

CMPE 343
Fall 2022
Programming Homework 2

This assignment is due by 23:55 on Friday 11, November 2022.

You are welcome to ask your HW related questions by joining the recitation hours given below:

- CMPE343-HW1-OfficeHour1: October 31 Monday, 19:00-20:00 PM, Zoom ID: <https://tedu.zoom.us/j/99691748310>
- CMPE343-HW1-OfficeHour2: November 07 Monday, 19:00-20:00 PM, Zoom ID: <https://tedu.zoom.us/j/96969884179>

Note: Please make sure that you have read the HW document well before participating.

PROGRAMMING TASK

In this part, you must implement your own graph data structure by taking inspiration from your textbook and use it to help to solve problem. You are not allowed to use any external library or .jar file. Any solutions without using graph data structure are not evaluated!

Question 1 (25 points):

You are working in the office of TEDU Student Affairs, and you are asked to develop a program that shows which course is a prerequisite for another course. Assume that we have 7 courses in our department and these course codes are: 112,113,114, 223,224,225, and 315.

- 112 is prerequisite for 113.
- 113 is prerequisite for 114, (112 is also prerequisite for 114)
- 223 is prerequisite for 224
- 224 is prerequisite for 225, (223 is also prerequisite for 225)

Your program must take prerequisite schedule and one course as inputs. It must print all prerequisite courses for the given course in ascending order. If there is no prerequisite course for the given course, it must print “There is no prerequisite for this course!”.

Sample run:

<pre>7 // number of courses 112,113,114,223,224,225,315 // courses 4 // # number of prerequisites 112-113 113-114 223-224 224-225 112 // enter a course to check prerequisites There is no prerequisite for this course!</pre>
<pre>7 // number of courses 112,113,114,223,224,225,315 // courses 4 // # number of prerequisites 112-113 113-114 223-224 224-225 113 // enter a course to check prerequisites 112</pre>
<pre>7 // number of courses 112,113,114,223,224,225,315 // courses 4 // # number of prerequisites 112-113 113-114 223-224 224-225 112-225 225 // enter a course to check prerequisites 112,223,224</pre>

Note that, for above inputs, even if 112-114 conditions is not specified in the inputs, 112 is prerequisite for 114, because of 113.

You must use graph data structure to solve this question. **Other solutions that do not use graph approach are not going to be evaluated.**

Question 2 (25 points):

In this question, you must design kind of SoS game. Assume that we have following matrix:

S S S S O

S O S O S

S S S S O

S S S S O

In this game you must convert all “O” characters to X which are surrounded by “S”.

Therefore, your output must be:

S S S S O

S X S O S

S S S S O

S S S S O

Sample run:

```
4 // enter number of rows
5 // enter number of columns
S S S S O // enter matrix
S O S O S
S S S S O
S S S S O

S S S S O
S X S O S
S S S S O
S S S S O
```

You must use graph data structure to solve this question. **Other solutions that do not use graph are not going to be evaluated.**

WHAT TO HAND IN

- You must upload your code into VPL on LMS for each question. **If you do not upload your code into VPL on LMS, your homework are not going to be evaluated.**
- The Java sources must be WELL DOCUMENTED as comments, as part of your grade is going to be based on the level of your comments.
- You must upload maximum-3 pages PDF report document that explains your own answers for programming task in a clearly readable PA report format (refer to PA REPORT FORMAT section).

PA REPORT FORMAT

A programming assignment report is a self-description of a programming assignment and your solution. The report must not be hand-written. You may use a word processor or the on-line editor of your choice and prepare as a PDF document. The report must be grammatically correct and use complete English sentences. Each report should include the following sections, in the order given:

Information (%2.5): This section includes your ID, name, section, assignment number information properly.

Problem Statement and Code Design (%15): Include a brief summary of the problem and/or your sub-tasks to be completed in this assignment. You should show your modular design rationale by creating a structure chart that indicates your top-down, stepwise refinement of the problem solution. You may create the structure chart using available graphical tools like MS PowerPoint, SmartDraw etc.

Implementation, Functionality, Performance Comparison (%20): Since you have modular source code, you should describe each sub-module (program) in this section. Each sub-module should include names and types of any input/output parameters as well as the pseudocode algorithm that used for completing its task. By this way, you give meaning to each chart boxes from the previous section. Also, you should add your performance comparison, part II, here.

Testing (%7.5): You should provide a tester class that is able to identify key test points of your program. This class should be able to generate additional (apart from the given sample input/output) test data for the purpose of being clear on what aspects of the solution are being tested with each set. This section should also include a description of any program *bugs* that is, tests which has incorrect results. You should write these to describe your tests, summarize your results, and argue that they cover all types of program behavior.

Final Assessments (%5): In this final section, you should briefly answer the following questions:

- What were the trouble points in completing this assignment?
- Which parts were the most challenging for you?
- What did you like about the assignment? What did you learn from it?

GRADING:

- Codes (%50: %25 for Q1 and %25 for Q2)
 - Available test cases evaluation on VPL: %15
 - Hidden test cases evaluation: %15
 - Approach to the problem: %20
- Report (%50: %25 for Q1 and %25 for Q2)
 - Information: %2.5
 - Problem Statement and Code design: %15
 - Implementation, Functionality, Performance Comparison: %20
 - Testing: %7.5
 - Final Assessments: %5

IMPORTANT

IMPORTANT NOTES: Do not start your homework before reading these notes!!!

1. **This assignment is due by 23:55 on Friday, April 11th.**
2. You must upload your homework to Moodle before the deadline. No hardcopy submission is needed. You must upload your code files to VPL and you must upload your report to the submission reserved for the report in pdf format.
3. The standard rules about late homework submissions apply (**20 points will be deducted for each late day**). Please see the course syllabus for further discussion of the late homework policy as well as academic integrity.
4. You ARE NOT ALLOWED to modify the given method names. However, if necessary, you may define additional data members and member functions.
5. Your classes' name MUST BE as shown in the homework description.
6. You must implement your work individually. The works that have high similarity are going to be graded as 0.
7. The submissions that do not obey these rules are not going to be graded.

8. To increase the efficiency of the grading process as well as the readability of your code, you have to follow the following instructions about the format and general layout of your program.

8. Do not forget to write down your id, name, section, assignment number or any other information relevant to your program in the beginning of your Java files. Example:

```
//-----  
// Title: Scheduler tester class  
// Author: Name/Surname  
// ID: 2100000000  
// Section: 1  
// Assignment: 1  
// Description: This class tests the ...  
//-----
```

9. Since your codes will be checked without your observation, you should report everything about your implementation. Add detailed comments to your classes, functions, declarations etc. Make sure that you explain each function in the beginning of your function structure. Example:

```
void setVariable(char varName, int varValue)  
//-----  
// Summary: Assigns a value to the variable whose  
// name is given.  
// Precondition: varName is a char and varValue is an  
// integer  
// Postcondition: The value of the variable is set.  
//-----  
{  
    // Body of the function  
}
```

10. Indentation, indentation, indentation...

11. This homework will be graded by your TAs, Deniz Merve Gündüz, and Mehmet Bahadır Aşkın. Thus, you may ask them your homework related questions through office hours. You are also welcome to ask your course instructor İsmail Bora Çelikkale for help.