

LP-V

HPC Assignment No-1

BFS

```
#include <iostream>

#include <queue>

#include <omp.h>

using namespace std;

const int MAX = 1000;

int graph[MAX][MAX], visited[MAX];

void bfs(int start, int n) {

    queue<int> q;

    visited[start] = 1;

    q.push(start);

    while (!q.empty()) {

        int curr = q.front();

        q.pop();

#pragma omp parallel for shared(graph, visited, q) schedule(dynamic)

        for (int i = 0; i < n; i++) {

            if (graph[curr][i] && !visited[i]) {

                visited[i] = 1;

                q.push(i);

            }

        }

    }

}

int main() {

    int n, start;

    cout << "Enter number of vertices: ";

    cin >> n;

    cout << "Enter adjacency matrix:\n";

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

```

        for (int j = 0; j < n; j++) {
            cin >> graph[i][j];
        }
    }

    cout << "Enter starting vertex: ";

    cin >> start;

#pragma omp parallel num_threads(4)

    {
        bfs(start, n);
    }

    cout << "BFS traversal: ";

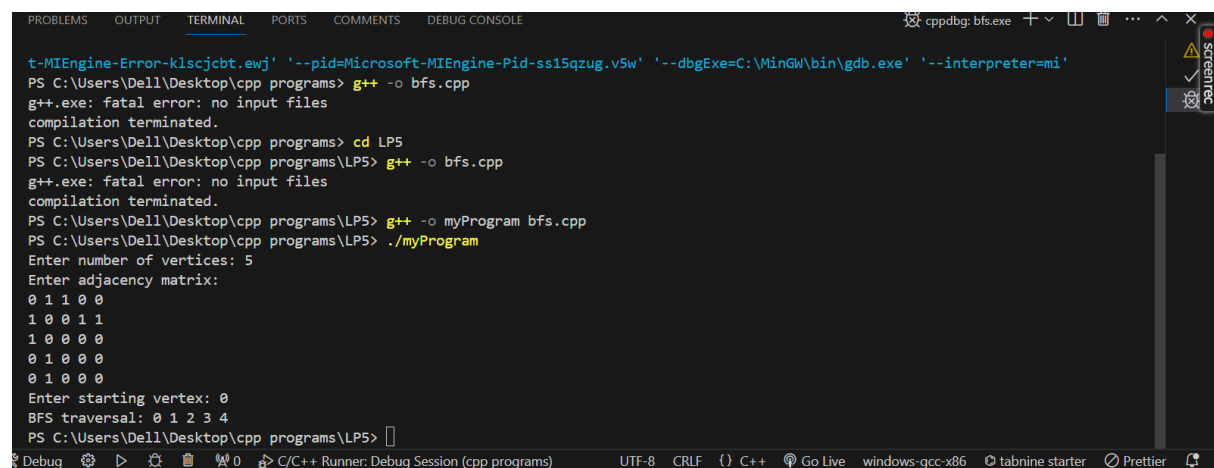
    for (int i = 0; i < n; i++) {
        if (visited[i])
            cout << i << " ";
    }

    cout << endl;

    return 0;
}

```

OUTPUT



```

t-MIEngine-Error-klscjcbt.ewj' '--pid=Microsoft-MIEngine-Pid-ss15qzug.v5w' '--dbgExe=C:\MinGW\bin\gdb.exe' '--interpreter=mi'
PS C:\Users\Dell\Desktop\cpp programs> g++ -o bfs.cpp
g++.exe: fatal error: no input files
compilation terminated.
PS C:\Users\Dell\Desktop\cpp programs> cd LP5
PS C:\Users\Dell\Desktop\cpp programs\LP5> g++ -o bfs.cpp
g++.exe: fatal error: no input files
compilation terminated.
PS C:\Users\Dell\Desktop\cpp programs\LP5> g++ -o myProgram bfs.cpp
PS C:\Users\Dell\Desktop\cpp programs\LP5> ./myProgram
Enter number of vertices: 5
Enter adjacency matrix:
0 1 0 0
1 0 0 1
1 0 0 0
0 1 0 0
0 1 0 0
Enter starting vertex: 0
BFS traversal: 0 1 2 3 4
PS C:\Users\Dell\Desktop\cpp programs\LP5>

```

DFS

```
#include <iostream>

#include <stack>

#include <omp.h>

using namespace std;

const int MAX = 1000;

int graph[MAX][MAX], visited[MAX];

void dfs(int start, int n) {

    stack<int> s;

    s.push(start);

    while (!s.empty()) {

        int curr = s.top();

        s.pop();

        if (!visited[curr]) {

            visited[curr] = 1;

#pragma omp parallel for shared(graph, visited, s) schedule(dynamic)

            for (int i = 0; i < n; i++) {

                if (graph[curr][i] && !visited[i]) {

                    s.push(i);

