# Program Design (II)

Homework #2
Due: 2022/4/22 13:00 pm (CST)

NOTE: Please upload your C program to Domjudge before the due date and time.

In this program, you will need to build a course roadmap using an interconnected linked list. Users can check what is the prerequisite course and what is the next course of the current course. For example, the prerequisite course for "Data Structure" is "Program Design", and the next course for "Data Structure" is "Algorithms". Therefore, you need to design and build an interconnected linked list using structure. The name of the structure should be `course` and it will have three members: `char name[NAME_LENGTH]`, `struct course *next`, and `struct course *prior`. The `name` represents the course name. The `next` and `prior` are the structure pointers pointed to another `struct course.` The `next` points to the next course and the `prior` points to the prerequisite course. The figure below illustrates an example of how `struct course` connects with each other.
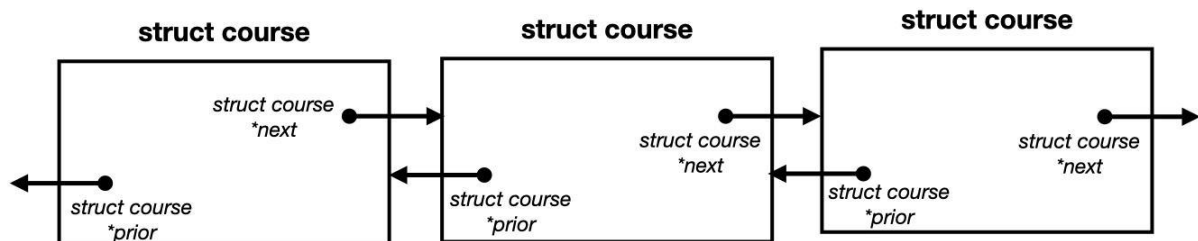
To let the program know how to build the interconnected linked list, users need to input the following format first via terminal and the table below describes the meaning of each input component:

```
<number of courses>
<course_id> <next> <piror> <course name>
...
```

Input component	Meaning
`<number of courses>`	The number of courses will be described in the following input. For example, if the value of `<number of courses>` is 3, there will be three separate lines of `<course_id> <next> <piror> <course name>`
`<course_id>`	The identity number of the course. The ID numbers are also used by `<next>` and `<prior>` to indicate which courses to connect with.
`<next>`	The value of `<next>` indicates the course with `<course_id>` that this course will connect to via the structure pointer `next`. For example, if the value of `<next>` is 1, the next course that this course connects to will be the course with `course_id` 1.
`<prior>`	The value of `<prior>` indicates the course with `<course_id>` that this course will connect to via the structure pointer `prior`. For example, if the value of `<prior>` is 0, the prerequisite course that this course connects to will be the course with `course_id` 0.
`<course name>`	`<course name>` indicates the name of this course. There will be no space in the course name. For example, the "Program Design" will become ProgramDesign during loading course information. For simplicity, the name of the course will be no greater than 49 characters.

Using the definition of each input component and design of `struct course`, the figure below illustrates an example input of three courses and how the linked list of struct courses would look like:

```
3                       <num of course>
0 1 -1 Vector           <course_id> <next> <prior> <course name>
1 2 0 Algebra           ...
2 0 2 GraphTheory
```
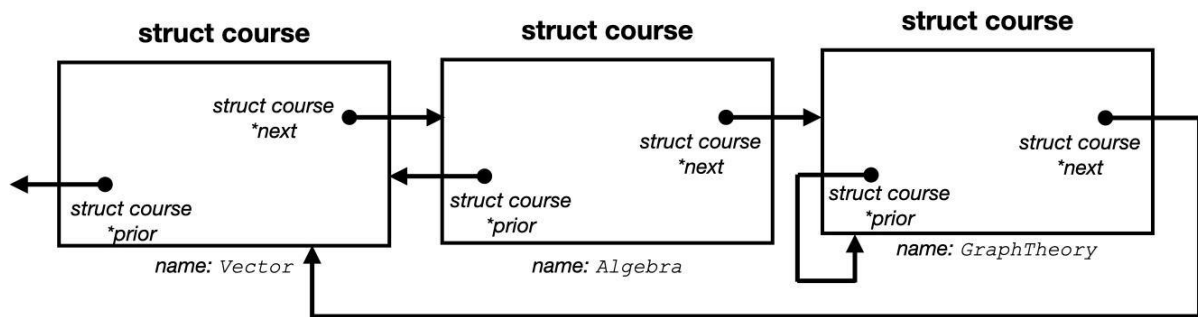

