

Faculty of Engineering, University of Jaffna,

Department of Computer Engineering

EC4070: Data Structures and algorithms

Lab 04

Chapter 4: Hashing and set ADT

Duration: 3 Hours

Lecturer: Ms. Sujanthika M.

Instructions

- i. Submit the code files and screenshot of the outputs in a zipped folder by naming as 2022EAAA_Lab04(AAA – Your Registration Number)
- ii. Submit your zip file before the given deadline.
- iii. Any plagiarized work will be given 0 marks.

Question 1: Hashing with chaining

Main canteen of the university has n hungry students waiting in line. Each unique order i , is placed by a student at time t_i , and the order takes d_i units of time to process.

Now canteen incharge wanted to know the sequence in which the orders will be completed so that order which is completed first is served first which resulting in less chaos in canteen during the break hours but he doesn't know how to solve this problem, so he asks

you to solve this problem Given the information for all n orders, can you find and print the order in which all n students will receive their orders? If two or more orders are fulfilled at the exact same time t , sort them by ascending order number.

Input Format

The first line contains a single integer n , denoting the number of orders. Each of the n subsequent lines contains two space-separated integers describing the respective values of t_i and d_i for order.

Output Format

Print a single line of space-separated order numbers describing the sequence in which the students receive their orders. If two or more students receive their order at the same time, print the smallest order number first.

Sample Input

```
5
8 1
4 2
5 6
3 1
4 3
```

Sample Output

```
4 2 5 1 3
```

Question 2: Inventory Management System

You are tasked with creating an **Inventory Management System** for a store using **hash tables**. The system tracks items and their quantities and supports the following operations:

1. **Add Item:** Add a new item to the inventory with its quantity. If the item already exists, increase its quantity.
2. **Update Quantity:** Update the quantity of an existing item. If the item does not exist, return an appropriate error message.
3. **Check Item:** Check if an item is in stock. If it is, display its quantity. If not, display a message saying it is out of stock.
4. **Display Inventory:** Display all items in the inventory along with their quantities in alphabetical order.

Input Format

1. The first line contains an integer n ($1 \leq n \leq 1000$) — the number of operations.
2. The next n lines contain one of the following commands:
 - 1 ItemName Quantity (Add an item with the specified quantity).
 - 2 ItemName Quantity (Update the quantity of an existing item).
 - 3 ItemName (Check if an item is in stock).
 - 4 (Display the full inventory).

Output Format

- For command 3, display the item quantity or "ItemName is out of stock."
- For command 4, display all items in the inventory in the format:
ItemName: Quantity.

Sample Input

```
6
1 Apples 50
1 Bananas 30
3 Apples
2 Bananas 40
4
3 Oranges
```

Sample Output

```
50
Apples: 50
Bananas: 40
Oranges is out of stock.
```