[CENG 315 ALL Sections] Algorithms

Dashboard / My courses / 571 - Computer Engineering / CENG 315 ALL Sections / November 27 - December 3 / THE5

Navigation

- Dashboard
  - Site home
  - Site pages
  - My courses
    - 571 - Computer Engineering
      - CENG 223 All Sections
      - CENG 223 Section 2
      - CENG 315 ALL Sections
        - Participants
        - Competencies
        - Grades
        - General
        - October 2 - October 8
        - October 9 - October 15
        - October 15 - October 22
        - October 23 - October 29
        - October 30 - November 5
        - November 6 - November 12
        - November 13 - November 19
        - November 20 - November 26
        - November 27 - December 3
          - THE5
            - Description
            - Submission view
        - December 3 - December 10
        - December 11 - December 17
        - December 18 - December 24
        - December 25 - December 31
        - January 1 - January 7
        - January 8 - January 14
      - CENG 315 Section 1

Description | Submission view

Grade

Reviewed on Saturday, December 9, 2023, 9:40 PM by Automatic grade

Grade: 92.00 / 100.00

Assessment report [-]

[-]**Output of make**

```
g++ -c -o sol5.o sol5.cpp
```

[-]**For input 01:**

Compilable check: Correct

Result vectors: Correct

[-]**For input 02:**

Compilable check: Correct

Result vectors: Correct

[-]**For input 03:**

Compilable check: Correct

Result vectors: Correct

[-]**For input 04:**

Compilable check: Correct

Result vectors: Correct

[-]**For input 05:**

Compilable check: Correct

Result vectors: Correct

[-]**For input 06:**

Compilable check: Correct

Result vectors: Correct

[-]**For input 07:**

Compilable check: Correct

Result vectors: Correct

[-]**For input 08:**

Compilable check: Correct

Result vectors: Correct

[-]**For input 09:**

Compilable check: Correct

Result vectors: Incorrect

[-]**For input 10:**

Compilable check: Correct

Result vectors: Correct

Submitted on Saturday, December 9, 2023, 9:40 PM (Download)

the5.cpp

```cpp
#include "the5.h"

// do not add extra libraries here

bool recurs(int index, int nElements,std::vector<bool>& isVisited,const std::vector<std::vector<int>>& dependencyMatrix, std::vector<int>& compileOrder,std:
{
    if(isVisited[index] == true)
        return 0;
    bool isInCycle = false;
    int indexOfCycleBegin;
    for(int i = 0; i < isCycle.size();i++)
        if(isCycle[i] == index){
            isInCycle = true;
            indexOfCycleBegin = i;
        }
    if(isInCycle)
    {
        /*for(int i = 0 ; i <indexOfCycleBegin; i++)
            isCycle.erase(isCycle.begin());
        cyclicDependencies.push_back(isCycle);
        isCycle.clear();*/
        return 1;
    }
    else
    {
        isCycle.push_back(index);
    }
    bool output = 0;
    for (int i = 0; i<nElements; i++)
    {
        if (dependencyMatrix[i][index] == 1)
            if(recurs(i,nElements,isVisited,dependencyMatrix, compileOrder,isCycle,cyclicDependencies)){
                output++;


            }
    }

    compileOrder.push_back(index);
    isVisited[index] = true;
    return output;
}

void run(const std::vector<std::vector<int>>& dependencyMatrix,
void run(const std::vector<std::vector<int>>& dependencyMatrix,
        bool& isCompilable,
        std::vector<int>& compileOrder,
        std::vector<std::vector<int>>& cyclicDependencies){

    int nElements = dependencyMatrix[0].size();
    std::vector<bool> isVisited(nElements,false);
    isCompilable = true;
    for (int i = 0; i<nElements; i++)
    {
        std::vector<int> isCycle;
        if (recurs(i,nElements,isVisited,dependencyMatrix, compileOrder,isCycle,cyclicDependencies))
        {
            isCompilable = false;
           cyclicDependencies.push_back(isCycle);
            isCycle.clear();
        }
    }

}
```

test.cpp

```
#include <iostream>
#include <fstream>
#include "the5.h"


void test(){
    char inp_file[] = "inp01.txt"; // 01 to 10 is available
    int size;
    std::ifstream infile (inp_file);
    if(!infile.is_open()){
        std::cout << "Input file cannot be opened" << std::endl;
        std::cout << "File name: " << inp_file << std::endl;
        return;
    }
    infile >> size;
    std::vector<std::vector<int>> input_array(size, std::vector<int>(size, 0));
    std::vector<int> compilation_order;
    std::vector<std::vector<int>> cyclic_dependencies;
    bool compilable = false;

    for(int idx=0; idx < size ; idx++)
        for(int idy=0; idy < size; idy++)
            infile >> input_array[idx][idy];

    std::cout << "Number of files: " << size << std::endl <<
                 "Dependency matrix: [" << std::endl;
    for(int idx=0; idx < size; idx++){
        std::cout << "                        [";
        for(int idy=0; idy < size-1; idy++){
            std::cout << input_array[idx][idy] << ", ";
        }
        std::cout << input_array[idx][size-1] << "]" << std::endl;
    }
    std::cout << "                   ]"<< std::endl;

    run(input_array, compilable, compilation_order, cyclic_dependencies);
    if(compilable){
        std::cout << "Compilation is possible." << std::endl;
        std::cout <<"Sorted order: {";
        for(int idx=0; idx < size-1; idx++) std::cout << compilation_order[idx] << ", ";
        std::cout << compilation_order[size - 1] << "}" << std::endl;
    }
    else{
        std::cout << "Compilation is not possible. Cyclic dependencies: " << std::endl;
        for(int idx=0; idx < (int)cyclic_dependencies.size(); idx++){
            std::cout << (idx+1) << ": ";
            for(int idy=0; idy < (int)cyclic_dependencies[idx].size()-1; idy++){
                std::cout << cyclic_dependencies[idx][idy] << " ";
            }
            std::cout << cyclic_dependencies[idx].back() << std::endl;

        }
    }
    std::cout << "--------------------------------------------" << std::endl;
    infile.close();
}

int main(){
    test();
    return 0;
}
```

the5.h

```
#ifndef THE5_THE5_H
#define THE5_THE5_H
#include <vector>
#include <memory>
#include <utility>
//this file will be overwritten during execution
//changes made in this file will not affectyour grade or execution
//it is shared with you so that you can work on a local environment if needed

void run(const std::vector<std::vector<int>>& dependencyMatrix,
        bool& isCompilable,
        std::vector<int>& compileOrder,
        std::vector<std::vector<int>>& cyclicDependencies);
#endif //THE5_THE5_H
```

VPL