Task 01 .What do you understand about data structures?

Data structures is used to store and handle the large amount data is used to handle large amount of data set .

Task 02 .What are the types of data structures you know .. list them out..

List of data structures

1. Array
2. Stacks
3. Queues
4. Linked list
5. Hash tables
6. Trees
7. Graph

Task 03 . What all operations can we do in Data structures?

Ans : Sorting , update , delete, search, add , merging , accessing

Task 04 . What are static and dynamic arrays? Explain or summarize key points in a table?

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Static Data Structure** | **Dynamic Data Structure** |
| **Memory allocation** | Memory is allocated at compile-time | Memory is allocated at run-time |
| **Size** | Size is fixed and cannot be modified | Size can be modified during runtime |
| **Memory utilization** | Memory utilization may be inefficient | Memory utilization is efficient as memory can be reused |
| **Access** | Access time is faster as it is fixed | Access time may be slower due to indexing and pointer usage |
| **Examples** | Arrays, Stacks, Queues, Trees (with fixed size) | Lists, Trees (with variable size), Hash tables |

What is the binary value of a?

97/2 = 1

48/2 = 0

24/2 = 0

12/2 = 0

6/2 = 0

3/2 =1

1

Asci of a – 97 🡪1100001

8 bit representation == 01100001

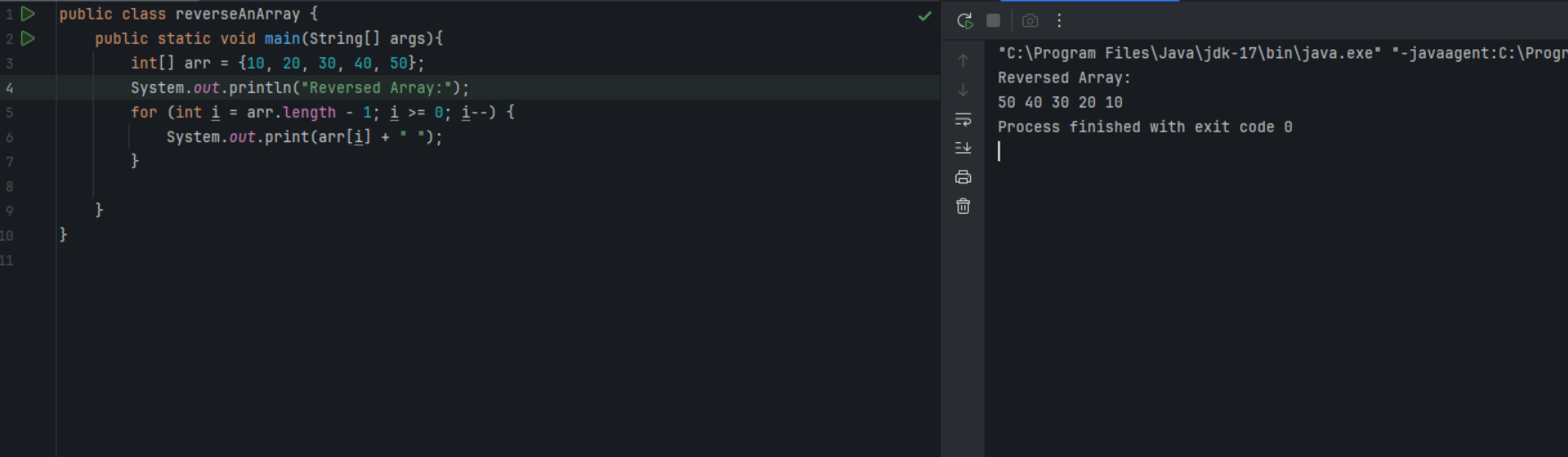
18 bit representation == 0000000001100001

Types of Computer memory with examples.. Explain ..

