

Programming Design, Spring 2015

Suggested Solution for Homework 2

Solution provider: Tammy Chang

Problem 1

(a) Yes, the program is correct!

First the program declare a character variable `a`, initializing it to zero. And the program will read in one character entered by user as `a`. Then check the condition in `while`, which means that if `a` doesn't equal to both `'y'` and `'Y'`, the program will ask the user to enter one character as `a` again. Therefore, if `a` is equal to either `'y'` or `'Y'`, one of the conditions in `while` will not be true, the program will terminate. In other words, using `&&` as the logic operator in the `while` condition-checking section is correct. If we change it to `||` then if we enter `'Y'`, even though the first part of condition-checking (`a` equals `'y'`) is false, the second part (`a` equals `'Y'`) will be true, instead of terminating the program will ask the user to enter again.

(b) Even if we enter multiple characters, this program can still be compiled. The reason of this phenomenon is that it will store these characters into a buffer queue. And every time the program automatically `cin` one character in the queue as `a`, to continue rest of the process. Hence, there's no different between entering `"abc"` in one time and entering `'a'`, `'b'`, `'c'` separately.

Problem 2

In the statement `i = i + 1`; the program will return `i` after executing `i + 1` operation. And in statement `++i`, the program will also return `i` after executing `++` operation. Therefore the two statements are equivalent. On the other hand, in `i++` statement, the program will return `i` before executing `++` operation. Here's an example:

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

int main() {
    int i = 5;
```

```
int a = i = i+1;
cout << "The value of a, i is: "<< a <<" , " << i << endl;

int b = ++i;
cout << "The value of b, i is: "<< b <<" , " << i << endl;

int c = i++;
cout << "The value of c, i is: "<< c <<" , " << i << endl;
return 0;
}
```

Output:

The value of a, i is: 6, 6

The value of b, i is: 7, 7

The value of c, i is: 7, 8

You can see that a, b are both equal to the value of i after i+1 and ++ operations, while in i++ statement, the program first return value of i, which equals to 7, so c equals to b. Then do the ++ operation and return it as i, so i equals to 8.

Problem 3

Please see the .cpp file.