## Assignment 1B

```
!nvcc --version
%env OMP_NUM_THREADS=3
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2023 NVIDIA Corporation
Built on Tue_Aug_15_22:02:13_PDT_2023
Cuda compilation tools, release 12.2, V12.2.140
Build cuda_12.2.r12.2/compiler.33191640_0
env: OMP_NUM_THREADS=3
```

```
%%writefile dfs.cpp

#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;
      cout << curr << " ";


        #pragma omp parallel for shared(graph, visited, s) schedule(dynamic)
        for(int i=0; i<n; i++) {
          if(graph[curr][i] == 1 && visited[i] == 0) {
            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;

  cout << "DFS traversal: ";

  #pragma omp parallel num_threads(4)
  {
    dfs(start, n);
  }

  cout << endl;
  return 0;
}
```

```
Overwriting dfs.cpp
```

```
!g++ dfs.cpp -o dfs -fopenmp
```

```
!./dfs
```

```
Enter number of vertices: 5
Enter adjacency matrix:
0 1 1 0 0
1 0 0 1 0
1 0 0 1 1
0 1 1 0 1
0 0 1 1 0
Enter starting vertex: 0
DFS traversal: 0 2 4 3 1
```