Hov Mining Simulator

Hov Mining Simulator is a classic console game developed using C programming language. In this game the player is role-playing as manager of a gold mining site. The mining site can have up to 999 caves, which indexed by a number between 1 and 999. The game implements Binary Search Tree (BST) data structure to store the cave data using its index as the key. Based on the input order of the record, each cave position in the tree representation will be different. As the consequences, each cave will have their own depth value. The cave in the surface (root), has depth value of 1, while its child has depth value of 2 and so on. Each cave stores its total gold production.

Repetitive insert of the same cave index record will increase its total gold production. Please check on below illustration. For example, using the given data and inputted into the system by following this order:

* Insert cave 52, gold production 10
* Insert cave 31, gold production 15
* Insert cave 27, gold production 25
* Insert cave 40, gold production 11
* Insert cave 79, gold production 89
* Insert cave 65, gold production 4
* Insert cave 82, gold production 2
* Insert cave 94, gold production 51
* Insert cave 40, gold production 13

Therefore, the BST representation of the data is shown in **Figure 1**. Note that there are duplicate records for cave 40, therefore the later one will update the current gold production.

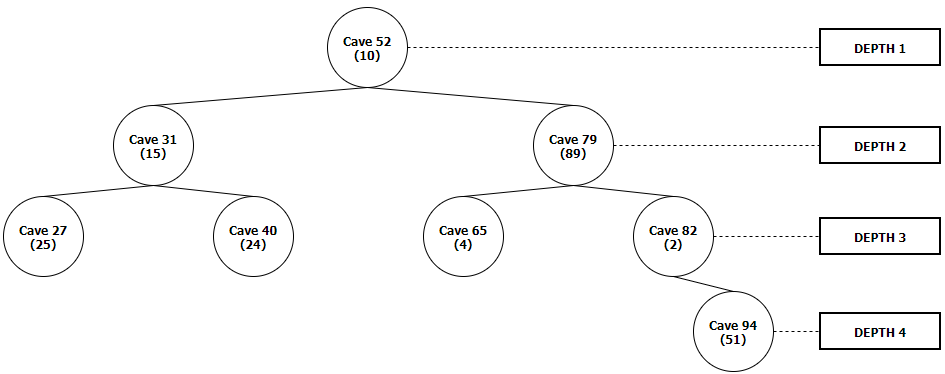


Figure . Binary Search Tree Representation of the Record

Your task is to re-create this game by following the given requirement. The game has four main menus as shown in **Figure 2,** they are: [1] Insert Mining Data, [2] Display All Cave Data, [3] Display Mining Reports, and [4] Exit. Each menu will be explained in each separated chapter. Note that each menu has different weight for the marking purpose.

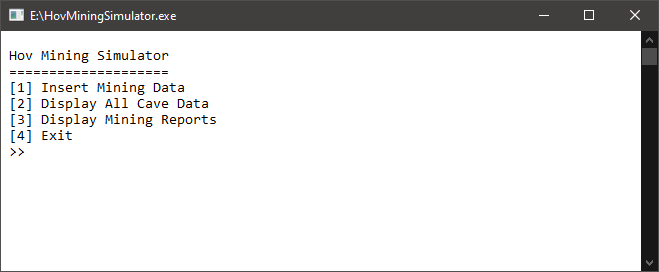


Figure . Main Menu of Hov Mining Simulator

1. Insert Mining Data (50%)

* This menu is used to insert mining record into the game memory
* User will input cave index (validate that the value must be between 1 and 999) and gold production (validate that the value must be between 1 and 100)
* Insert the record into the system and show the status message as shown in **Figure 3** and **Figure 4**.

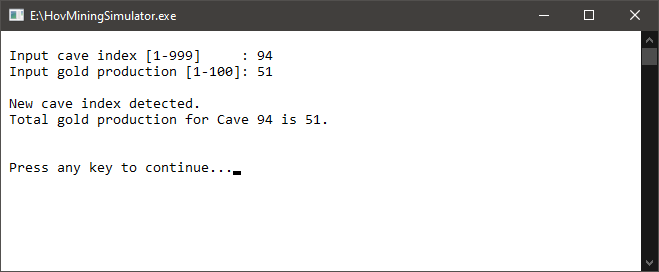


Figure . Menu Insert Mining Data when Inserting New Cave Record

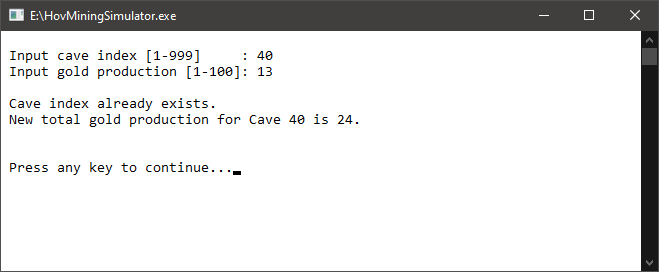


Figure . Menu Insert Mining Data when Inserting Existing Cave Record

1. Display All Cave Data (20%)

* This menu is used to display all existing cave record and its total gold production as shown in **Figure 5**. The result is sorted based on cave index in ascending order.
* Display error message if no record found as shown in **Figure 6**.

