**READ ME**

**Python Language** is used to code the solution for Rack-Switch Problem

Give input in the beginning of the program as asked by it

This code creates: -

1) 3 variables for size of Rack and calculate volume of it

2) Dictionary containing switches as key and all the other information as value of key in the form of array

3) From this dictionary a new dictionary is created specially for working on quantity

4) A set of height is created containing distinct height from all switches

Two Different functions are created **knapsack** and **knapsack\_height**

**Knapsack\_height** function is called first in the program where the base condition is set as when the remaining height of rack is less than minimum height of any available switch the program gets terminated

In **knapsack\_height** function for loop is applied on height set so that it selects each distinct height as maximum height of level and best cases can be obtained from multiple types of level which can best fit in Rack so that minimum rack volume is wasted and the score gained can be maximum

**knapsack\_height function** calls **knapsack** function in which switches are selected if the condition is true and an **algorithm is used by which a level can be filled by all the possible cases**, knapsack function is recursively called to fulfil the algorithm and give the result in which maximum space is covered in a level

From all the cases **result with best score** is selected as a level and is **passed to knapsack\_height function**

knapsack\_height recursively calls itself to fill Rack with all possible levels

All the cases are stored in an array named **guess**

From guess, the result with maximum score is selected

This case contains 3 different variables

1) Total score gained

2) Quantity of switches left after filling complete rack

3) A key named levelwise in which quantity used in each level is stored is appended and stored as the value of key

By using levelwise information is gathered about each level and is displayed

Volume of Rack used is calculated by using the total quantity of switches used to fill rack

**Final score obtained and Quantity of switches left is displayed at the end**

For more detailed understanding of the code comments are added in code which describes each part of code and all the variables of the code