

# CS69201: Computing Lab-1

## Task-1

July 31, 2024

### ===== Instructions =====

1. In the case of user input assume only valid values will be passed as input.
2. You can use C or C++ as the programming language. **However, you are not allowed to use any STL libraries in C++**
3. Regarding Submission: Create a single C/CPP file. -> <rollno>\_Knapsack.c and submit it to Moodle. For example, if your roll number is 24CS60R15, then your file name should be 24CS60R15\_Knapsack.c.

=====

In the bustling city of Arteria, the annual Grand Art Auction is the event of the year, attracting collectors from all over the world. Among them is the mysterious and eccentric Mr. Fabiano, known for his unparalleled collection of priceless art. This year, however, Mr. Fabiano is not just attending the auction—he's planning an audacious heist to steal the most valuable pieces of art.

Mr. Fabiano's plan is meticulous. He has identified the exact pieces he wants to steal, each with a specific weight and value. However, his getaway vehicle has a strict weight limit, and he must ensure that he can maximize the value of the art he takes without exceeding this limit.

Now there are certain clauses associated in how he shall steal the art pieces. Mr Fabiano asks you - his partner-in-crime and analyst, to check how much reward he can get on stealing the art.

**Clause 1 :** To achieve this, he is willing to cut the pieces if necessary, taking only fractions of them to fit into his vehicle. Help Mr Fabiano execute his heist flawlessly by determining the best combination of art pieces he should take to maximize their total value.

**Clause 2:** Mr Fabiano thinks the idea of cutting the art pieces can arouse suspicion. He has thus decided to collect art pieces in full or reject them due to the limited weight limit. He insists on calculating the total value in the same concept as in Clause 1 but without cutting the art pieces. Help Mr. Fabiano execute the heist by determining the best combination of art pieces to maximize total value.

**Clause 3:** You find that Clause 2 is flawed as it is not providing you the maximum value. In order to get the maximum profit for the benefit of you and your partners, you try combinations of art pieces in a trial and error method and gather a subset of art collections which he can accommodate given his weight limit to maximize your total value. Compute the maximum value.

**Clause 4 :** Clause 3 is very slow to compute. On analyzing your method, you figure out that you don't store the value of some subsets of the collection against the given weights while considering the subset with maximum collection. Given that you store your data in a cloud server with security patches enough to make it unhackable, you store the precompute values of such subsets in a tabular format and get your objective done requiring less space and less time. Optimize your code and compute the output (Note Clause 3 and Clause 4 have the same result. Just that your code is optimized in Clause 4 given memory storage).

**[Create separate functions for each clause]**

Input:

The first line contains two integers,  $n$  and  $W$  — the number of art pieces and the maximum weight capacity of the getaway vehicle.

The next  $n$  lines each contain two integers,  $w_i$  and  $v_i$  — the weight and value of the  $i$ th art piece.

Output: Print a single number — the maximum total value of art pieces that Mr. Fabiano can fit into his vehicle.

**Testcase**

**Input**

**4 30  
25 100  
10 30  
12 35  
15 90**

**Output :**

**Clause 1 : 150  
Clause 2 : 120  
Clause 3 : 125  
Clause 4 : 125**