**Task 1:**

Data structure is a way of organizing and storing data in a computer so that it can be accessed and modified efficiently. The main purpose of data structure is to store and retrieve data efficiently. With different data structures, we can optimize the performance. It helps in solving problems with less memory usage and faster execution

**Task 2:**

There are linear data structures and non-linear data structures.

1. Linear Data Structures

Array

ArrayList

LinkedList

Stack

Queue

1. Non Linear Data Structures

Tree

Binary Tree

Graph

Trie

Heap

1. Hash Based Data Structures

HashMap

HashSet

LinkedHashMap

**Task 3:**

Insert - Add an element

Delete - Remove an element

Search - Find an element

Access/Get - Retrieve an element without removing

Update - Modify an existing element

Traverse - Visit all elements

Sort - Arrange elements in order

Filter - Select elements based on condition

Merge - Combine two structures

Resize - Increase or decrease structure size

Clone - Duplicate the structure

Check Size - Get number of elements

Check Empty - Check if structure is empty

**Task 4:**

| Feature | Static Array | Dynamic Array |
| --- | --- | --- |
| Size | Fixed at the time of declaration | It can resize itself during runtime |
| Performance | Fast Access | It is slightly lower due to resizing |
| Memory | Efficient, no extra spaces | It will allocate extra memory upon resizing |
| Flexibility | Inflexible; size cannot change | Flexible; can accommodate changing data sizes |
| Limitations | Wastes memory if oversized, or may overflow | Overhead of resizing and copying data |

