&r);
        root = insert(root, r);
    }
    postorder(root);
    return 0;
}
```

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