|  |  |  |  |
| --- | --- | --- | --- |
|  | Bansilal Ramnath Agarwal Charitable Trust's  Vishwakarma Institute of Information Technology  **Department of**  **Artificial Intelligence and Data Science** | | |
| Student Name: Mohammad Faiz Nishat Parvej Saiyad | | | |
| Class: TY | Division: A | | Roll No:371034 |
| Semester: V | | Academic Year:2022-23 | |
| Subject Name & Code: Design and Analysis of Algorithms | | | |
| Title of Assignment: **Solve the following instance of the knapsack problem given the knapsack capacity in w=20 using greedy methods. Total no of item is 5.**  **Item   Weight   Profit**  **X1        3             10**  **X1        5             20**  **X1        5             21**  **X1        8             30**  **X1       4             16** | | | |
| Date of Performance: | | Date of Submission: | |

**Aim:**

**Solve the following instance of the knapsack problem given the knapsack capacity in w=20 using greedy methods. Total no of item is 5.**

**Item   Weight   Profit**

**X1        3             10**

**X1        5             20**

**X1        5             21**

**X1        8             30**

**X1       4             16**

**Problem Statement:**

To solve Knapsack problem using Greedy Algorithmic Approach

**Software Requirements:**

Text Editor: VSCode, Neovim, etc

Environment: Python 3.10

Terminal Emulator

**Background Information:**

**Greedy Algorithm:**

Greedy is an algorithmic paradigm that builds up a solution piece by piece, always choosing the next piece that offers the most obvious and immediate benefit. So the problems where choosing locally optimal also leads to global solution are the best fit for Greedy.

For example consider the Fractional Knapsack Problem. The local optimal strategy is to choose the item that has maximum value vs weight ratio. This strategy also leads to a globally optimal solution because we are allowed to take fractions of an item.

