

PACK 'EM ON

Mr. Jack wants to plan an excursion with his family this weekend. He wants to pack as many useful items as possible, but is restricted with the amount of space left in his car, which is fixed. Each useful item is available for a specific cost. So, he tries to minimize the total amount spent in buying these items. Help Jack to choose the number of each type of useful item so that he utilizes the maximum space of his car, and incurs the minimum possible cost.

You are required to develop a game which is a real time implementation of the above given problem. Develop the game based on the given conditions and directions.

This game aims at testing the basic understanding of the problem, algorithm development and practical approach to a problem and also optimising the solution.

Problem statement:

- A player 'X' starts the game by clicking on the start button which initiates the timer.
- One packing box 'PB' with a random size of S is generated.
- S can take values only among the values $\{100,200,300,400,500,\dots,1000\}$ (expressed in cubic meters)
- Five types of small boxes of various sizes S_i and corresponding costs C_i are also generated ($1 \leq i \leq 5$).
- The maximum number of each small box that can be used is N_i .
- Each of S_i is a multiple of 5 (expressed in cubic meters) with a range from 5 to 25.
- N_i is generated randomly and represents the maximum number of small boxes of size S_i .
- The task assigned to the player is to assign a number $n_i < N_i$, against corresponding small box such that the total size occupied by all the small boxes is optimum and fits properly inside the packing box.
- The user inputs are the number of each small boxes that are considered
- Once the player finishes with the game, an algorithm must be developed to find the optimum solution with the least possible cost and maximal space utilization. The player wins if his solution is optimum.
- The total size of all the small boxes must be maximum possible and lesser than the size of the packing box, i.e. once all the boxes are arranged inside the packing box, there must not be any other small box that can be fitted inside the space left.
- The output textbox contains a brief summary of what the player has selected and also generates the end result of the game.

The contestants are required to assume the additional components required to design the user interface for the game, apart from the given below. The minimum components required in the interface are:

- Timer
- Game start button
- Finish button
- Output text box
- Score card
- A table consisting of four columns containing the box id in the first column , box size in the second column and maximum number of that corresponding box in the third , and cost of each corresponding box in the last column.
- Text boxes against each small box to take user input.

PS: - All the valid assumptions are to be considered by the contestant while developing the game .
Valid assumptions also carry additional weightage.

For ex: Maximum Total volume of all the boxes should be greater than or equal to the size of the main packing box.

Marking Scheme:

1. Basic user interface along with required components : 10%
2. Algorithm development to randomly generate packing box with valid size : 10%
3. Algorithm development to randomly generate five types of small boxes with valid maximum number of each box , valid size of each box, and cost of each box : 15%
4. Proper comments at appropriate places : 10%
5. Algorithm development to take user inputs and calculate the total size obtained and total cost of all the boxes : 10%
6. Algorithm development to generate the optimal solution : 15%
7. Algorithm to Compare the optimal solution with the user's solution and generating the output : 10%
8. Code with proper line spacing and indentation : 10%
9. Compiling and execution of the code : 10%