Lab 11 Pointers (II)

Please test the correctness of your program in **Q2** and **Q3** on **PASS**.
In this lab, the length of all input cstrings are less than 100.

Q-1. [For Practice]

Download **ex1.cpp, ex2.cpp** and **ex3.cpp**. Compile and execute the program. Explain the output.

<table>
<tr><th>File</th><th>Program segment</th></tr>
<tr><td>ex1.cpp</td><td><pre>
#include <iostream>
using namespace std;

int main() {
    int v = 5, *ptr;
    ptr = &v;
    *ptr = 42;
    cout << "v = " << v << endl;
    cin >> *ptr; // Let’s enter 100
    // What happens if you write cin >> ptr; ?
    cout << "v = " << v << endl;
    v = 7;
    cout << "*ptr is " << *ptr << endl;
    cout << "Address of v is " << ptr << endl;
    return 0;
}
</pre></td></tr>
<tr><td>ex2.cpp</td><td><pre>
#include <iostream>
using namespace std;

void f(int *a, int *b) {
    int *c;
    c = a;
    *c = *c + 10;
    *b = *b + 10;
}

int main() {
    int x = 3, y = 4;
    int *ptr1;
    ptr1 = &x;
    f(ptr1, &y);
    cout << "x = " << x << endl;
    cout << "y = " << y << endl;
    cout << "*ptr1 = " << *ptr1 << endl;
    return 0;
}
</pre></td></tr>
</table>

[Q-1. Continue...]

ex3.cpp

```cpp
#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. [For Practice]

Design a function to count the number of characters in each word, as well as the total number of words. Words are separated by spaces.

You need to design a function `countWord(char* arr, int* count)` to implement the operation. In this function, you should have a for loop to check through the string.

a. When current char is the first character of a word, store the current index **i** and increment the **count** using call by pointer;

b. When current char is the last character of a word, output the number of characters in the word.

c. Otherwise, skip.

Note:

1. This question is used to practice call by pointer. It can be easily implemented just in main function. Please do not do so and solve the question with the function described above.
2. Please also try **NOT** to use array access operators [] inside the function `countWord`.

Expected output:

Example

```
Enter the content of the string:
I am a CityU student
Word 1 has 1 characters.
Word 2 has 2 characters.
Word 3 has 1 characters.
Word 4 has 2 characters.
Word 5 has 7 characters.
The number of words in the string is: 5
```

Q-3. [will be marked]

You must click submit to submit your solution. Deadline is 11:59pm, April 4, 2023.

Write a program including a function called `deleteStr(char* str, char delete_char).` The return type is **char pointer**. The function is used to delete all the given **delete_char** existing in the given input string **str**. Use dynamic allocation to arrange space for the return cstring.

Note:

1. Use cin.get (delete_char) to get the input character because the char could be space.
2. You may first traverse the string str to record the number of given chars, making it convenient to set the size of dynamic array.
3. You can assume that the input strings **str** and char **delete_char** are not null.
4. Spaces can exist at the beginning or end of the input string.
5. No need to output the comment "//The input char is space" shown in the example 1.

Expected output:

Example 1

```
Enter the input string: we are happy
Enter the input delete_char: _        // The input char is space
The modified string is: wearehappy
```

Example 2

```
Enter the input string: what date is it today
Enter the input delete_char: t
The modified string is: wha dae is i oday
```

Example 3

```
Enter the input string: count the number of it
Enter the input delete_char: u
The modified string is: cont the nmber of it
```