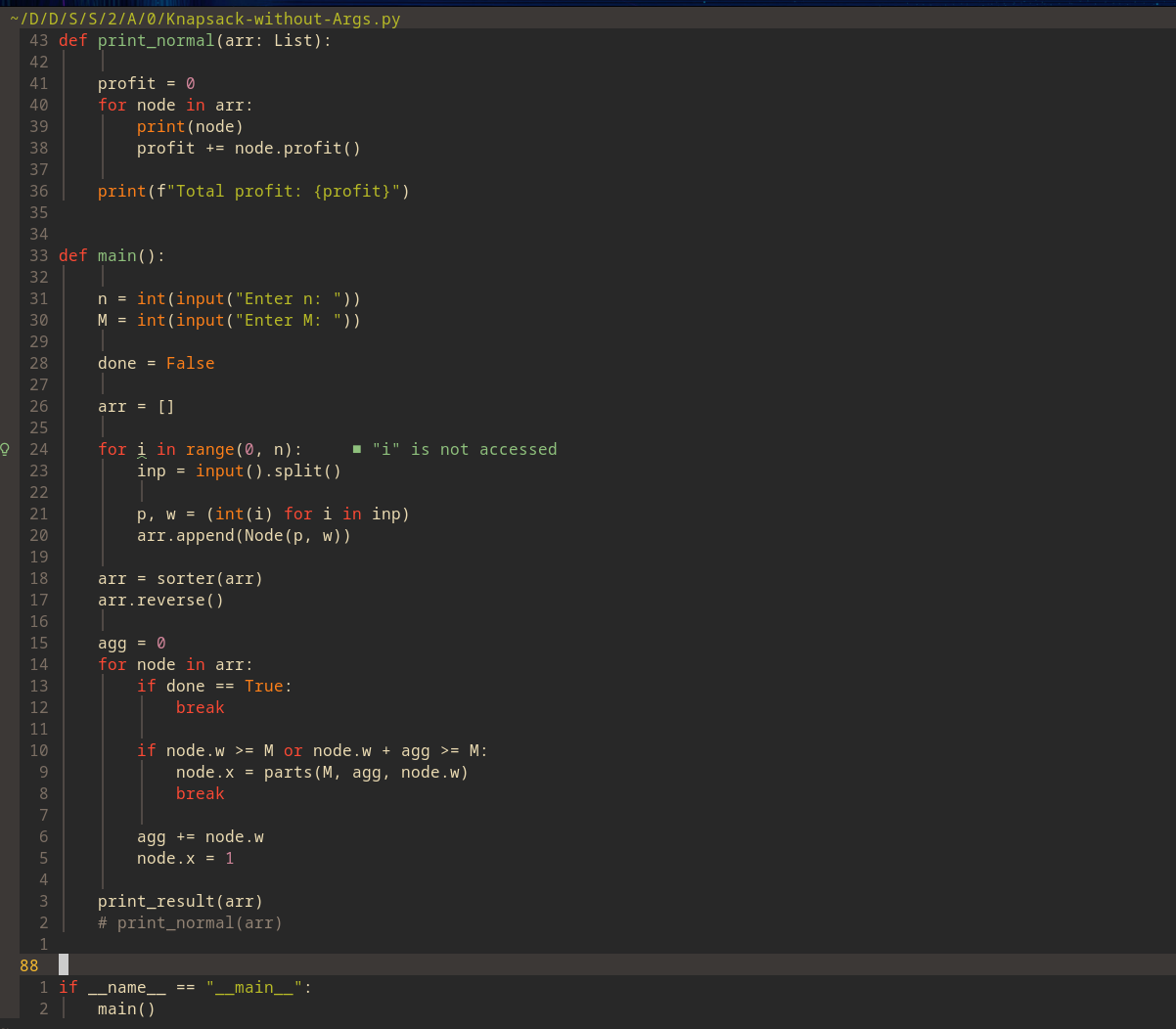
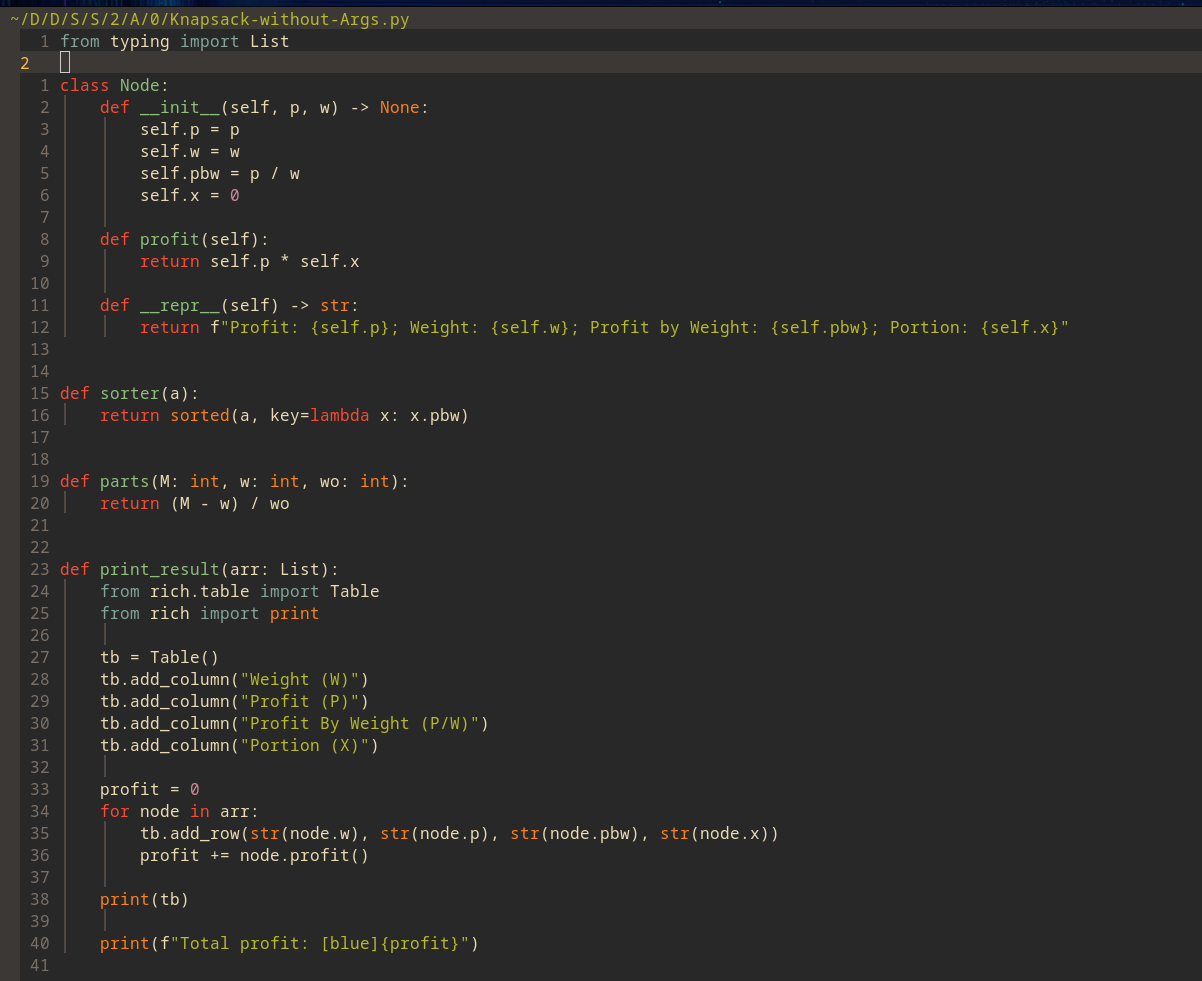
**Fractional Knapsack Problem:**