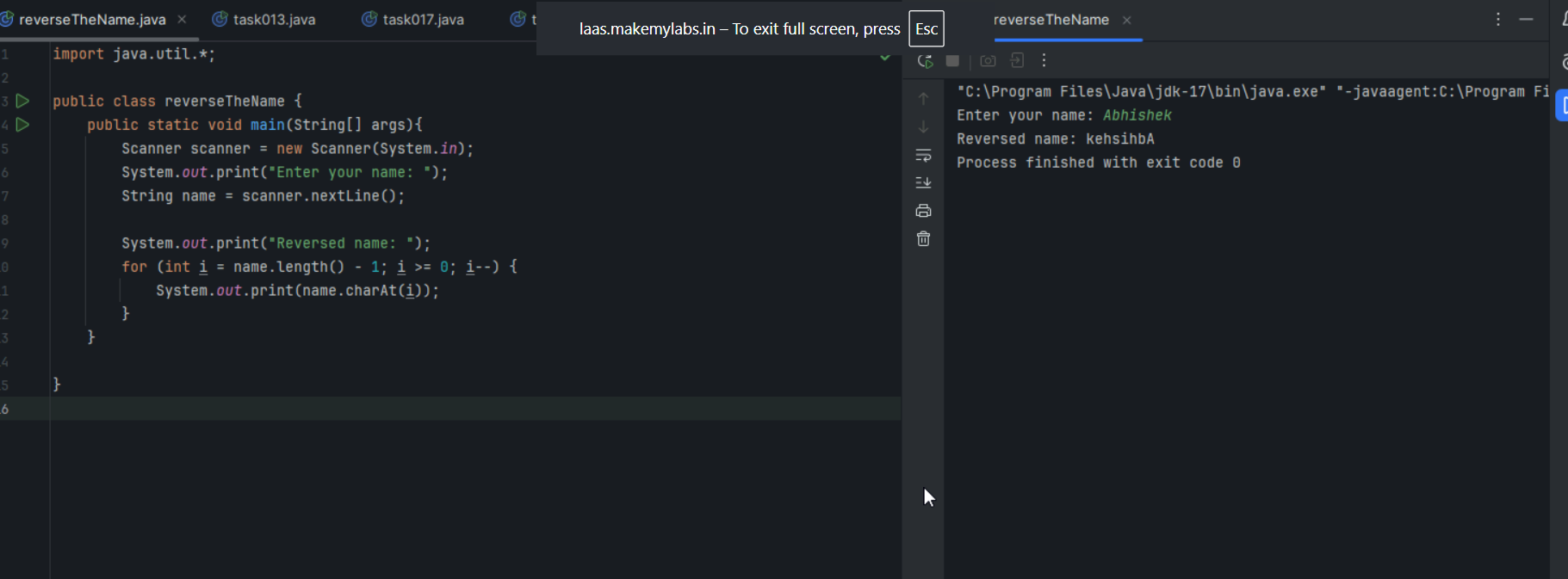
* **MROM(Masked ROM):**Hard-wired devices with a pre-programmed collection of data or instructions were the first ROMs. Masked ROMs are a type of low-cost ROM that works in this way.
* **PROM (Programmable Read Only Memory):** This read-only memory is modifiable once by the user. The user purchases a blank PROM and uses a [PROM](https://www.geeksforgeeks.org/prom-full-form/) program to put the required contents into the PROM. Its content can't be erased once written.
* **EPROM (Erasable Programmable Read Only Memory):** [EPROM](https://www.geeksforgeeks.org/eprom-full-form/) is an extension to PROM where you can erase the content of ROM by exposing it to Ultraviolet rays for nearly 40 minutes.
* **EEPROM (Electrically Erasable Programmable Read Only Memory):**Here the written contents can be erased electrically. You can delete and reprogram [EEPROM](https://www.geeksforgeeks.org/eeprom-full-form/) up to 10,000 times. Erasing and programming take very little time, i.e., nearly  4 -10 ms(milliseconds). Any area in an EEPROM can be wiped and programmed selectively.
* **S RAM (Static RAM):** [S RAM](https://www.geeksforgeeks.org/sram-full-form/) uses transistors and the circuits of this memory are capable of retaining their state as long as the power is applied. This memory consists of the number of flip flops with each flip flop storing 1 bit. It has less access time and hence, it is faster.
* **D RAM (Dynamic RAM):** [D RAM](https://www.geeksforgeeks.org/dram-full-form/) uses capacitors and transistors and stores the data as a charge on the capacitors. They contain thousands of memory cells. It needs refreshing of charge on capacitor after a few milliseconds. This memory is slower than S RAM.

.

Task 07



Task 08



Task 09 : = leetcode account

Task 10

Understand the code :---

Array 1: 11 34 66 75

Array 2: 1 5 19 50 89 100

Array after merging: 1 5 11 19 34 50 66 75 89 100

Task 10 : rewrite the code in such a way that it has to take unsorted list and then ,merge in an array the sorted list.

Task 11

What do you understand by hash table?

A Hash Table is a data structure used to store key-value pairs, like a mongo db .it stores the data in the type of key value pairs .  They provide efficient insertion, lookup, and deletion of data, often with an average time complexity of O(1). A Hash table is defined as **a data structure used to insert, look up, and remove key-value pairs** quickly.

HashMap multiple treads at a time . hash table—vice versa

Task 14;

Difference between hash map and hash table --

| **Feature** | **HashMap** | **Hash table** |
| --- | --- | --- |
| **Thread Safety** | ❌ Not synchronized (not thread-safe) | ✅ Synchronized (thread-safe) |
| **Performance** | Faster in single-threaded applications | Slower due to synchronization |
| **Null Keys/Values** | Allows **1 null key** and **multiple null values** | ❌ Does **not allow null keys or null values** |
| **Introduced In** | Java 1.2 (part of the Java Collections Framework) | Java 1.0 (legacy class) |
| **Preferred Use** | Modern, preferred for most use cases | Use only when thread-safety is needed without external sync |
| **Iterator Type** | Fail-fast (throws ConcurrentModificationException if modified during iteration) | Not fail-fast |
| **Package** | java.util | java.util |
| **Legacy?** | ✅ Modern | ❗ Legacy (use discouraged unless needed) |

HashMap<String, String> map = new HashMap<>();

map.put(null, "value"); // Allowed

map.put("key", null); // Allowed

Hashtable<String, String> table = new Hashtable<>();

table.put(null, "value"); // ❌ Throws NullPointerException

table.put("key", null); // ❌ Throws NullPointerException