                }

            }

        }

    }

}

int main() {

    int n, start;

    cout << "Enter number of vertices: ";

    cin >> n;

    cout << "Enter adjacency matrix:\n";

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {
```

```

        cin >> graph[i][j];
    }
}

cout << "Enter starting vertex: ";

cin >> start;

#pragma omp parallel num_threads(4)

{
    dfs(start, n);
}

cout << "DFS traversal: ";

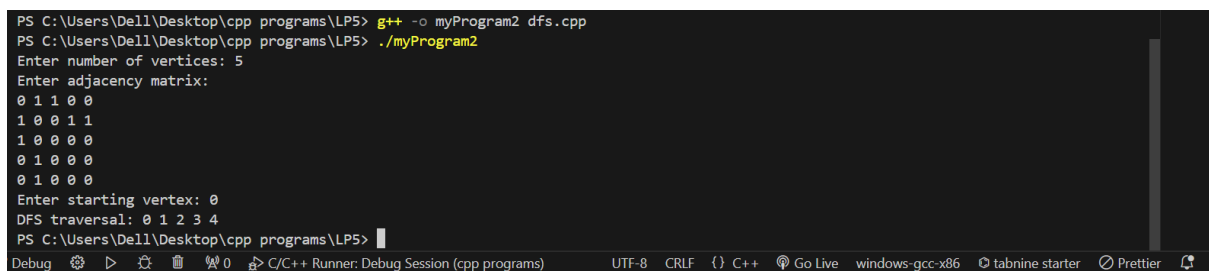
for (int i = 0; i < n; i++) {
    if (visited[i])
        cout << i << " ";
}

cout << endl;

return 0;
}

```

OUTPUT:



```

PS C:\Users\De11\Desktop\cpp programs\LP5> g++ -o myProgram2 dfs.cpp
PS C:\Users\De11\Desktop\cpp programs\LP5> ./myProgram2
Enter number of vertices: 5
Enter adjacency matrix:
0 1 1 0 0
1 0 0 1 1
1 0 0 0 0
0 1 0 0 0
0 1 0 0 0
Enter starting vertex: 0
DFS traversal: 0 1 2 3 4
PS C:\Users\De11\Desktop\cpp programs\LP5>

```

HPC Assignment No-2

Bubble Sort

```
#include<iostream>
#include<stdlib.h>
#include<omp.h>
using namespace std;
void bubble(int *, int);
void swap(int &, int &);
void bubble(int *a, int n)
{
    for( int i = 0; i < n; i++ )
    {
        int first = i % 2;

        #pragma omp parallel for shared(a,first)
        for( int j = first; j < n-1; j += 2 )
        {
            if( a[j] > a[j+1] )
            {
                swap( a[j], a[j+1] );
            }
        }
    }
}
void swap(int &a, int &b)
{
    int test;
    test=a;
    a=b;
    b=test;
}
int main()
{
    int *a,n;
    cout<<"\n enter total no of elements=>";
    cin>>n;
    a=new int[n];

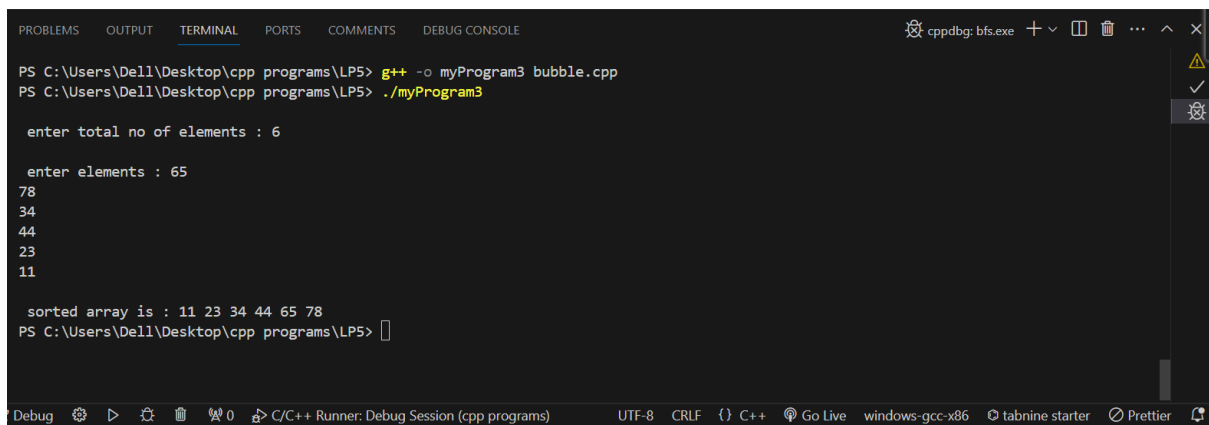
    cout<<"\n enter elements=>";

    for(int i=0;i<n;i++)
    {
        cin>>a[i];
    }
}
```

```
bubble(a,n);
cout<<"\n sorted array is=>";
for(int i=0;i<n;i++)
{
    cout<<a[i]<<endl;
}

