DATA STRUCTURE AND ALGORITHM ASSIGNMENT
REG No. BSCCS/2025/40317
NAME: Quentin Omwenga Maoncha

Using a programming language of your choice, in groups, writecode to represent each of the data structures classification and types.
Research where the data structuretypes ar applied and give reasons why.
Give examples of aplications that are using the data structure types and algorithm.Give reasons why.
Research how data structures and algorithms work within systems.

1.Linear Data Structures
Linear structures organize data elememtns in a sewquence,where each item has a logical predecessor and a successor excpet for the very first and last items.
Beacause the data is arranfed linearly, it is easy for a computer to traverse the items one by one.

A)Stacks
A stack is a llinear data structure- that follows the **Last-In,First-Out (LIFO)** principle.
This means the last element added to the stack is the first one to be removed.

key Operations
Push :Adding an item to the top of the stack +
Pop : Removing the item from the top of the stack -
Peek(or Top) : Looking at the top item without removing it
isEmpty : Checking if the stack item has any items.

Implementation in Python
In python, the simplest wat to implement a stack is by using the built-in *list*.
We use .append() to push and .pop() to remove the last item.

```
# Initializing an empty stack
stack = []

# Pushing elements
stack.append("A")
stack.append("B")
stack.append("C")

# Popping an element (removes "C")
top_element = stack.pop()
```

In real world scenarios, the Stacks aare used in:
1. **Browser History;**when you navigate websites,eac new page is pushed onto a stack.
   When you hit a back button,the browser “pops”the current page to reveal the previous ones.
2. **Undo.Redo Mechanisms**;In software like Word or Photoshop,every edit you make is pushed onto an “undo stack”
3. **Function Calls(The Call Stack);**This is how programming languages work.When one function calls another,the computer pushes the current location onto a stack so it knows exactly where to treturn when one function finishes.

B)Queue
Queue follows the **First-In,First-Out(FIFO)**
The first element will be the first element out.

Ket Operations
Enqueue: Adding an element to the back(tail) of the queue +
Dequeue : Removing an element from the front (head) of the queue -
Peek: Looking at the front element without removing it
isEmpty: Checking if the queue is empty
Implementation in Python
The *.pop(0)* method can be used to remove the first item but is slow for large datasets because every ither items has to shift one position to the left.
Collections.deque, which is used because it is lightning-fast for adding or removing items from either end.from collections import deque

```
# Initializing the queue
queue = deque()

# Enqueue: Adding to the back
queue.append("Alice")
queue.append("Bob")
queue.append("Charlie")

# Dequeue: Removing from the front
first_served = queue.popleft() # Removes "Alice"
```

IN real world,
1.Operationg System Scheduling: Your computer's CPU can only do so many things at once.It uses a queue to manage tasks(playing music while you work) handling them in order as they arrived
2.Message Buffering:In video streaming(like YouTube),data is downloaded into a "buffer"(a queue). The player takes data from the front of the queue to show you the video,while new data is added to the back.
3.Print Jobs:If five people send documents to one printer,the printer uses a queue to ensure the first person who clicked "Print" gets their pages first.

C)Array
Python lists are actually **Dynamic Arrays.**
A bassic array is a collection of items stores in **contiguous memeory locations.**
This means all the items are sitting right next to each other in your computer's RAM.
Because the computer knows exactly where the first itekm is and how big each item is,it can jump to any index eg.(my_list[500])

Python's dynamic arrays handle this;
1**1.Allocation ;**Python sets aside a bit more space than you currently need.
2.Appending ;When you use *.append(),Python puts the item in that extra space.*
Resizing ;If the extra space runs out, Python automatically allocates a new,;arge chuk of memory,copies everything over,and deletes the old one.

Python Code Examples

```
# Creating a dynamic array (List)
fruits = ["Apple", "Banana", "Cherry"]

# Accessing by index (O(1) - Instant)
print(fruits[1]) # Banana

# Adding to the end (Amortized O(1) - Usually instant)
fruits.append("Date")

# Inserting/Deleting at the start (O(n) - Slow!)
# Because every other item has to shift over one spot.
fruits.insert(0, "Elderberry")
```

C)Linked Lists

Unlike an array,which stores data in one continuous block of memory, a linked list stores data in scattered “Nodes.”

Each node has two things;

1.**The Data**

2.A Pointer