

Name : Saloni Satappa Bailkar
Roll no: COBA013

HPC Lab Assignment No. 01(a)

Code:

```
#include <iostream>
#include <queue>

using namespace std;

class node
{
public:
    node *left, *right;
    int data;
};

node *insert(node *root, int data)
{
    if (!root)
    {
        root = new node;
        root->left = NULL;
        root->right = NULL;
        root->data = data;
        return root;
    }
    queue<node *> q;
    q.push(root);

    while (!q.empty())
    {
        node *temp = q.front();
        q.pop();

        if (temp->left == NULL)
        {
            temp->left = new node;
            temp->left->left = NULL;
            temp->left->right = NULL;
            temp->left->data = data;
            return root;
        }
    }
}
```

```

        else
        {
            q.push(temp->left);
        }
        if (temp->right == NULL)
        {
            temp->right = new node;
            temp->right->left = NULL;
            temp->right->right = NULL;
            temp->right->data = data;
            return root;
        }
        else
        {
            q.push(temp->right);
        }
    }

    return root; // Return root if no insertion happened
}

```

```

void bfs(node *head)
{
    if (!head)
        return;

    queue<node *> q;
    q.push(head);

    while (!q.empty())
    {
        node *currNode = q.front();
        q.pop();
        cout << "\t" << currNode->data;

        if (currNode->left)
            q.push(currNode->left);
        if (currNode->right)
            q.push(currNode->right);
    }
}

```

```

int main()
{

```

```

node *root = NULL;
int data;
char ans;

do
{
    cout << "\n Enter the data => ";
    cin >> data;

    root = insert(root, data);

    cout << "Do you want to insert one or more nodes? (y/n) ";
    cin >> ans;

} while (ans == 'y' || ans == 'Y');

bfs(root);

return 0;
}

```

Output:

The screenshot shows the Eclipse IDE interface. The Project Explorer on the left lists various projects and files. The main editor window displays the source code of a C++ program. The Console window at the bottom shows the output of the program's execution. The output consists of five iterations of the program asking for data and whether to insert more nodes. The data values entered are 4, 10, 7, 12, and 15. The program terminates with an exit value of 0.

```

1 #include <iostream>
2 #include <queue>
<terminated> (exit value: 0) First2.exe [C/C++ Application] C:\Users\hp\workspace\abc\First2\Debug\First2.exe (11/03/24, 10:15 pm)

Enter the data => 4
Do you want to insert one or more nodes? (y/n) y

Enter the data => 10
Do you want to insert one or more nodes? (y/n) y

Enter the data => 7
Do you want to insert one or more nodes? (y/n) y

Enter the data => 12
Do you want to insert one or more nodes? (y/n) y

Enter the data => 15
Do you want to insert one or more nodes? (y/n) n
4      10      7      12      15

```

Name : Saloni Satappa Bailkar
Roll no: COBA013

HPC Lab Assignment No. 1(b)