**Task 5:**

01100001

**Task 6:**

1. Primary Memory

RAM : Temporary memory, fast, volatile

ROM : Permanent memory, non-volatile

Cache : Very fast, stores frequently used data

Registers : Inside CPU, smallest and fastest

2. Secondary Memory

HDD : Magnetic storage

SSD : Faster than HDD

Optical Disk : CD, DVD

Flash Drive : USB pen drive

Memory Card : microSD

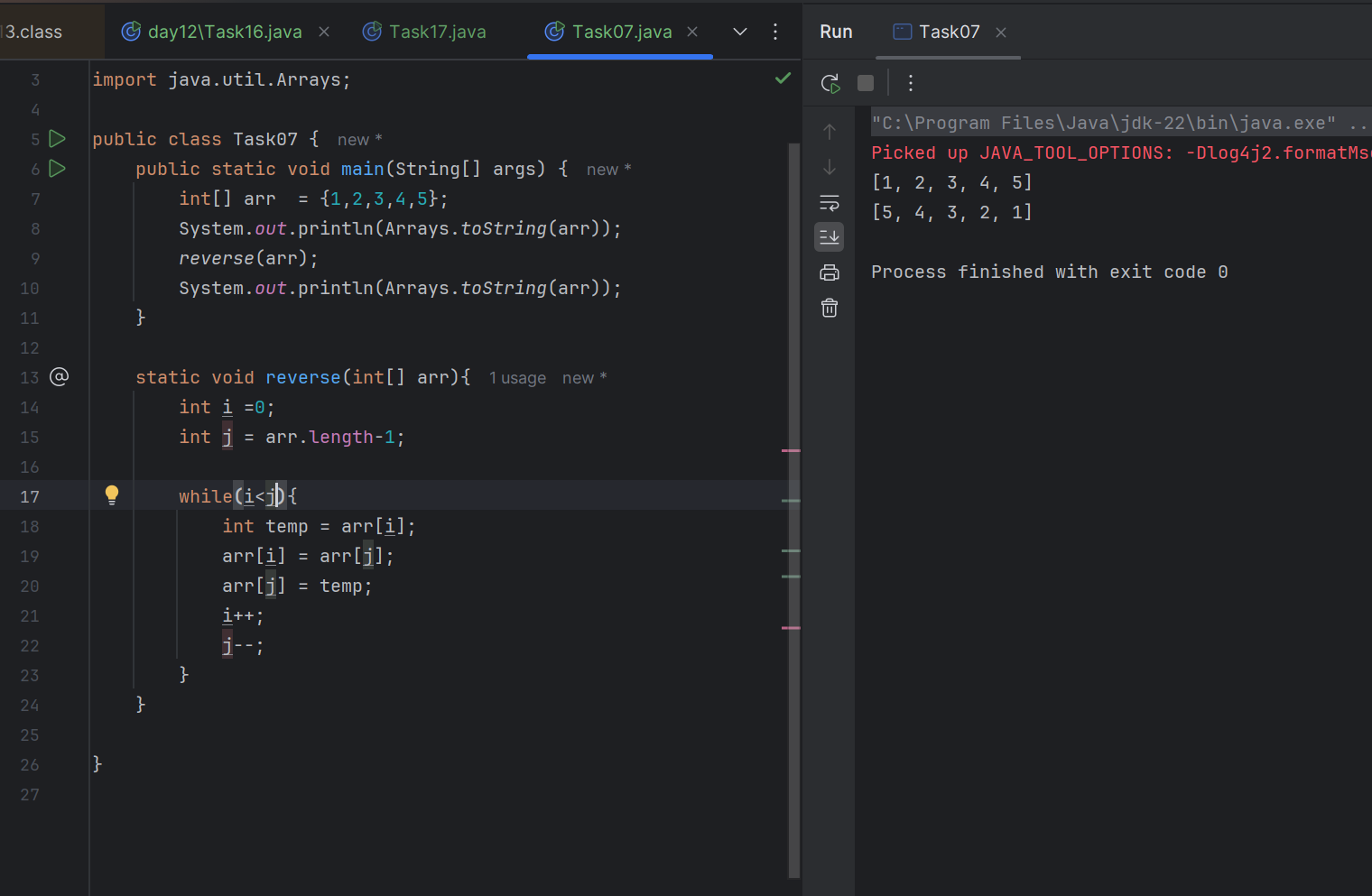
3. Tertiary Memory

Magnetic Tape : Used for backups in servers