You can see that the `prior` and `next` pointers of a course can connect to the course itself or the course that is far from it. Although the ID number of the courses in this example is in ascending order (i.e., 0, 1, 2), you can see from the other example input that the ID numbers can be arbitrary. Lastly, here are the requirements of the program that you need to follow. If you fail to do so, you will get zero points for this homework.

1. You need to use dynamic memory allocation to allocate memory space for `struct course`

2. After the linked list of `struct course` is built, each `strcut course` can only be accessed by other the pointers of other `strcut course` in the linked list.

3. You cannot allocate a large block of memory for `struct course` in advance without considering users' inputs.

The table below lists other examples of input and outputs. The underscored numbers are the input from users.

Example Inputs and Outputs
<pre>4
0 1 -1 ComputerProgramming
1 2 0 Algorithms
2 3 1 Calculus
3 4 2 OperatingSystems
Current Course: ComputerProgramming
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Algorithms
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Calculus
[n] Next course. [p] Prior course [q] Quit:n
Current Course: OperatingSystems
[n] Next course. [p] Prior course [q] Quit:n
There is no next course.
Current Course: OperatingSystems
[n] Next course. [p] Prior course [q] Quit:p
Current Course: Calculus
[n] Next course. [p] Prior course [q] Quit:p
Current Course: Algorithms
[n] Next course. [p] Prior course [q] Quit:p
Current Course: ComputerProgramming
[n] Next course. [p] Prior course [q] Quit:p
There is no previous course.
Current Course: ComputerProgramming
[n] Next course. [p] Prior course [q] Quit:q</pre>

```
3
0 1 -1 Vector
1 2 0 Algebra
2 0 2 GraphTheory
Current Course: Vector
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Algebra
[n] Next course. [p] Prior course [q] Quit:n
Current Course: GraphTheory
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Vector
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Algebra
[n] Next course. [p] Prior course [q] Quit:n
Current Course: GraphTheory
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Vector
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Algebra
[n] Next course. [p] Prior course [q] Quit:n
Current Course: GraphTheory
[n] Next course. [p] Prior course [q] Quit:p
Current Course: GraphTheory
[n] Next course. [p] Prior course [q] Quit:p
Current Course: GraphTheory
[n] Next course. [p] Prior course [q] Quit:p
Current Course: GraphTheory
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Vector
[n] Next course. [p] Prior course [q] Quit:q
```

```
4
0 100 -1 C
100 101 0 C++
101 99 100 Java
99 100 101 Python
Current Course: C
[n] Next course. [p] Prior course [q] Quit:n
Current Course: C++
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Java
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Python
[n] Next course. [p] Prior course [q] Quit:n
Current Course: C++
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Java
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Python
[n] Next course. [p] Prior course [q] Quit:n
Current Course: C++
[n] Next course. [p] Prior course [q] Quit:p
Current Course: C
[n] Next course. [p] Prior course [q] Quit:p
There is no previous course.
Current Course: C
[n] Next course. [p] Prior course [q] Quit:p
There is no previous course.
Current Course: C
[n] Next course. [p] Prior course [q] Quit:n
Current Course: C++
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Java
[n] Next course. [p] Prior course [q] Quit:n
Current Course: Python
[n] Next course. [p] Prior course [q] Quit:n
Current Course: C++
[n] Next course. [p] Prior course [q] Quit:p
Current Course: C
[n] Next course. [p] Prior course [q] Quit:q
```