Code:

```
#include <iostream>
#include <vector>
#include <stack>
using namespace std;

const int MAX = 100000;
vector<int> graph[MAX];
bool visited[MAX];

void dfs(int node) {
    stack<int> s;
    s.push(node);

    while (!s.empty()) {
        int curr_node = s.top();
        s.pop();

        if (!visited[curr_node]) {
            visited[curr_node] = true;
            for (int i = 0; i < graph[curr_node].size(); i++) {
                int adj_node = graph[curr_node][i];
                if (!visited[adj_node]) {
                    s.push(adj_node);
                }
            }
        }
    }
}

int main() {
    int n, m, start_node;
    cin >> n >> m >> start_node;

    for (int i = 0; i < m; i++) {
```

```

int u, v;
cin >> u >> v;
graph[u].push_back(v);
graph[v].push_back(u);
}

for (int i = 0; i < n; i++) {
    visited[i] = false;
}

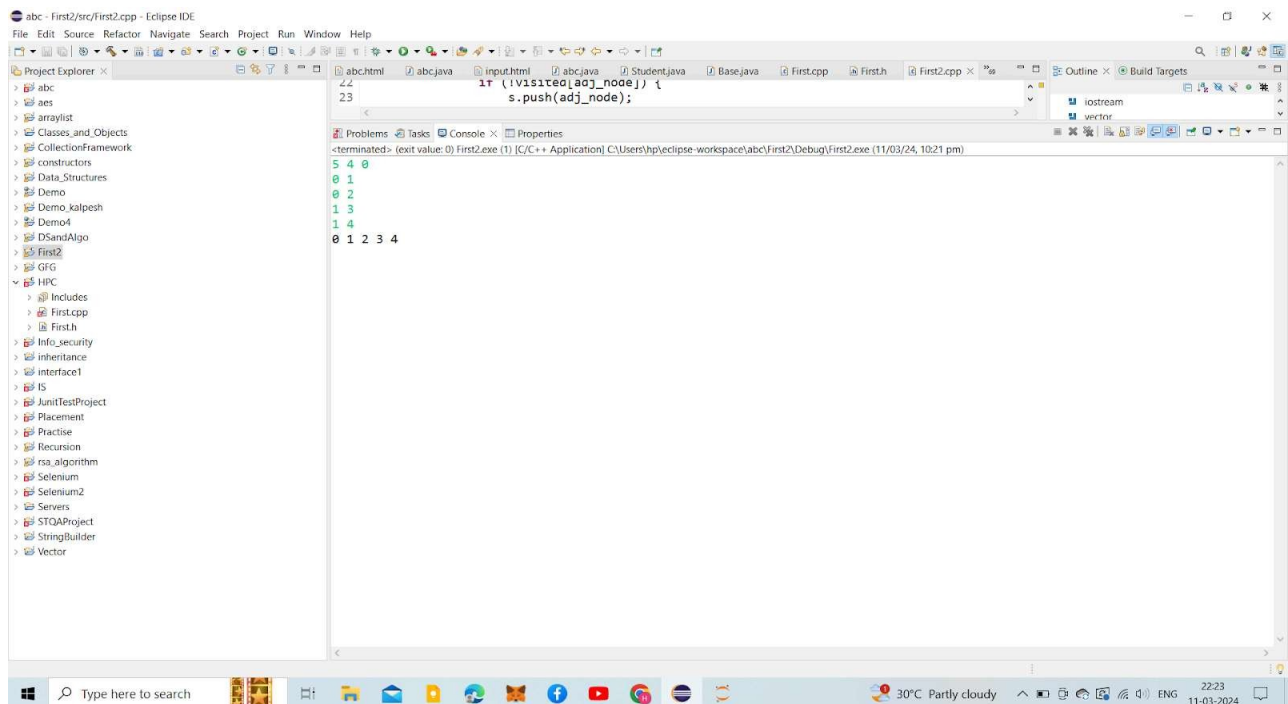
dfs(start_node);

for (int i = 0; i < n; i++) {
    if (visited[i]) {
        cout << i << " ";
    }
}

return 0;
}

```

Output:



Name : Saloni Satappa Bailkar

Roll no: COBA013

HPC Lab Assignment No.02(a)