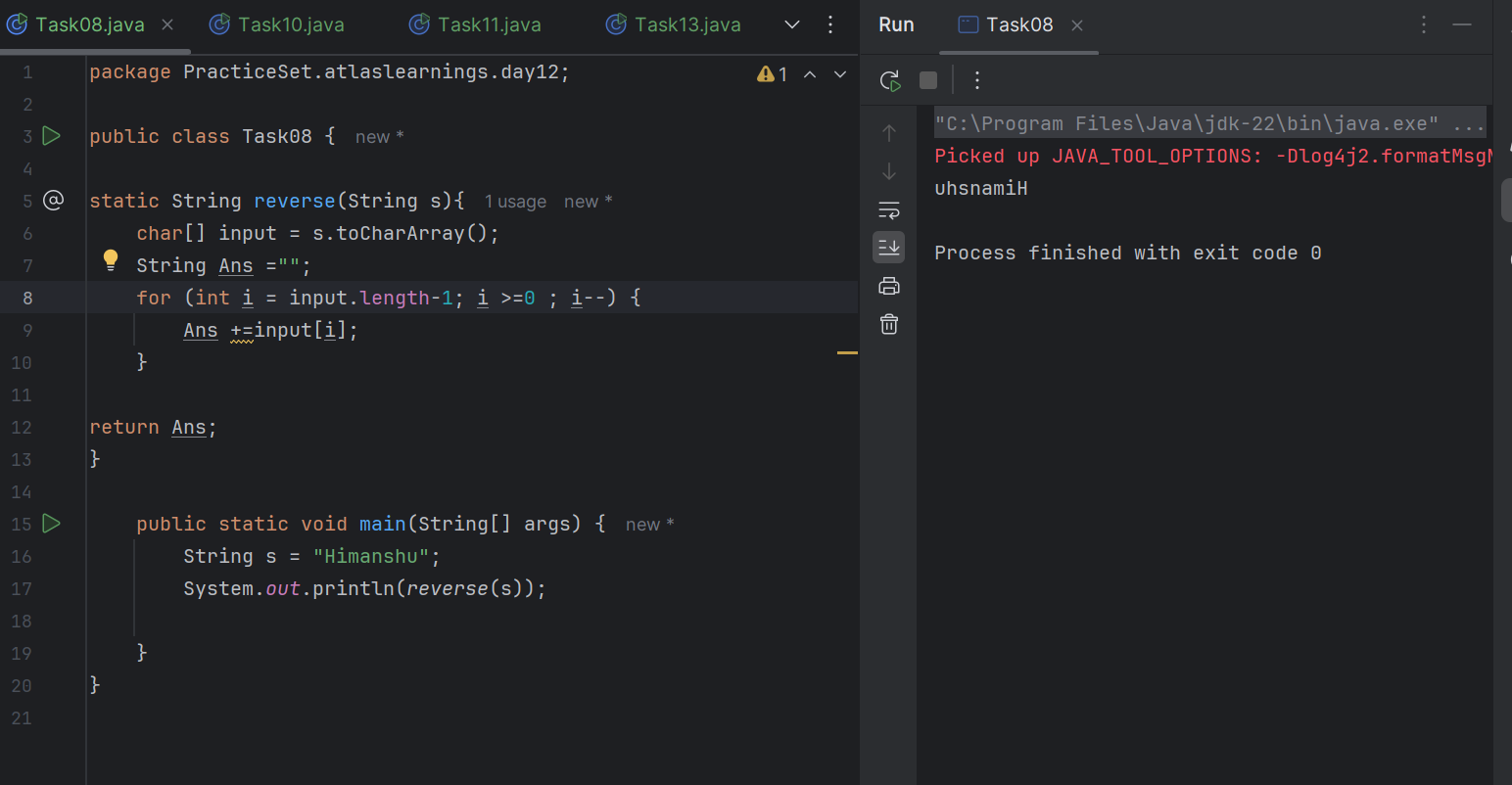
4. Virtual Memory

Uses hard disk as extra RAM when needed

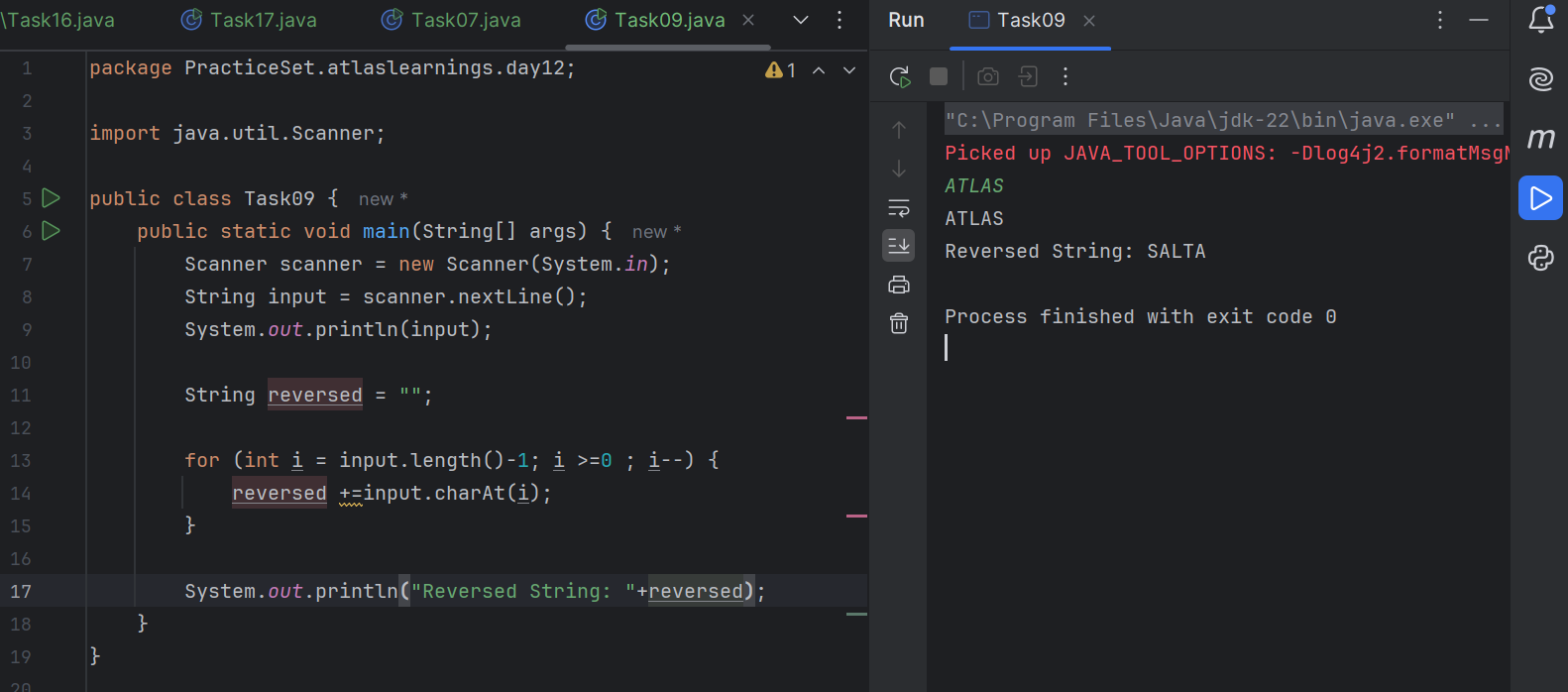
**Task 7:**



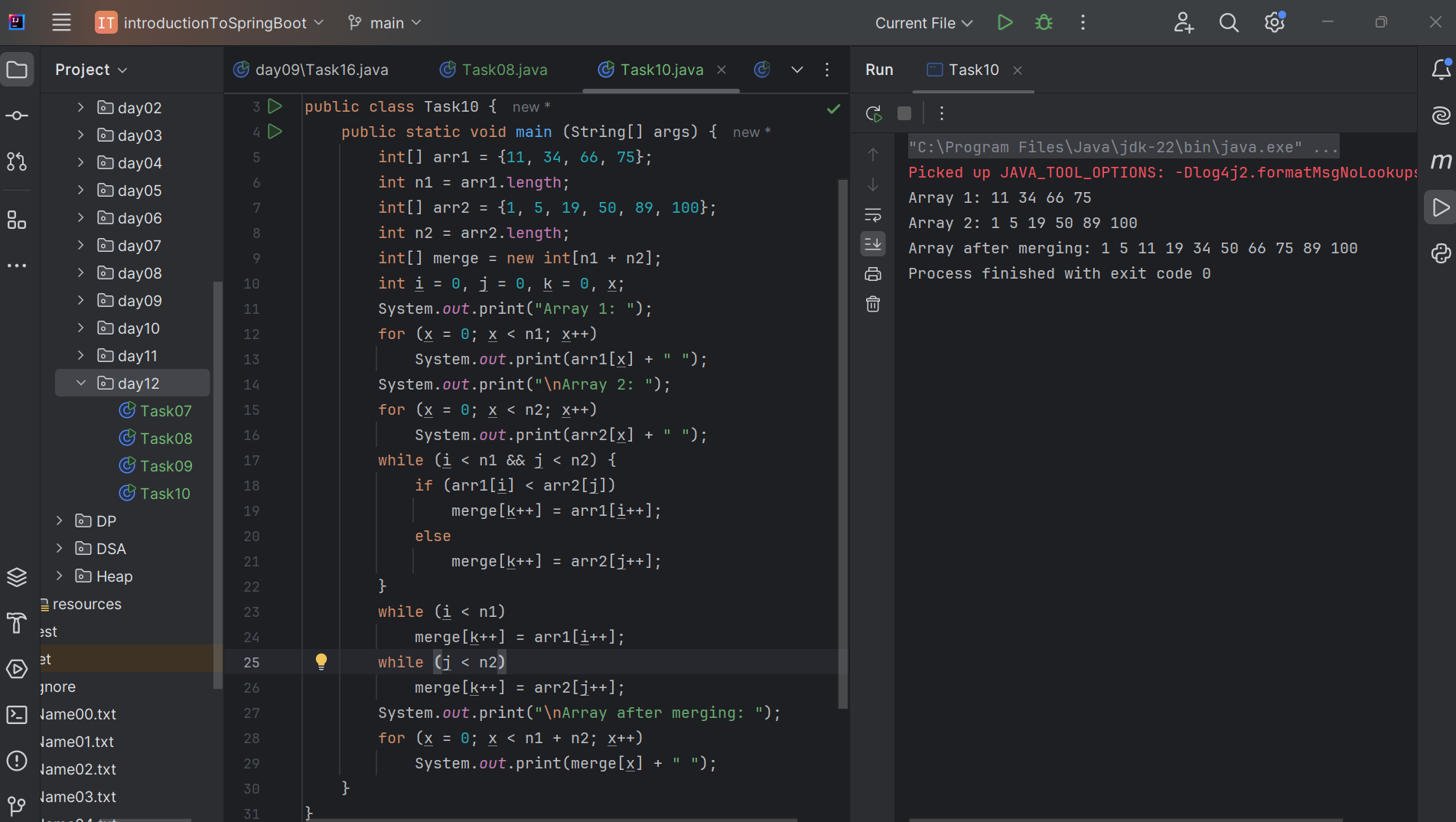