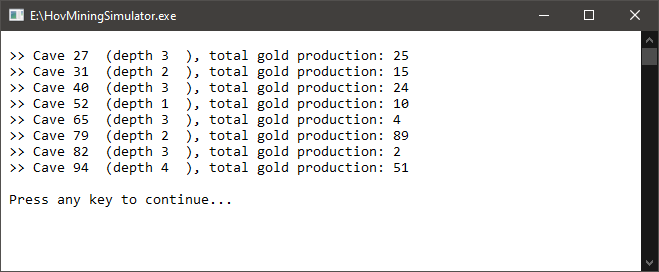


Figure . Menu Display All Cave Data – Non-Empty Record

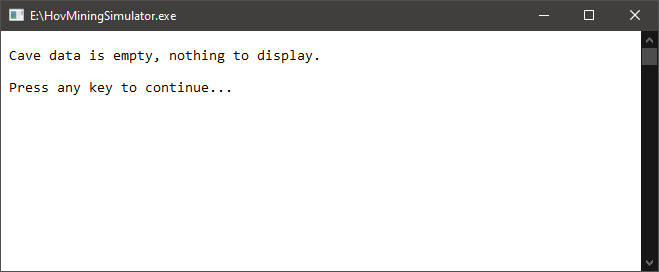


Figure . Menu Display All Cave Data –Empty Record

1. Display Mining Reports (20%)

* This menu is used to display total gold production for each level as shown in **Figure 7**. For example, in depth level 3 there are 4 caves: 27 (25 gold), 40 (24 gold), 65 (4 gold), and 82 (2 gold). Therefore, the gold production sum of depth level 3 can be calculated as follow: **.**
* Display the result from the highest to the lowest level of the cave.
* Display error message if no record found as shown in **Figure 8**.

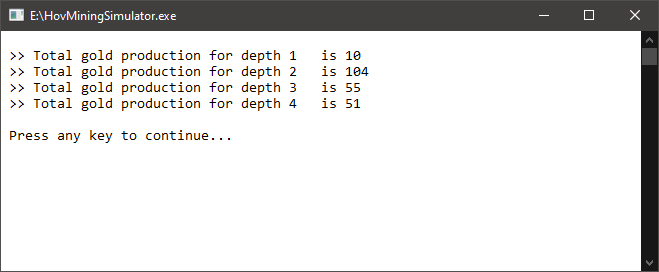


Figure . Menu Display Mining Reports - Non-Empty Record

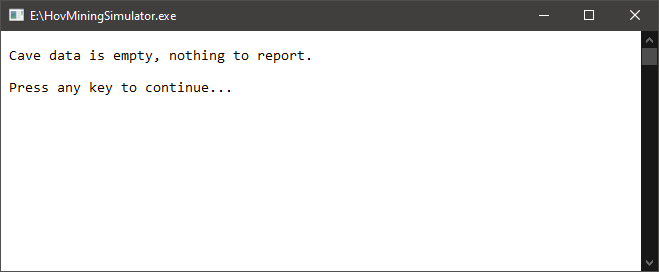


Figure . Menu Display Mining Reports - Empty Record

1. Exit (10%)

* This menu can be used to exit from the game.
* Before the program close, it will remove all data in the memory then show a thank you message as shown in **Figure 9**.

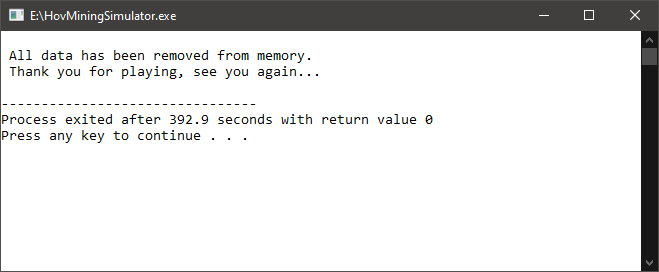


Figure . Menu Exit

– Wish you all the best! –

“Talk is cheap. Show me the code.” ― **Linus Torvalds**