Code:

```
#include <iostream>
```

```
#include <omp.h>
```

```
void bubble(int *, int);
```

```
void swap(int &, int &);
```

```
void bubble(int *a, int n)
```

```
{
```

```
    bool swapped;
```

```
    for (int i = 0; i < n - 1; i++)
```

```
    {
```

```
        swapped = false;
```

```
        #pragma omp parallel for shared(a, n) reduction(||:swapped)
```

```
        for (int j = 0; j < n - i - 1; j++)
```

```
        {
```

```
            if (a[j] > a[j + 1])
```

```
            {
```

```
                swap(a[j], a[j + 1]);
```

```
                swapped = true;
```

```
            }
```

```
        }
```

```
        if (!swapped)
```

```
            break; // Array is already sorted
```

```
    }
```

```
}
```

```
void swap(int &a, int &b)
```

```
{
```

```
    int temp = a;
```

```
    a = b;
```

```
    b = temp;
```

```
}
```

```
int main()
```

```
{
```

```
int *a, n;
std::cout << "Enter total number of elements: ";
std::cin >> n;

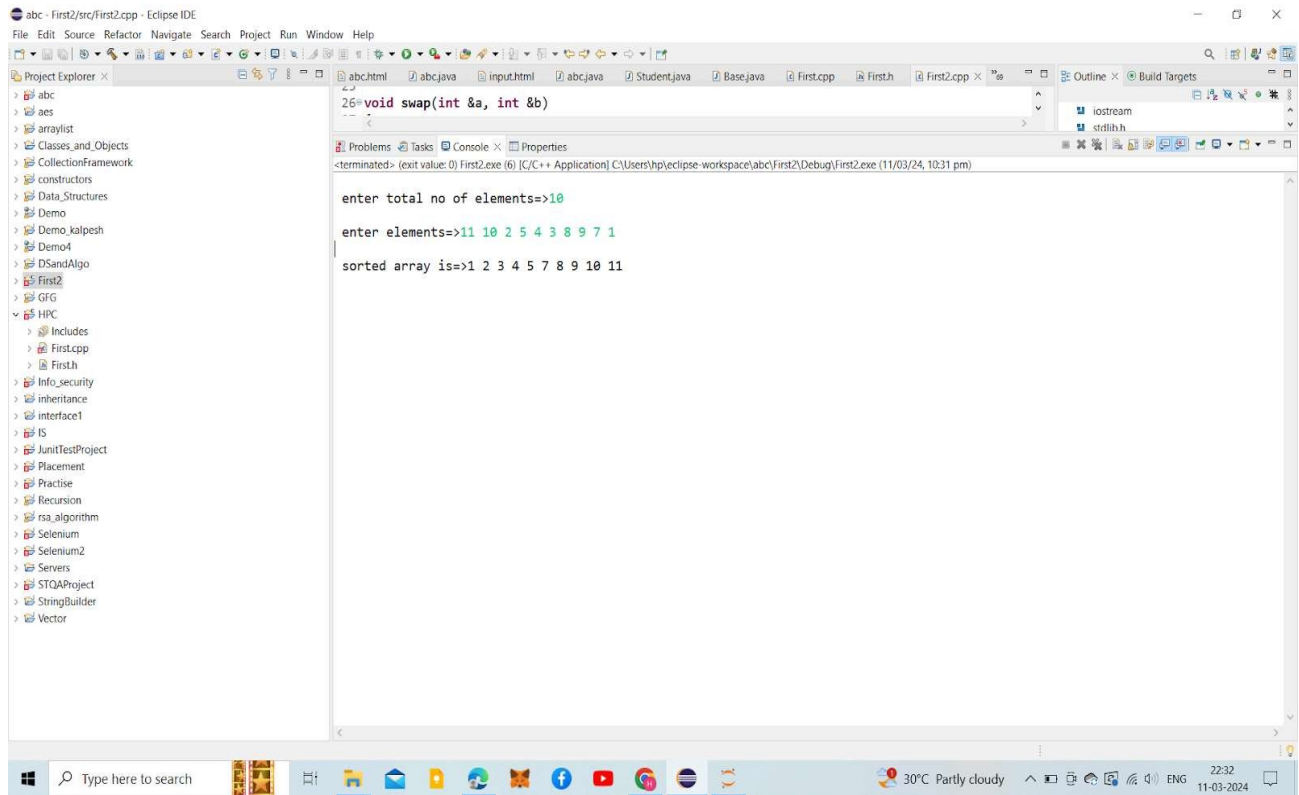
a = new int[n];
std::cout << "Enter elements: ";
for (int i = 0; i < n; i++)
{
    std::cin >> a[i];
}

bubble(a, n);

std::cout << "Sorted array: ";
for (int i = 0; i < n; i++)
{
    std::cout << a[i] << " ";
}
std::cout << std::endl;

delete[] a; // Deallocate memory
return 0;
}
```

Output:



The screenshot displays the Eclipse IDE interface. The Project Explorer on the left shows a project named 'abc' with various source files and folders. The main editor window shows a C++ file 'First2.cpp' with the following code:

```
26 void swap(int &a, int &b)
```

The Console window at the bottom shows the program's output:

```
<terminated> (exit value: 0) First2.exe (6) [C/C++ Application] C:\Users\hp\workspace\abc\First2\Debug\First2.exe (11/03/24, 10:31 pm)

enter total no of elements=>10

enter elements=>11 10 2 5 4 3 8 9 7 1

sorted array is=>1 2 3 4 5 7 8 9 10 11
```

The Windows taskbar at the bottom shows the system clock as 22:32 on 11-03-2024, along with weather and other system icons.

Name : Saloni Satappa Bailkar

Roll no: COBA013

HPC Lab Assignment No. 02(b)