Given the weights and values of N items, in the form of {value, weight} put these items in a knapsack of capacity W to get the maximum total value in the knapsack. In Fractional Knapsack, we can break items for maximizing the total value of the knapsack

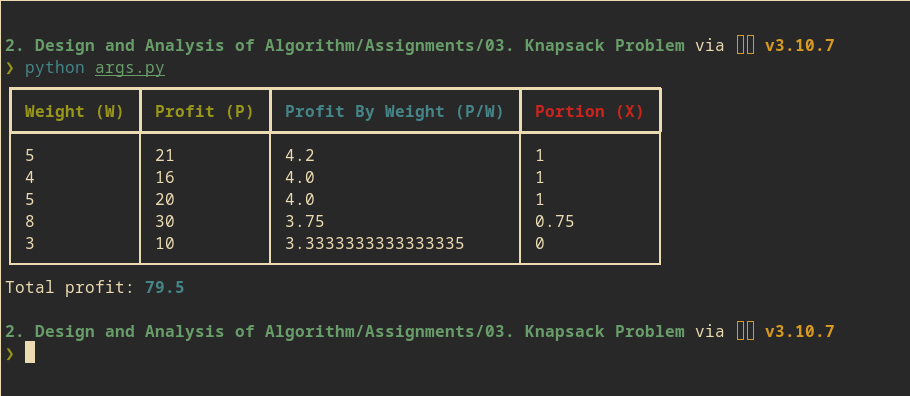
Follow the given steps to solve the problem using the above approach:

* Calculate the ratio(value/weight) for each item.
* Sort all the items in decreasing order of the ratio.
* Initialize res =0, curr\_cap = given\_cap.
* Do the following for every item “i” in the sorted order:
* If the weight of the current item is less than or equal to the remaining capacity then add the value of that item into the result
* Else add the current item as much as we can and break out of the loop.
* Return res.

**Code:**



**Output:**



**Conclusion:**

Implemented Fractional Knapsack problem using the Greedy Method.