Lab 4 String

In this lab, the length of all the input cstrings are less than 50.

Question-1.

Write a program to derive name and ID**.** The input contains the name and ID number of a student. In the main function, use the char array to store the input line, then split the name info and the ID info and store them in another two char arrays. Finally, print the name and ID respectively.
Note that the name may contains several spaces but ID number contains no space.

Hint: For storing the input line, you can define a char array like char arr[50], then use cin.getline(arr, 50) to store the input into the defined cstring.

Expected Outcomes:

Example 1

```
Enter the student info:
Jameson Alexander Grant 51012345
Student name is: Jameson Alexander Grant
Student ID is: 51012345
```

Example 2

```
Enter the student info:
Theodore Wesley Cole 40054321
Student name is: Theodore Wesley Cole
Student ID is: 40054321
```

Example 3

```
Enter the student info:
Victoria Joy 12345678
Student name is: Victoria Joy
Student ID is: 12345678
```

Question-2.

Write a program to find the location of a given char pattern in the string**.** Users should type in one input word and one input char pattern. Then the program can find the index of the last position of the given input char pattern occurring in the given word. The output index should be the index of the first char in the char pattern. If the char does not exist, the output value should be -1.

Expected Outcomes:

Example 1

```
The word and chars are:
hello llo
The last position of llo is: 2
```

Example 2

```
The word and chars are:
find-character-code -c
The last position of -c is: 14
```

Example 3

```
The word and chars are:
programming ran
The last position of ran is: -1
```

Question-3.

Write a program that receives two strings (str1 and str2) as input and outputs a new string that contains the characters from str1 and str2 and is alphanumerically ordered. For example, if str1 is "EFCHJI" and str2 is "BADG", the function must create and return a new string "ABCDEFGHIJ". You can assume that characters in the strings str1 and str2 are unique (i.e., they cannot appear in both strings).
You may use library <cstring> in your solution. You may assume that the arguments are **equal sized** strings.

Hint:
-- Before merge two strings together you may sort the two strings in alphanumerical order respectively.
-- The size of str1 and str 2 can be set as 50.

Expected Outcomes:

Example 1	Example 2
Input String 1: BADG Input String 2: FHCE Merged String: ABCDEFGH	Input String 1: OVPWX Input String 2: QRSTU Merged String: OPQRSTUVWX
Example 3	Example 4
Input String 1: JEI Input String 2: FHG Merged String: EFGHIJ	Input String 1: CBALJK Input String 2: FDEGHI Merged String: ABCDEFGHIJKL