مراجعه عامة مادة تراكيب البيانات

كلمات مهم نعرف معناها :

- consecutive integers: أعداد صحيحة متتالية
- contiguous: متجاور
- adjacent: مُتجاور
- conceptually: من الناحية المفهومية
- orderly manner: بطريقة منظمة
- constrained: مقيد
- frequent: متكرر
- explicit links: روابط صريحة
- finite: محدود
- efficient: فعال
- which dimension is prioritized for storing elements in memory?: أيّ بُعد يتم تفضيله لتخزين العناصر في الذاكرة؟

معلومات مهم نعرف معناها :

- المكدس (Stack) هو اختصار لمبدأ : LIFO "آخر من يدخل، هو أول من يخرج" (Last-In, First-Out).
- الطابور (Queue): يتبع مبدأ FIFO (First-In, First-Out)، أي "الأول يدخل، الأول يخرج"، تماماً مثل طابور الانتظار.
- المصفوفة (Array): هي مجموعة من العناصر يتم الوصول إليها عن طريق فهرس (index)، وليس لها نظام دخول وخروج محدد.
- الشجرة (Tree): هي هيكلية بيانات هرمية تستخدم للبحث والفرز، وليست لتنظيم الدخول والخروج بالتسلسل.

Stack :LIFO stands for Last-In, First-Out. This principle means that the last item you add to the structure is the very first one you remove.

Queue: A queue is FIFO (First-In, First-Out), like a line of people waiting. The first person in line is the first to be served.

.Array: An array is a collection of elements accessed by an index. It doesn't have a built-in LIFO or FIFO structure

Tree: A tree is a hierarchical data structure used for sorting and searching, not for sequential access like LIFO

The classification of data structures **based on the organization** of data is linear and non-linear.

The classification of data structures **based on the memory** of data is static and dynamic.

How to Choose the Right Data Structure?

The choice of an appropriate data structure depends on several key factors:

1. Based on the Operations Performed

Choose the structure that best supports the operations you will perform most frequently.
For frequent insertions and deletions: Use a Linked List.
For fast access by index (indexing): Use an Array.

2. Based on the Time Complexity of Operations

Select the structure that performs your most critical operations in the fastest possible time.
For frequent searches: Use a Binary Search Tree.

- It utilizes computer memory efficiently in comparison to a linear data structure.
- Its examples are trees and graphs

3. Based on the Space Complexity of Operations

Choose the structure that uses memory space most efficiently for your needs.
To store a large amount of data compactly: Use an Array, as it doesn't require extra memory for pointers.

4. Based on the Amount of Memory Used

This factor is important when the amount of data is unknown beforehand or changes often.
To store a large amount of data with flexible memory: Use a Linked List, as it can grow dynamically as needed.

اساله صح وخطا

1- Column-major and row-major storage result in identical memory addresses for elements when the matrix is symmetric. [ ]

2- An identity matrix is a special type of diagonal matrix [ ]

3- In a diagonal matrix, at least one of the elements on the main diagonal must be non-zero [ ]

4- The identity matrix is commutative with any square matrix A when multiplied (i.e., A * I = I * A) [ ]

5- In a lower triangular matrix, all elements above the main diagonal must be zero [ ]

ANS :

1- True

2-True

3-false

4-True

5- True

اساله نماذج

```
List List::copy (){
  List temp(size);
  for (int i = 0 ; i<size ; i++){
    temp.data[i] = data[i];
  }
  return temp;
}
```

```
void copyMatrix(matrix &m1) {
   for (int i = 0; i < 2; i++)
      for (int j = 0; j < 4; j++)
         Matrix[i][j] = m1.matrix[i][j];
}
```

الميزة	دالة نسخ المصفوفة (Matrix)	دالة نسخ القائمة (List/Array)
الهدف من الدالة	تعديل المصفوفة الحالية لتصبح نسخة من مصفوفة أخرى.	إنشاء قائمة جديدة تمامًا كنسخة من القائمة الحالية.
القيمة الراجعة (Return)	`void` (لا ترجع أي قيمة).	`List` (ترجع كائنًا جديدًا من نوع List).
حلقات التكرار (Loops)	حلقتان متداخلتان (`for` داخل `for`) لأنها تتعامل مع بعدين (صفوف وأعمدة).	حلقة واحدة (`for`) لأنها تتعامل مع بعد واحد.
المعاملات (Parameters)	تستقبل المصفوفة المراد نسخها كمعامل (`matrix &m1`).	لا تستقبل أي معامل، لأنها تنسخ من الكائن الحالي (`this`).

اساله نماذج

Question 1 (Human Style)

Given the array `arr = {45, 12, 88, 7, 24}`, what is the **3rd** element?

Answer: The 3rd element is **88**. (Explanation: When counting in human terms (position), we start from 1. The first element is 45, the second is 12, and the third is 88.)

Question 2 (Computer Style)

Given the array `arr = {45, 12, 88, 7, 24}`, what is the value at **index 3**?

Answer: The value at index 3 is **7**. (Explanation: In programming (index), counting starts from 0. So, index 0 is 45, index 1 is 12, index 2 is 88, and index 3 is 7.)

Question 3 (Computer Style)

What is the result of deleting the element at **index 0** from the array `arr = {100, 200, 300, 400}`?

Answer: `{200, 300, 400}` (Explanation: The element at index 0 is the first element, which is 100. Removing it leaves the rest of the array.)

Question 4 (Human Style)

What is the result of deleting the **4th** element from the array `arr = {100, 200, 300, 400}`?

Answer: `{100, 200, 300}` (Explanation: The 4th element in the list is 400. After removing it, the first three elements remain.)

اساله نماذج

Question 5 (Human Style)

To access the value **24** in the array `arr = {45, 12, 88, 7, 24}`, what **position** would you use?

Answer: The **5th** position. (Explanation: Counting from the start, 24 is the fifth element in the sequence.)

Question 6 (Computer Style)

To access the value **45** in the array `arr = {45, 12, 88, 7, 24}`, what **index** would you use?

Answer: Index 0. (Explanation: Array indexing always starts at 0, so the first element is located at index 0.)

Question 7 (Human Style)

If you swap the **1st** and **3rd** elements in the array `arr = {5, 10, 15, 20}`, what is the new array?

Answer: `{15, 10, 5, 20}` (Explanation: The 1st element (5) and the 3rd element (15) trade places.)

Question 8 (Computer Style)

If you swap the elements at **index 1** and **index 3** in the array `arr = {5, 10, 15, 20}`, what is the new array?

Answer: `{5, 20, 15, 10}` (Explanation: The element at index 1 (10) and the element at index 3 (20) trade places.)

اساله نماذج

Question 9 (Computer Style)

What element is located at `index = (length of array - 1)` for the array `arr = {3, 6, 9, 12, 15}`?

Answer: 15. (Explanation: The length of the array is 5. Therefore, the index is (5 - 1) = 4. The element at index 4 is the last one, 15.)

Question 10 (Human Style)

What is the value of the element that comes just before the **3rd** element in the array `arr = {3, 6, 9, 12, 15}`?

Answer: 6. (Explanation: The 3rd element is 9. The element immediately before it is the 2nd element, which is 6.)