Tower of Hanoi Assignment

**Class** : Advanced C Programming

**Submission Date** : 2024/04/05

**Student ID** : 2022742021

**Affiliation** : Kwangwoon University

**Full name** : Juho Kim

**Major** : Software Engineering

# Introduction

- Brief overview of the Tower of Hanoi assignment

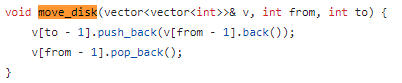
This is a text-based application of solving Tower of Hanoi puzzle in a console window by using C++. The objective of Hanoi Tower is to move all the disks to another rod stacked in descending order (largest to smallest from bottom to top). The game was developed applying the knowledge of vectors, error handling methods, Object Oriented Programming (OOP) and other concepts from advanced c programming.

- Purpose of the report

# Requirements Fulfillment

Following are the assignment requirements along with its corresponding code implementation

◼ Only one disk can be moved at a time

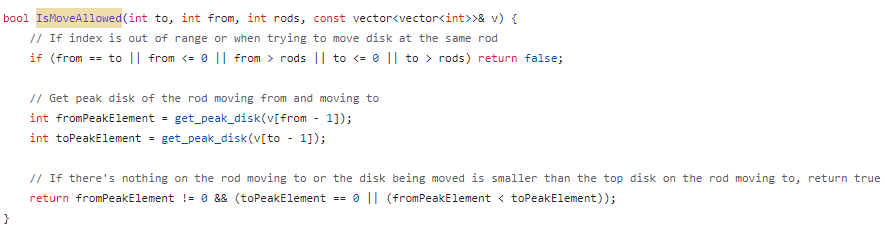


The move function enables only one disks to be moved to another rod.

◼ Each move consists of taking the upper disk from one of the stack and placing it on top of another stack or on an empty rod.

◼ No larger disk may be placed on top of a smaller disk.

In order to meet the 2 requirements mentioned above, following function was created to check if movement is allowed:



If the user tries to place the disks on the same rod or any other rod out of the range (that is a nonexistent rod), false is returned to indicate that the movement is not allowed.



To check the top disk of the rod moving from and to, the following code was implemented



If there’s nothing on the rod moving to or the disk being moved is smaller than the top disk on the rod moving to, true is returned to indicate that the movement is allowed. However, if user tries to move from a rod which doesn’t have disks, it will return false.



**All disks are stacked on the first rod in decreasing order of sizes**

**Number of moves (starts from 1)**

**Show allowed indices**

# Flow chart of play\_game()

The general flow chart of hanoi tower has been specified based on the assignment code as follows :

|  |  |
| --- | --- |
| **General flowchart** | **Flowchart based on the assignment code** |
|  |  |

- Explanation of how each requirement was fulfilled

- For each requirement:

- Description

- Implementation approach

- Code snippets (if applicable)

Additional Features

- Description of any additional features implemented

- Explanation of how each feature was implemented

- Code snippets (if applicable)

Code Correctness Demonstration

- Control flow correctness as specified in the flow chart

- Handling of valid and erroneous inputs

- Determination of puzzle solved status

- Screenshots for each example case

Conclusion

- Key challenges tackled

- Limitations addressed

- Future improvements