**Task 08:**



**Task 09:**



**Task 10:**

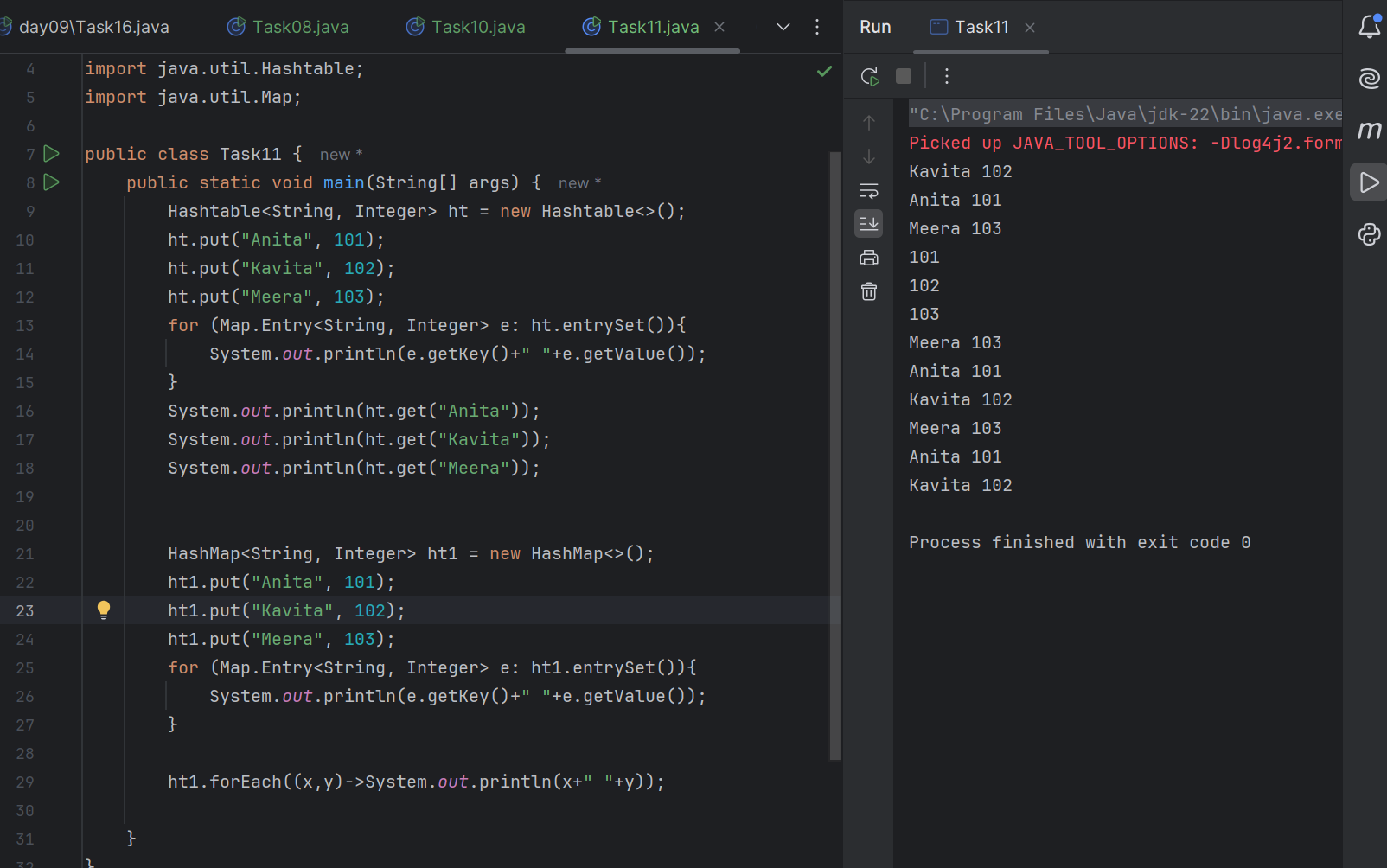


This code will merge 2 arrays in a sorted manner and if any elements are left from any of the arrays after merging, it will put remaining elements in the merged array.

