**LAB SHEET 4.B**

#include <iostream>

#include "omp.h"

#include <cmath>

using namespace std;

int main()

{

int N,id;

cout<<"\nEnter the No of nodes in your tree: ";

cin>>N;

int \*s= new int[N];

int \*depth= new int[N];

cout<<"Enter the edge list: ";

for(int x=0; x<N; x++)

{

cin>>s[x];

depth[x]=1;

}

clock\_t start, end;

double cpu\_time\_used;

start = clock();

omp\_set\_num\_threads(N);

#pragma omp parallel

{

#pragma omp critical

{

id=omp\_get\_thread\_num();

for(int i=0; i<=ceil(log (N)); i++)

if(s[id]!=-1)

{

cout<<"\nid: "<<id<<" Val of s[s[id]]: "<<s[s[id]];

depth[id]+=depth[s[id]];

cout<<" Depth: "<<depth[id];

s[id]=s[s[id]];

}

}

}

cout<<"\nDepth: ";

for(int i=0; i<N; i++)

cout<<" "<<depth[i];

end = clock();

cpu\_time\_used = ((double) (end - start))/ CLOCKS\_PER\_SEC;

cout<<"\n\nTime Taken by New\_ET: "<<cpu\_time\_used<<"\n"; //<<cpu\_time\_used/1000<<"\n";....result in msec

return 0;

}

**OUTPUT:**

****

**/\***

**\*NIKITA JAIN**

**\*MTECH (CSD)**

**\*14535031**

**\*/**