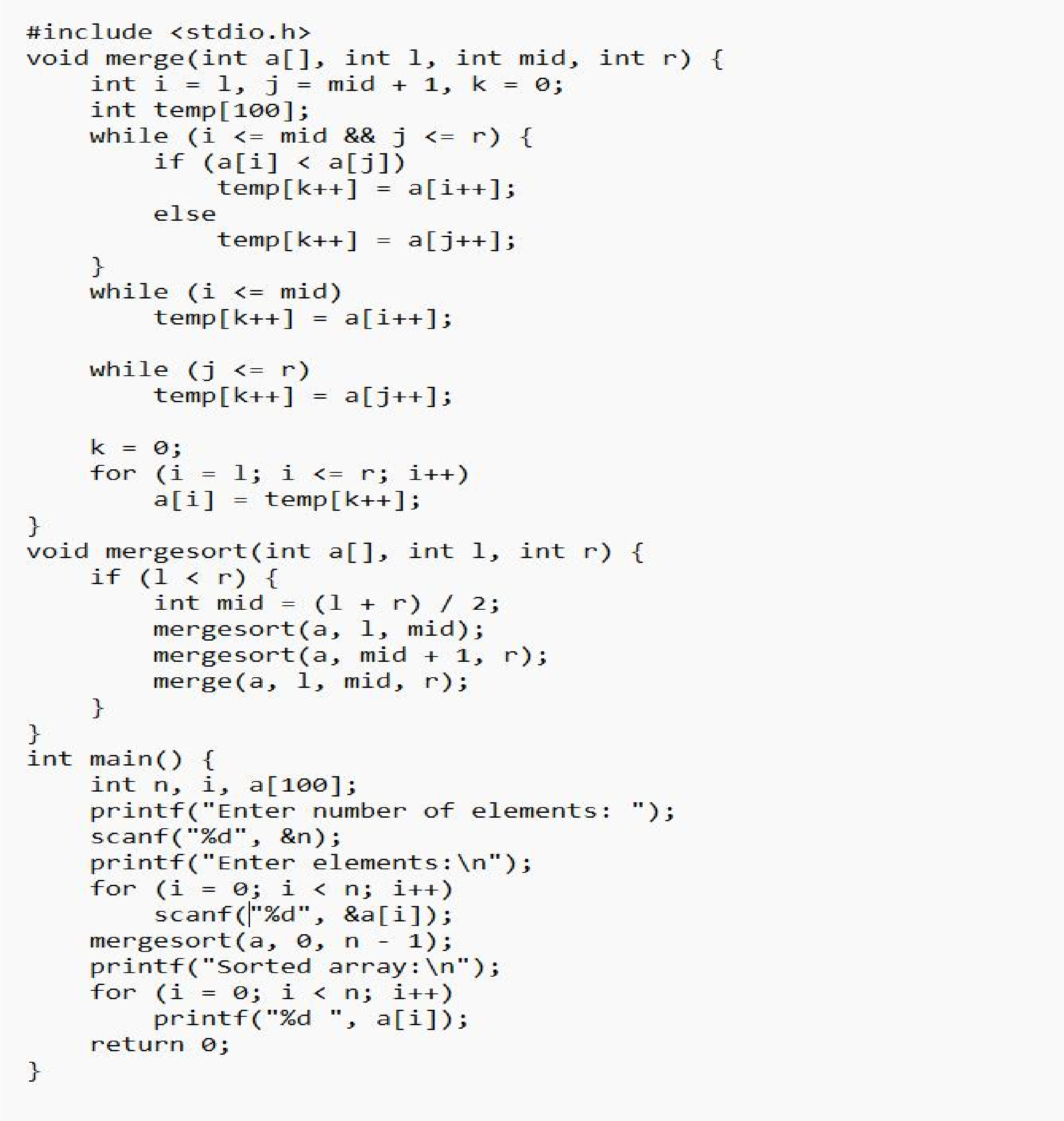
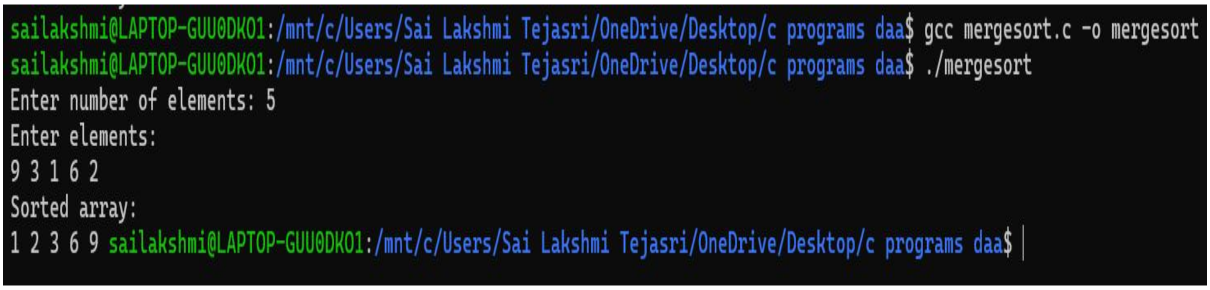
NAME :SAI LAKSMI TEJASRI K