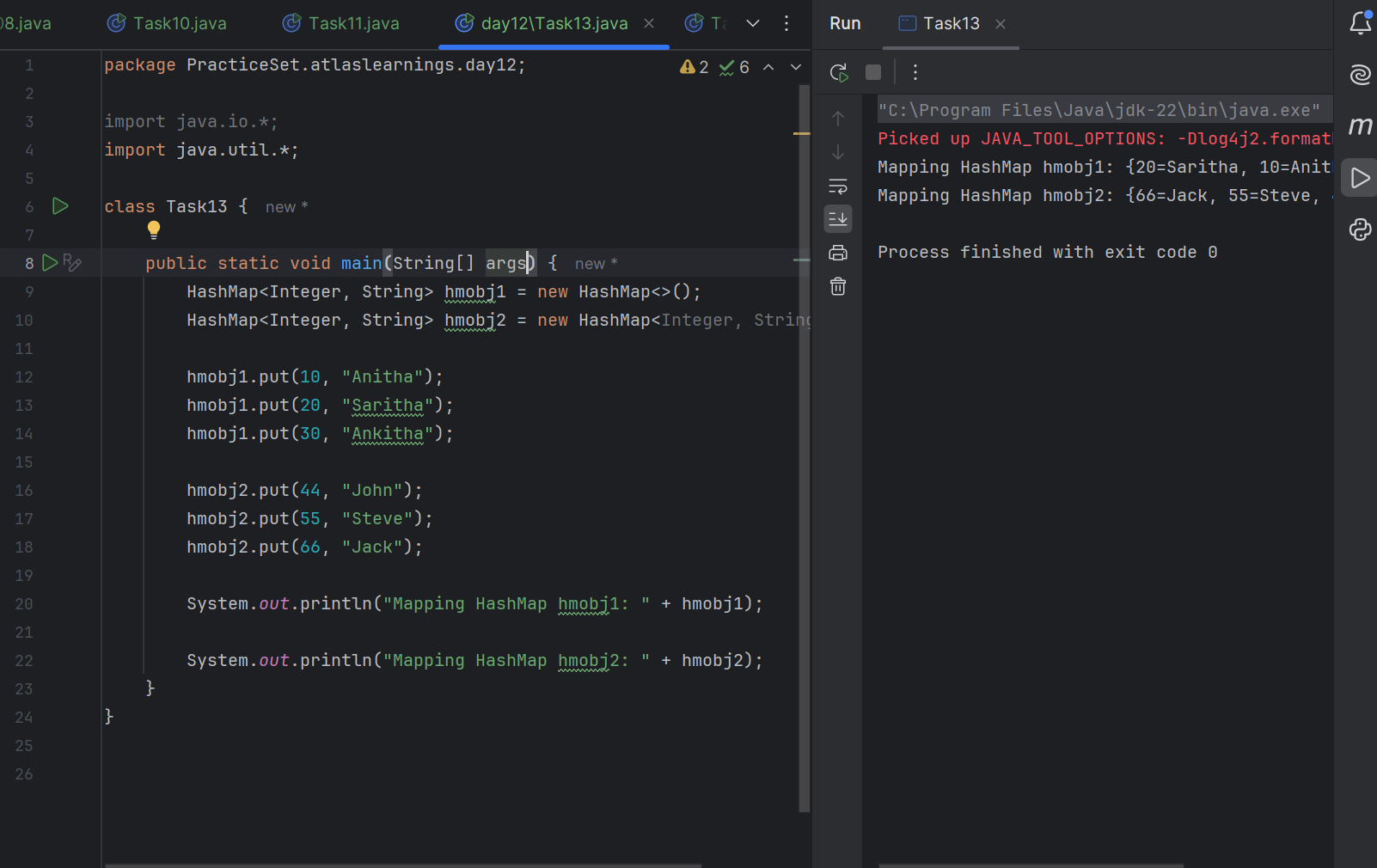
**Task 11:**

A hashtable is a data structure used to store key–value pairs in a way that allows very fast access to data. It uses a concept called hashing to convert a key into an index, where the value is stored. Fast lookup (almost constant time: O(1)). Easy to insert, delete, and search by key

