Part01

Q1. What is the default value assigned to array elements in C#?

based on the dtype of elements e.g. int-> 0, string->null

Q2. What is the difference between Array.Clone() and Array.Copy()?

- Copy(): shallow copy from a source arr to a destination arr (both must be existed) + we can specify some functionalities like coping from index i only to j (noting that the dist must be large enough to hold source elements)

- Clone(): deep copy but this creating a new instance and you have no option to specify coping from index I to j. it only copy the segment that the source existed in the RAM into the new instance (returned arr)

Q3. What is the difference between GetLength() and Length for multi dimensional arrays?

- GetLength(idx): returns the number of elements in the idxth row (0-based)
- Length(): returns the total number of elements in the entire array

Q4. What is the difference between Array.Copy() and Array.ConstrainedCopy()?

- Copy: if an exception thrown while the process, may be some elements are already copied
- ConstrainedCopy(): more guarantees e.g. all copy or nothing copied (this make the array unchanged if any exception occurred)

Q5. Why is foreach preferred for read-only operations on arrays?

Because it takes only a copy from the array into a virtual one so if we supposed that the write operations are supported: they're volatiled changes

Q6. Why is input validation important when working with user inputs?

The user may enter a bad input or doesn't enter a required input. Which may causing an exception

Q7. How can you format the output of a 2D array for better readability?

With using nested loops for rows and columns

Q8. When should you prefer a switch statement over if-else?

A lot of statements with little body code. Knowing that keys in switches stored in a hash table so we can jump directly to the desired key without passing all previous keys (unlike if-else)

Q9. What is the time complexity of Array.Sort()?

O(n log n)

Q10. Which loop (for or foreach) is more efficient for calculating the sum of an

array, and why?

For is better than foreach in performance (slightly) because of no overhead for coping data in an enumerator.

Foreach is better than foreach in readability which is leading to avoiding risks.

Part02

1- LinkedIn article about loops statements in Csharp

2- Define an enum called DayOfWeek with values: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.
Write a program that takes an integer input from the user (1-7) and prints the corresponding day using the enum.
Use Enum.Parse to convert an integer to an enum value.
Solved in .cs code

3- What happens if the user enters a value outside the range of 1 to 7?
Prints the value that the user input (key as a value)

Part03 (Bonus)

5. default size of the stack and the heap and what are consideration

Stack: 1MB per thread (we can resize it when creating a new thread using constructor overloading)

Heap: the size if dynamic [grow and shrink as needed] and managed by the GC

So, there isn't a default size

6. Time complexity:

The standard methodology to measure the performance of the code based on best/avg/worst cases. Usually, we depends on the worst case (O(..))

Time complexity parameter: n

Which is responding to number of repetition statements (a statement repeats n times or lgn times ..etc and may be constant [O(1)])