ROLL NO:CH.SC.U4CSE24243

1)MERGE SORT

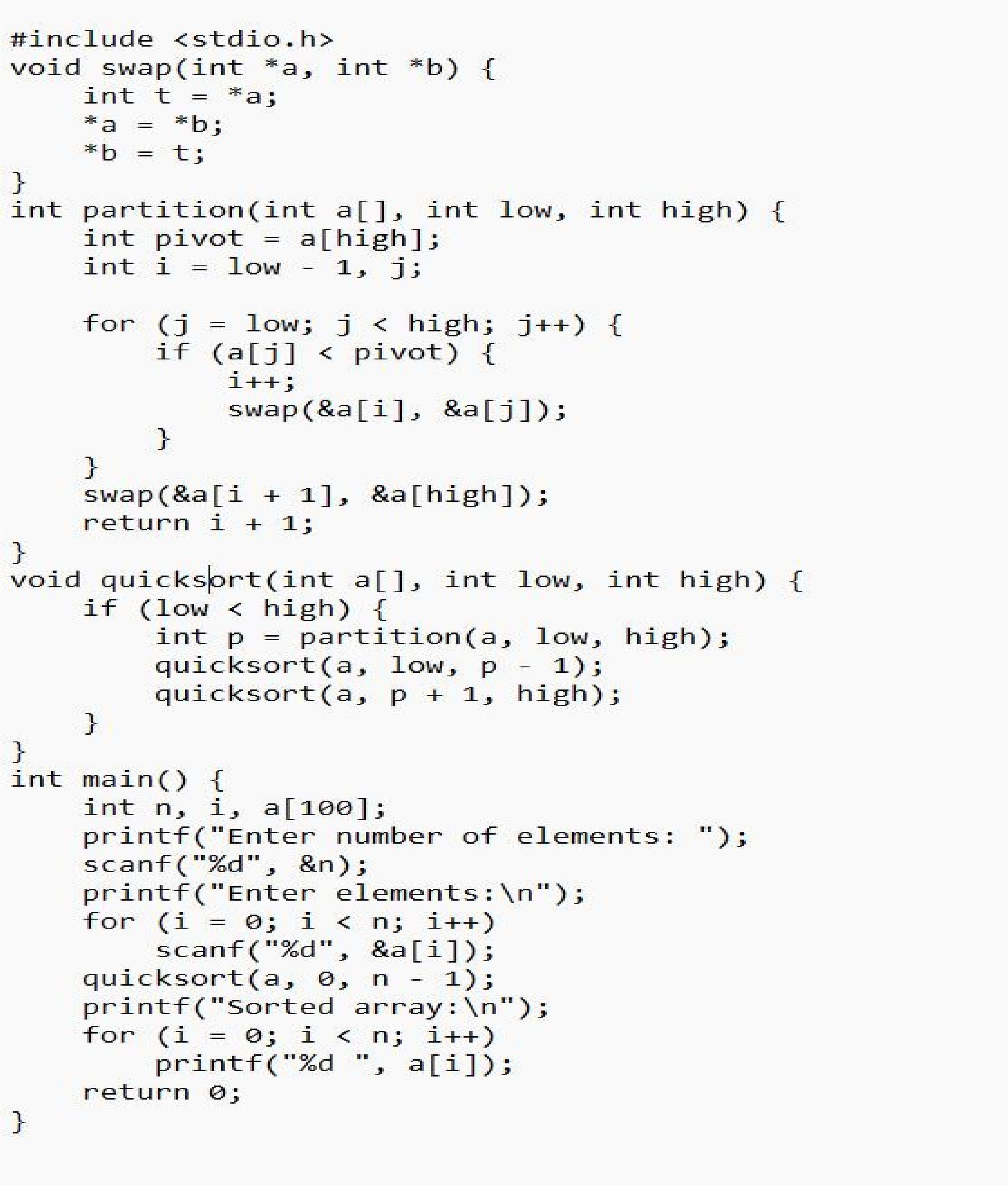


```
#include <stdio.h>
void merge(int a[], int l, int mid, int r) {
    int i = l, j = mid + 1, k = 0;
    int temp[100];
    while (i <= mid && j <= r) {
        if (a[i] < a[j])
            temp[k++] = a[i++];
        else
            temp[k++] = a[j++];
    }
    while (i <= mid)
        temp[k++] = a[i++];

    while (j <= r)
        temp[k++] = a[j++];

    k = 0;
    for (i = l; i <= r; i++)
        a[i] = temp[k++];
}
void mergesort(int a[], int l, int r) {
    if (l < r) {
        int mid = (l + r) / 2;
        mergesort(a, l, mid);
        mergesort(a, mid + 1, r);
        merge(a, l, mid, r);
    }
}
int main() {
    int n, i, a[100];
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter elements:\n");
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);
    mergesort(a, 0, n - 1);
    printf("Sorted array:\n");
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    return 0;
}
```