**Task 12:**

****

**Task 13:**

****

**Task14:**

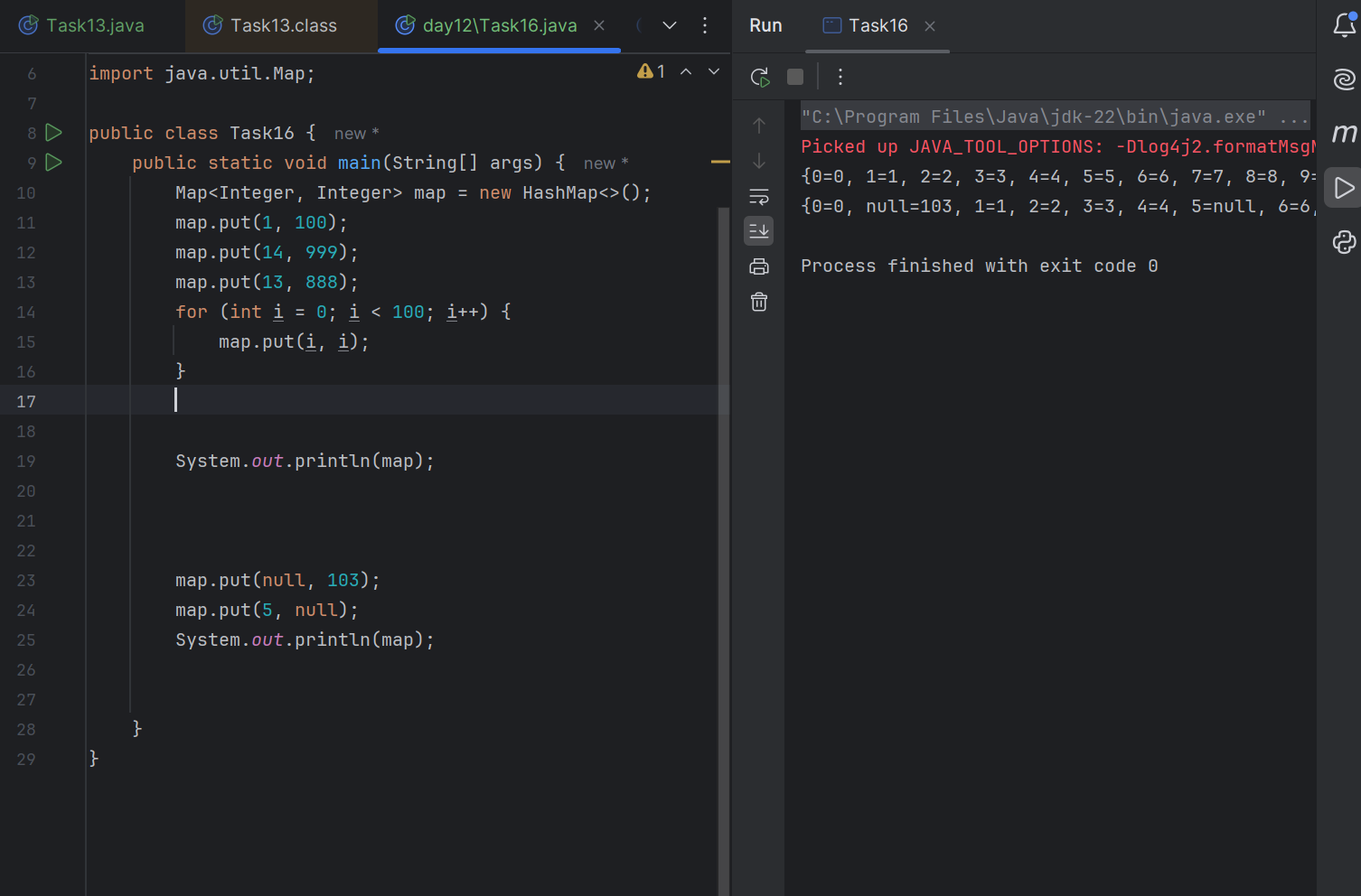
Differences between HashTable and HashMap

HashMap is not thread-safe whereas HashTable is thread-safe.

HashMap is faster as there is no synchronization overhead whereas HashTable is slower due to synchronization.

HashMap allows one null key and one null value, whereas HashTable does not allow null keys or values.

**Task 15:**

****

**Task 16:**