return 0;
}
```

Output :

A screenshot of a Visual Studio Code window showing a C++ program being executed in a debugger. The window has a dark theme. At the top, there are tabs for PROBLEMS, OUTPUT, TERMINAL, PORTS, COMMENTS, and DEBUG CONSOLE. The TERMINAL tab is active, showing the command prompt output. The program prompts the user to enter the total number of elements (6) and then the elements themselves (78, 34, 44, 23, 11). The output shows the sorted array: 11 23 34 44 65 78. The bottom status bar shows the current file is 'C:/C++ Runner: Debug Session (cpp programs)', the encoding is UTF-8, the line ending is CRLF, and the language is C++. Other icons for Go Live, windows-gcc-x86, tabnine starter, and Prettier are also visible.

```
PROBLEMS OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE
cppdbg: bfs.exe + - [ ] ... ^ x
PS C:\Users\Dell\Desktop\cpp programs\LP5> g++ -o myProgram3 bubble.cpp
PS C:\Users\Dell\Desktop\cpp programs\LP5> ./myProgram3

enter total no of elements : 6

enter elements : 65
78
34
44
23
11

sorted array is : 11 23 34 44 65 78
PS C:\Users\Dell\Desktop\cpp programs\LP5> [ ]

Debug [ ] [ ] [ ] [ ] 0 C/C++ Runner: Debug Session (cpp programs) UTF-8 CRLF ( ) C++ Go Live windows-gcc-x86 tabnine starter Prettier [ ]
```

Merge Sort

```
#include<iostream>

#include<stdlib.h>

#include<omp.h>

using namespace std;

void mergesort(int a[],int i,int j);

void merge(int a[],int i1,int j1,int i2,int j2);

void mergesort(int a[],int i,int j)
{
    int mid;
    if(i<j)
    {
        mid=(i+j)/2;
        #pragma omp parallel sections
        {
            #pragma omp section
            {
                mergesort(a,i,mid);
            }

            #pragma omp section
            {
                mergesort(a,mid+1,j);
            }
        }
        merge(a,i,mid,mid+1,j);
    }
}

void merge(int a[],int i1,int j1,int i2,int j2)
{
    int temp[1000];
    int i,j,k;
```

```

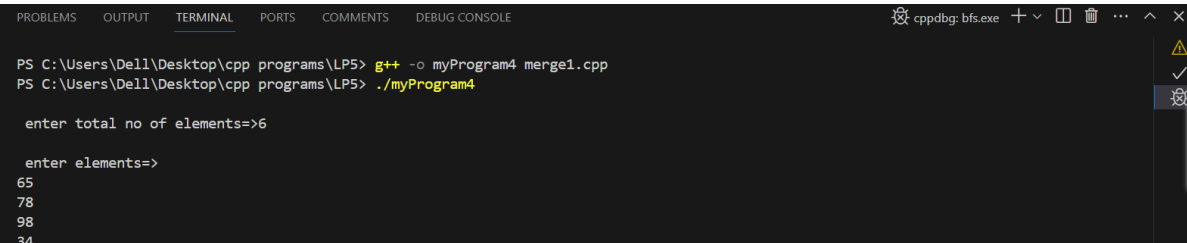
i=i1;
j=i2;
k=0;
while(i<=j1 && j<=j2)
{
    if(a[i]<a[j])
    {
        temp[k++]=a[i++];
    }
    else
    {
        temp[k++]=a[j++];
    }
}
while(i<=j1)
{
    temp[k++]=a[i++];
}
while(j<=j2)
{
    temp[k++]=a[j++];
}
for(i=i1,j=0;i<=j2;i++,j++)
{
    a[i]=temp[j];
}
}

int main()
{
    int *a,n,i;

    cout<<"\n enter total no of elements=>";

```


●



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal displays the execution of a C++ program. The user has entered the total number of elements (6) and the elements themselves (65, 78, 98, 34, 11, 23). The program has sorted the array and printed the sorted elements (11, 23, 34, 65, 78, 98). The terminal output is as follows:

```
PS C:\Users\Dell\Desktop\cpp programs\LP5> g++ -o myProgram4 merge1.cpp
PS C:\Users\Dell\Desktop\cpp programs\LP5> ./myProgram4

enter total no of elements=>6

enter elements=>
65
78
98
34
11
23

sorted array is=>
11
23
34
65
78
98
PS C:\Users\Dell\Desktop\cpp programs\LP5> 
```

The Visual Studio Code interface includes a sidebar on the left with tabs for PROBLEMS, OUTPUT, TERMINAL, PORTS, COMMENTS, and DEBUG CONSOLE. The TERMINAL tab is active. The status bar at the bottom shows the current file is 'C/C++ Runner: Debug Session (cpp programs)', the encoding is UTF-8, the language is C++, and various extensions like Go Live, windows-gcc-x86, tabnine starter, and Prettier are installed.