Lab 11 Strings

Please test the correctness of your programs in **Q-2**, **Q-3**, **Q-4**, and **Q-5** using **PASS**.

Q-1.

Download and compile `digits.cpp`. Explain the output of the program.

```
#include <iostream>
using namespace std;

int main() {
    int digits[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int *p1 = digits;
    int *p2 = digits + 9;
    while (p1 <= p2) {
        cout << p1 << " " << p2 << " " << *p1 << " " << *p2 << endl;
        p1 = p1 + 2;
    }
    return 0;
}
```

Q-2.

Write a program to read a string and count the number of characters and the number of vowels (`'a'`, `'e'`, `'i'`, `'o'`, `'u'`) in the string. For the counting of vowels, you should consider both uppercase and lowercase versions of the characters.

Hint:

1. *You may use `cin >>` to read the string to a char array.*
2. *You may use the function `strlen()` in `<cstring>` to count the number of characters in the string.*
3. *You may need to define a counter (initialized to 0) and write a for-loop to examine each character in the string (until the null character '\0' is encountered). If the character is vowel, update the count.*

Expected Outputs:

Example 1	Example 2
Easter The number of characters is: 6 The number of vowels is: 3	Hello The number of characters is: 5 The number of vowels is: 2
Example 3	**Example 4**
AEIOU The number of characters is: 5 The number of vowels is: 5	CityU The number of characters is: 5 The number of vowels is: 2

Q-3.

Download `convert.cpp`. Modify it to convert all lowercase letters in a string to uppercase letters and convert all uppercase letters to lowercase letters. The input string may contain multiple words.

Hint: You can use `cin.getline()` function in `<cstring>` to read the string to a char array. You may also assume that the maximum size of char array is 50.

Expected Outputs:

Example 1	Example 2
HeLLO hEllo	CityU cITYu
Example 3	**Example 4**
Course 2311 cOURSE 2311	a Survey A sURVEY

Q-4.

Download `sort.cpp`. The program defines an array called course with six `cstring`s representing the course titles. Complete the program by sorting course in an ascending alphabetical order.

***Hints:** You can use `strcmp()` for comparison.*

The output of your program should look like the following:

```
C++ Programming
Data structures
English
Internet
Java Programming
Mathematics
```

Q-5.

Download `opt.cpp`. The program defines an array with 10 `cstring`s representing the students list, and an array with 6 `cstring`s representing the course list. The program has already *randomly* assigned each student to register one course. Complete the program so that it can

1) count the number of registrations for each course.
2) print the course list in descending order according to the number of registrations.
3) print the registered students' names for each course and students' names should be sorted in ascending alphabetical order.

Note: Your actual output is likely to be different from the expected output if you're not using **Microsoft Visual C++ 2015** (**PASS**). In case of any inconsistency between your output and **PASS** output, the **PASS** output shall prevail.

Expected Outputs:

```
Enter the seed for random number generation:
2018
James registers Internet
Iverson registers English
Wade registers Data structures
Jordan registers Internet
George registers Data structures
Curry registers Internet
Westbrook registers C++ Programming
Durant registers Java Programming
Kobe registers Java Programming
Harden registers Java Programming

Students' list:
Curry
Durant
George
Harden
Iverson
James
Jordan
Kobe
Wade
Westbrook

3 students register Internet: Curry James Jordan
3 students register Java Programming: Durant Harden Kobe
2 students register Data structures: George Wade
1 student registers English: Iverson
1 student registers C++ Programming: Westbrook
Nobody registers Mathematics
```