C# Example Questions 3

Question 1 (String) ***

Write a C# class that receives a String value from the user:

- Print the last 2 characters as output.
- Find how many characters does this String value has?
- Change the letters of the given input to uppercase letters and print it.
- Change the letters of the given input to lowercase letters and print it.
- Compare two strings (uppercase and lowercase) by using ignoring case. If they are equal print true, otherwise print false.
- Find the first occurrence of letter ‘a’ and print its index. (At the runtime, evaluate what happens if the string does not have an ‘a’ in it.)
- Replace forth character with “#” sign, for all occurrences.
- Concatenate the given string with “That’s it.” print the new value, evaluate the length of new string.
- Print the substring which is composed of: first white space, the character before white space and the character after white space.

Sample Output (with “My favorite game is cozy grove.” entry)

```
Please enter a String:
My favourite game is cozy grove.
Last two characters in your entry is: e.
Your entry has: 32 characters
The UPPERCASE value of the string you entered is: MY FAVOURITE GAME IS COZY GROVE.
The lowercase value of the string you entered is: my favourite game is cozy grove.
Are they equal? true
Index of character 'a' in your entry is: 4
The forth character is replaced.: My #avourite game is cozy grove.
The strings are concatenated: My favourite game is cozy grove.That's it It's length
becomes: 41
y f
```

Question 2 (String) *

Write a C# class which gets a String from the user and produce a concatenated string with escape characters “, \, ‘ like “user entry\’ and displays it on the screen with only one line of print().

Sample Output

```
Please enter a String:
Hello world!
Concatenated string is: "Hello world\'
```

Question 3 (String) *

Write a C# class that has a single line of code which writes the following output:

Output

```
Mutlu olmak istiyorsan, bir “amaca” bağlan;
      ‘insanlara’ ya da ‘eşyalara’ değil.
```

Question 4 (Mathematical Operations) **

Write a C# code to read two integers from user and determine whether

- 1st integer is divisible by 2nd integer
- 2nd integer is divisible by 1st integer
- If at least one of them does not hold, print the message “Numbers are not equal.”.

Note: Inputs cannot be negative numbers or 0. If user enters a negative number or 0 as an input, the program must not make any calculation and print a proper error message. Investigate the sample runs for details.

Sample Run:

```
Enter 1st integer:
2
Enter 2nd integer:
3
1st number is not divisible by 2nd number.
2nd number is not divisible by 1st number.
Numbers are not equal.
```

```
Enter 1st integer:
0
Enter 2nd integer:
4
Input cannot be a negative number or 0!
```

```
Enter 1st integer:
3
Enter 2nd integer:
3
1st number is divisible by 2nd number.
2nd number is divisible by 1st number.
Numbers are equal.
```

Question 5 (While & String) **

Write a C# program which reads a whole sentence from the keyboard and counts the number of 'a', 'e', 'i', and ' ' (space) characters in it. Use **while** loops.

Sample Outputs:

```
What is your sentence:
This is an example string.
Your sentence consists of 2 a characters.
Your sentence consists of 2 e characters.
Your sentence consists of 3 i characters.
Your sentence consists of 4 space characters.
```

```
What is your sentence:
I love playing guitar!
Your sentence consists of 2 a characters.
Your sentence consists of 1 e characters.
Your sentence consists of 2 i characters.
Your sentence consists of 3 space characters.
```

Question 6 (While and Mathematical Operations) **

Write a program by using C# while loop, which gets a positive number from the user. The program should display all the odd numbers up to the input number and the sum of them. Note that if a non-positive integer is entered by the user, the program should prompt the user and get another integer.

Sample Outputs:

```
Enter a positive number:
-3
Number is not valid. Please reenter:
0
Number is not valid. Please reenter:
9
1 3 5 7 9
Sum of the odd numbers up to 9 is 25
```

```
Enter a positive number:
11
1 3 5 7 9 11
Sum of the odd numbers up to 11 is 36
```

Question 7 (For Loop & Arrays) ***

Write a C# program that accepts an array of strings and finds the second largest item in lexicographic order. For example, if the array is {“aa”, “bb”, “cc”, “dd”, “ee”} then the largest item is “ee”.

- The code will tell the user to enter the length of array.
- Reads the inputs of user, n values.

Hint: You need to compare the elements of the array with each other and find the largest item.

Sample Output

```
Please enter the length of array:
5
Please enter 5 different Strings.
Adana Ankara Samsun Malatya Manisa
The Original Array is: [Adana, Ankara, Samsun, Malatya, Manisa]
The largest item is Samsun
```

```
Please enter the length of array:
7
Please enter 5 different Strings.
Aylin Beliz Taylan Buğra Doruk Can Pelin
The Original Array is: [Aylin, Beliz, Taylan, Buğra, Doruk, Can, Pelin]
The largest item is Taylan
```