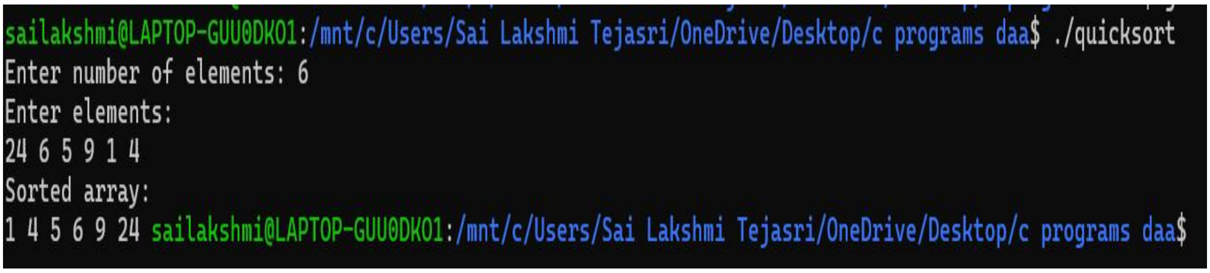


```
sailakshmi@LAPTOP-GUU0DK01:/mnt/c/Users/Sai Lakshmi Tejasri/OneDrive/Desktop/c programs daa$ gcc mergesort.c -o mergesort
sailakshmi@LAPTOP-GUU0DK01:/mnt/c/Users/Sai Lakshmi Tejasri/OneDrive/Desktop/c programs daa$ ./mergesort
Enter number of elements: 5
Enter elements:
9 3 1 6 2
Sorted array:
1 2 3 6 9 sailakshmi@LAPTOP-GUU0DK01:/mnt/c/Users/Sai Lakshmi Tejasri/OneDrive/Desktop/c programs daa$
```

2) QUICK SORT

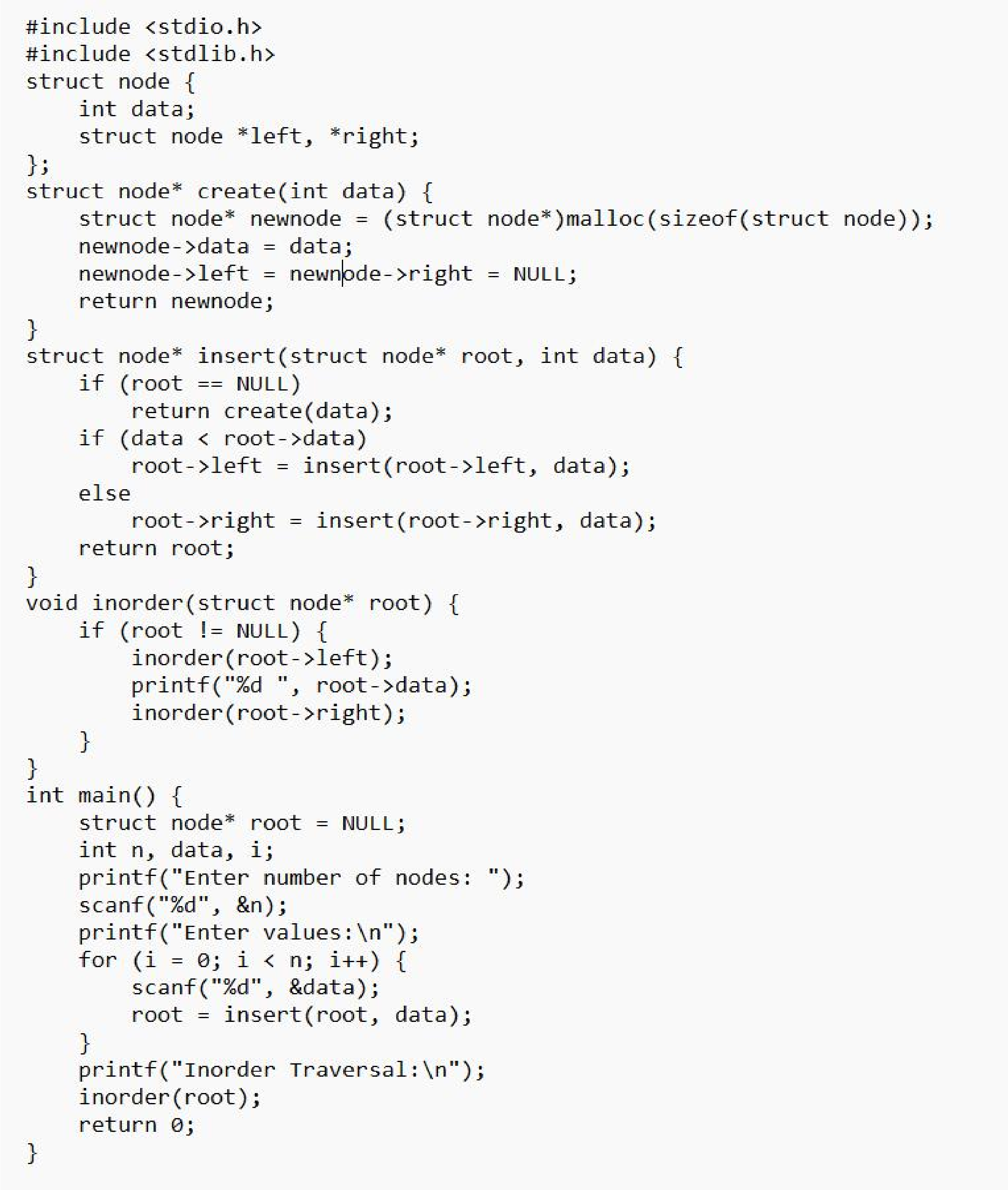


```c
#include <stdio.h>
void swap(int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}
int partition(int a[], int low, int high) {
    int pivot = a[high];
    int i = low - 1, j;

    for (j = low; j < high; j++) {
        if (a[j] < pivot) {
            i++;
            swap(&a[i], &a[j]);
        }
    }
    swap(&a[i + 1], &a[high]);
    return i + 1;
}
void quicksort(int a[], int low, int high) {
    if (low < high) {
        int p = partition(a, low, high);
        quicksort(a, low, p - 1);
        quicksort(a, p + 1, high);
    }
}
int main() {
    int n, i, a[100];
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter elements:\n");
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);
    quicksort(a, 0, n - 1);
    printf("Sorted array:\n");
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    return 0;
}
```

