Question 1

Create a class **DArray** that:

- Holds an integer pointer `arr` dynamically allocated.
- Stores the size `n` of the array. (do member initialisation for n = 5)

Implement:

- A **copy constructor** that deep copies the array.
- Parameterized constructor where DArray(int n), use this operator for initialization.
- Initialize and display functions to verify your output.
- A **destructor** to release memory.

In `main()`, create an **DArray** object, copy it using the copy constructor, and verify if the copied array works independently.

Question 2

Implement a class **Sequence** to store a sequence of non-negative integer values, and the length of the sequence. The class has the following private data members:

1. **int length** – the length of the sequence
2. **int *pseq** – a pointer to a dynamic integer array holding sequence of integers

The class shall provide the following public methods:

1. **Sequence()** – a default constructor that initializes length to 10 and store the sequence of all zeros in an array.
2. **Sequence(int lengthVal, int n1=0,int n2=0,int n3=0, int n4=0, int n5=0, int n6=0, int n7=0, int n8=0, int n9=0, int n10=0)** – another parameterized constructor should initialize the length and array to sequence values passed in the arguments.
3. **Sequence(Sequence &s)** – a copy constructor that creates a copy of a Sequence object.
4. **int getLength()** – a getter for length
5. **int* getSeq()** – a getter for the sequence of numbers
6. **void Sort(int n)** – a function that sorts the first **n** elements in the sequence array. You cannot use Bubble Sort Algorithm

void Rotate(int steps) – a method that rotates the sequence elements clockwise for the given steps

Example

```
Seq array = {1,3,4,6,2,6,0}

steps = 3

Sequence array after rotation = {2,6,0,1,3,4,6}
```

~Sequence() – a destructor to deallocate the dynamically created array