Code:

```
#include <iostream>
#include <omp.h>

void merge(int arr[], int l, int m, int r) {
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;
    int 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 (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 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;
    std::cout << "Enter the number of elements: ";
    std::cin >> n;

    int arr[n];
    std::cout << "Enter " << n << " elements: ";
    for (int i = 0; i < n; i++)
        std::cin >> arr[i];

    std::cout << "Given array is: ";
    for (int i = 0; i < n; i++)
        std::cout << arr[i] << " ";
    std::cout << std::endl;

    double start = omp_get_wtime();
    mergeSort(arr, 0, n - 1);
    double stop = omp_get_wtime();

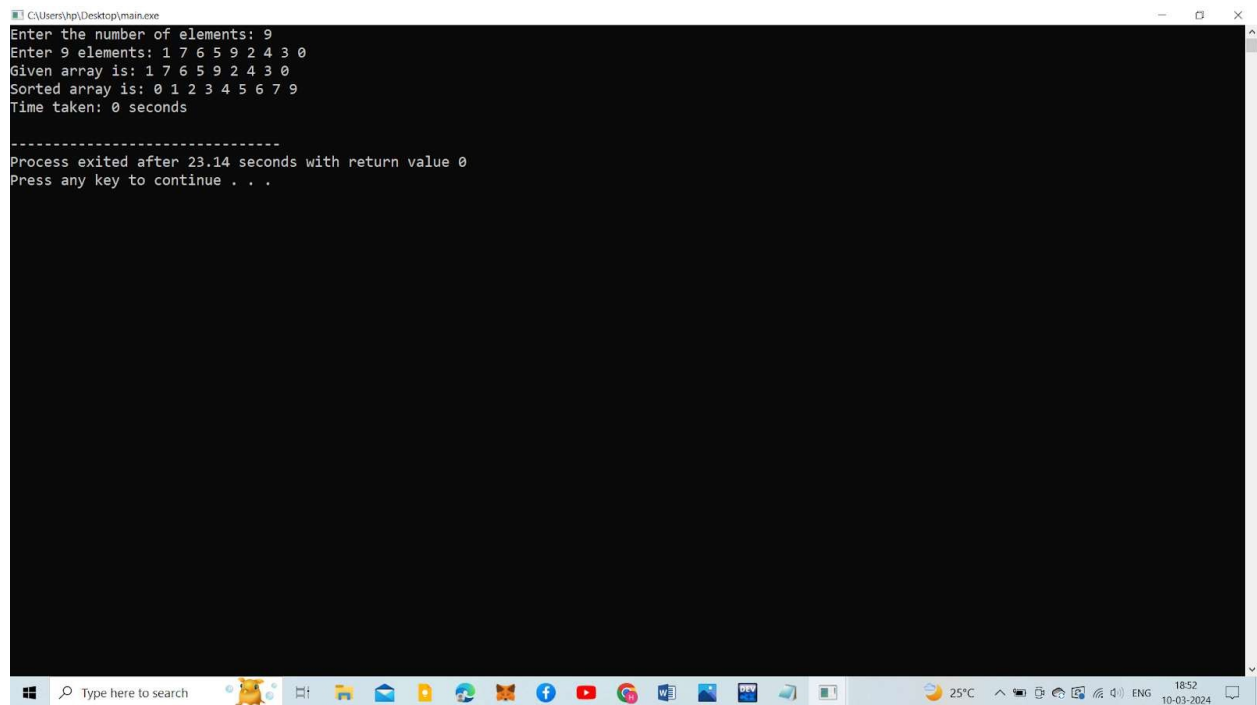
    std::cout << "Sorted array is: ";
    for (int i = 0; i < n; i++)
        std::cout << arr[i] << " ";
    std::cout << std::endl;
}

```

```
std::cout << "Time taken: " << stop - start << " seconds" << std::endl;

return 0;
}
```

Output:



```
C:\Users\hpl\Desktop\main.exe
Enter the number of elements: 9
Enter 9 elements: 1 7 6 5 9 2 4 3 0
Given array is: 1 7 6 5 9 2 4 3 0
Sorted array is: 0 1 2 3 4 5 6 7 9
Time taken: 0 seconds

-----
Process exited after 23.14 seconds with return value 0
Press any key to continue . . .
```

