University of Science and Technology of Hanoi *** **Final Examination** **Subject: Algorithms and Data Structures** Code: 02 No of pages: 02		***Date**: 24/10/2025* ***Academic year**: 2024–2025* ***Time**: 75 minutes* ***Important instructions*** 1. Only the course slides and your own exercises' code are allowed. 2. Copy or using Internet will lead to heavy penalty.	
Department	ICT	**Lecturer (or Head of Subject)**	**Dr. Đoàn Nhật Quang**
Student name		**Student's ID**	

Question 1: (8pts)
We want to compute a division between two integers (*divisor is different from 0*) using only subtraction and recursion as follows:

$$div(a,b) = \begin{cases} remainder = a \ if \ a < b \\ quotient = quotient + 1, div(a-b,b) \end{cases}$$

For example: quotient and remainder = 0 at the beginning

- div(10,3) = (quotient = 1, div(10-3,3)) = (quotient = 2, div (7-3,3)) = (quotient = 3, div(4-3,3))
- div(1,3) = (remainder = 1)
- Result: quotient = 3 and remainder = 1.

Note: *Use pointers to update the quotient and remainder in the recursive function. You can use the following declaration:* ***int div(int *q, int *r, int a, int b)***.

- Implement this recursive division algorithm in C/C++. (6 pts)
- Calculate the complexity of your algorithm. Justify your answer. (2 pts)

Question 2: (8 pts)
Let a **priority queue** of elements be defined as follows:

```
typedef struct {
    int data;
    int priority; // possible values: [0 .. 5]
} Element;
typedef struct {
    int size;
    Element data[100];
}Queue;
```

In this queue:

- Elements with higher priority values (smaller numbers) are placed at the front whenever a new element is added.

- Initial Queue:

Data: 30 | Priority: 1
Data: 25 | Priority: 3

- After **enqueue** (10, 0):

Data: 10 | Priority: 0
Data: 30 | Priority: 1
Data: 25 | Priority: 3

- The dequeue operation removes the front element (i.e., the element with the highest priority).

You have to implement the following functions in C/C++:

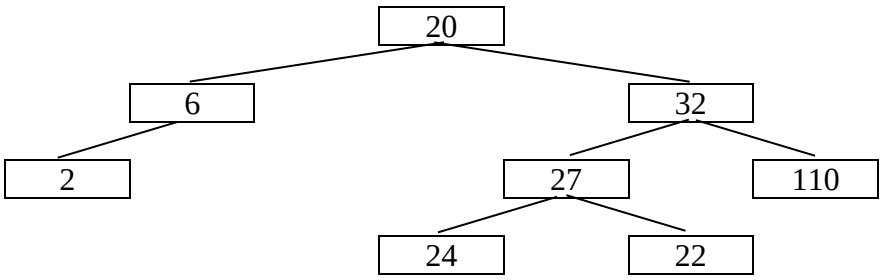
- Implement a function `init()` to initialize the queue and create an initial queue with at least six element**s** of your choice (data, priority).
- Implement the `enquene()` and `dequeue()` functions and other basic functions (`init()`, `display()`, etc.)
- Write a `main()` function to test all the above functions.

Question 3: (4 pts)

This problem requires you to traverse and display a Binary Search Tree (BST) level by level (from top to bottom, left to right) using a queue. The traversal process is as follows:

- Start from the root node.
- Insert the root into a queue.
- While the queue is not empty:
  - Dequeue the front element.
  - Display its value.
  - Enqueue its left and right children (if they exist).

- Show the traversal process and the final result. Justify your answer (2pts).
- Propose another traversal method to get the result sorted from the largest to the smallest values. (2pts)



--END --