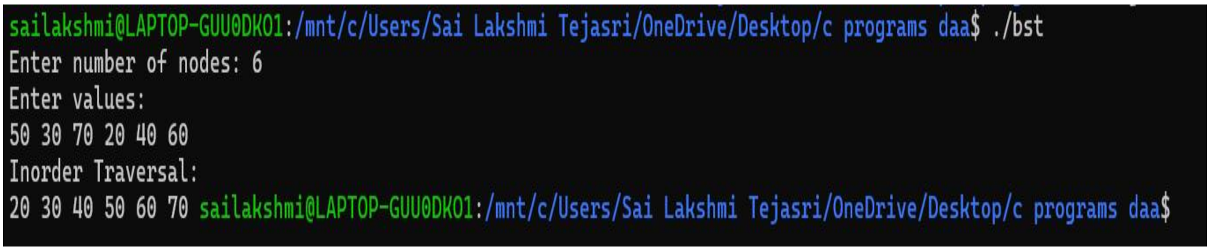


```
sailakshmi@LAPTOP-GUU0DK01:/mnt/c/Users/Sai Lakshmi Tejasri/OneDrive/Desktop/c programs daa$ ./quicksort
Enter number of elements: 6
Enter elements:
24 6 5 9 1 4
Sorted array:
1 4 5 6 9 24 sailakshmi@LAPTOP-GUU0DK01:/mnt/c/Users/Sai Lakshmi Tejasri/OneDrive/Desktop/c programs daa$
```

3)BINARY SEARCH TREE



```c
#include <stdio.h>
#include <stdlib.h>
struct node {
    int data;
    struct node *left, *right;
};
struct node* create(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 create(data);
    if (data < root->data)
        root->left = insert(root->left, data);
    else
        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, data, i;
    printf("Enter number of nodes: ");
    scanf("%d", &n);
    printf("Enter values:\n");
    for (i = 0; i < n; i++) {
        scanf("%d", &data);
        root = insert(root, data);
    }
    printf("Inorder Traversal:\n");
    inorder(root);
    return 0;
}
```



```
sailakshmi@LAPTOP-GUU0DK01:/mnt/c/Users/Sai Lakshmi Tejasri/OneDrive/Desktop/c programs daa$ ./bst
Enter number of nodes: 6
Enter values:
50 30 70 20 40 60
Inorder Traversal:
20 30 40 50 60 70 sailakshmi@LAPTOP-GUU0DK01:/mnt/c/Users/Sai Lakshmi Tejasri/OneDrive/Desktop/c programs daa$
```