Name : Saloni Satappa Bailkar
Roll no: COBA013
HPC Lab Assignment No. 03

Code:

```
#include <iostream>
#include <omp.h>
#include <climits> // Include the <climits> header for INT_MAX and INT_MIN
using namespace std;

void min_reduction(int arr[], int n) {
    int min_value = INT_MAX; // Use INT_MAX from <climits>
    #pragma omp parallel for reduction(min: min_value)
    for (int i = 0; i < n; i++) {
        if (arr[i] < min_value) {
            min_value = arr[i];
        }
    }
    cout << "Minimum value: " << min_value << endl;
}

void max_reduction(int arr[], int n) {
    int max_value = INT_MIN; // Use INT_MIN from <climits>
    #pragma omp parallel for reduction(max: max_value)
    for (int i = 0; i < n; i++) {
        if (arr[i] > max_value) {
            max_value = arr[i];
        }
    }
    cout << "Maximum value: " << max_value << endl;
}

void sum_reduction(int arr[], int n) {
    int sum = 0;
    #pragma omp parallel for reduction(+: sum)
    for (int i = 0; i < n; i++) {
```

```

        sum += arr[i];
    }
    cout << "Sum: " << sum << endl;
}

void average_reduction(int arr[], int n) {
    int sum = 0;
    #pragma omp parallel for reduction(+: sum)
    for (int i = 0; i < n; i++) {
        sum += arr[i];
    }
    cout << "Average: " << (double)sum / n << endl;
}

int main() {
    int *arr, n;
    cout << "Enter total no of elements=>\n";
    cin >> n;
    arr = new int[n];
    cout << "Enter elements=>\n";
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    min_reduction(arr, n);
    max_reduction(arr, n);
    sum_reduction(arr, n);
    average_reduction(arr, n);

    delete[] arr; // Free the dynamically allocated memory

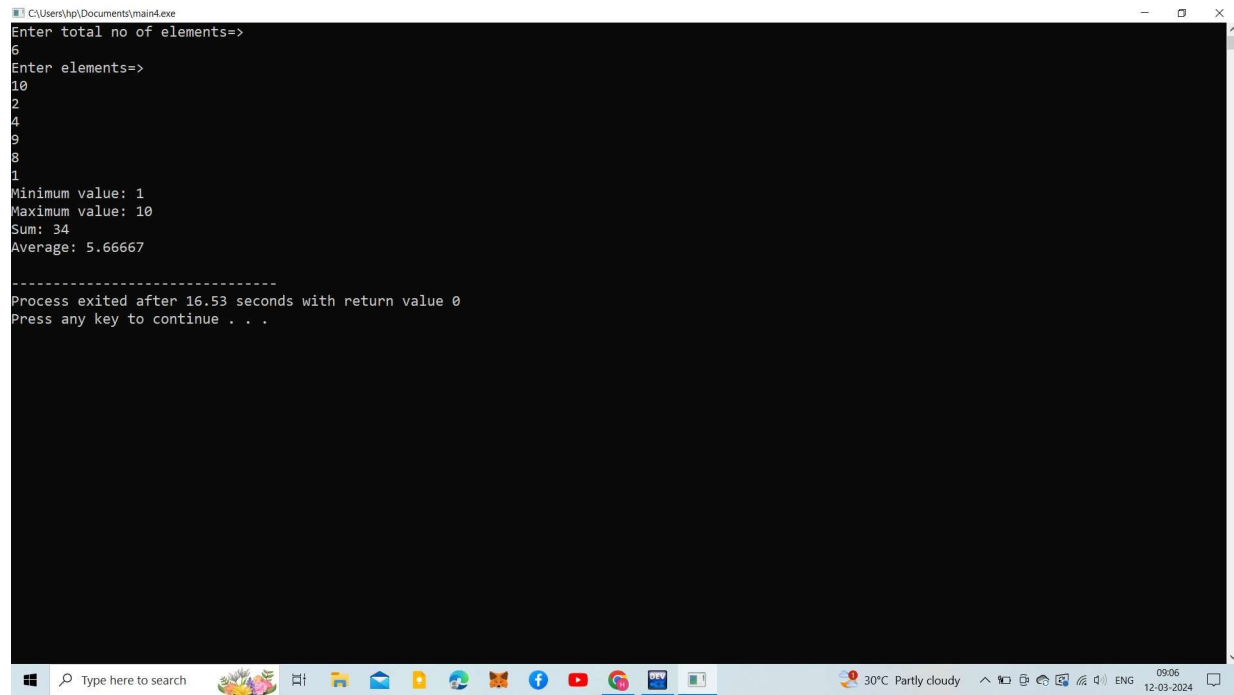
    return 0;
}

```

Output:

```
C:\Users\hpl\Documents\main4.exe
Enter total no of elements=>
6
Enter elements=>
10
2
4
9
8
1
Minimum value: 1
Maximum value: 10
Sum: 34
Average: 5.66667

-----
Process exited after 16.53 seconds with return value 0
Press any key to continue . . .
```

The image shows a Windows 10 desktop environment. A black command prompt window is open, displaying the output of a program. The program prompts for the total number of elements (6) and then for the elements themselves (10, 2, 4, 9, 8, 1). It then calculates and displays the minimum value (1), maximum value (10), sum (34), and average (5.66667). After a separator line, it shows the process exit message and a prompt to press any key to continue. The Windows taskbar is visible at the bottom, showing the search bar, several application icons, and system tray information including the date and time (09:06, 12-03-2024).