

## CS 31 Worksheet Solutions Week 2

This worksheet is entirely **optional**, and meant for extra practice. Some problems will be more challenging than others and are designed to have you apply your knowledge beyond the examples presented in lecture, discussion or projects. All exams will be done on paper, so it is in your best interest to practice these problems by hand and not rely on a compiler.

### Concepts

If Statements, Cin, Variables, Doubles, Ints

### Reading Problems

1. What do you think will happen when running the following program? Circle where the bug occurs and explain what incorrect behavior will happen. Add a fix.

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

int main()
{
    cout << "Enter your name: ";
    string name;
    getline( cin , name );

    cout << "Enter your UID: ";
    int UID;
    cin >> UID;

    cout << "Enter your major: ";
    string major;
    getline( cin , major );

    cout << "Enter your residence hall: ";
    string hall;
    getline( cin , hall );

    cout << "\n" << UID << " is the ID of " << name << ", a "
        << major << " student who lives in " << hall << endl;
}
```

When the user types an integer and hits Enter in response to the prompt for the UID, the `cin >> UID` consumes the digits of the integer that the user enters, but does NOT consume the newline that hitting

Enter delivers to the program. After the prompt for the major appears, the `getline(cin, major)`, finding that newline still unconsumed, will consume it and be satisfied, since it is supposed to read characters up to and including a newline. Therefore, it will not wait for the user to type anything, so the prompt for the residence hall is written immediately.

To fix the problem, any unconsumed input characters after the `cin >> UID` has finished must be consumed and discarded before the `getline(cin, major)` executes. To do this, after the `cin >> UID`; add the line `cin.ignore(10000, '\n');` to consume and discard characters up to and including the newline. (Instead of 10000, you could use any number larger than the number of characters you might expect a user to type before hitting Enter.)

2. What is the output produced by the following code?

```
int a = 340;
int b = 22;
if (a % 10 == 0) {
    a /= 10;
    cout << a << endl;
    if ((a + b) % 2 == 0) {
        b--;
        cout << b << endl;
    }
}
```

34  
21

The code first checks if `a` is divisible by 10. Since 340 is divisible by 10, `a` is divided by 10, so `a` is now 34, which is written out. Then it checks if `a+b` is even.  $34+22 = 56$ , which is true, so `b` is decremented by 1.  $b = 22-1=21$ , which is written out.

3. This code snippet is supposed to read an integer and tell you whether it is even or odd. Find the 3 bugs contained in the code and fix them.

```
int n;
cin >> n;
string evenOrOdd = "";
if (n%2 = 0) {
    evenOrOdd = even;
} else {
    evenOrOdd = odd;
}
cout << "The number is " << evenOrOdd << endl;
```

```
n%2 = 0 should be n%2 == 0
evenOrOdd = even; should be evenOrOdd = "even";
evenOrOdd = odd; should be evenOrOdd = "odd";
```

## Programming Problems

1. You are driving too fast and a police officer stops you. Write a program that prints the size of your ticket depending on your speed: no ticket, small ticket, or big ticket. If your speed is no more than 60, write no ticket; if it is over 60 and no more than 80, write small ticket; if it is over 80, write big ticket. There's an exception if it's your birthday: if your answer is yes, the limits are 5 higher in all cases. Here's a sample dialog:

```
What was your speed? 82
Is it your birthday? yes
small ticket
```

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

int main()
{
    cout << "What was your speed? ";
    int speed;
    cin >> speed;
    cin.ignore(10000, '\n');

    cout << "Is it your birthday? ";
    string birthday_ans;
    getline(cin, birthday_ans);
    if (birthday_ans == "yes")
        speed -= 5; // equivalent to treating the limits below as
                  // if they were 5 higher

    if (speed <= 60)
        cout << "no ticket" << endl;
    else if (speed <= 80)
        cout << "small ticket" << endl;
    else
        cout << "big ticket" << endl;
}
```

2. Write a program that reads a string of length one and writes the integer that the string represents as a Roman numeral (I is 1, V is 5, X is 10, L is 50, C is 100, D is 500, M is 1000; the letter may be upper or lower case). If the string is not a valid Roman numeral of length one, write `Invalid input`. Sample:

```
Enter Roman numeral letter: V
Its value is 5
```

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

int main()
{
    cout << "Enter a one-letter Roman numeral: ";
    string roman;
    getline(cin, roman);
    cout << "Its value is ";
    if (roman == "I" || roman == 'i')
        cout << 1 << endl;
    else if (roman == "V" || roman == 'v')
        cout << 5 << endl;
    else if (roman == "X" || roman == 'x')
        cout << 10 << endl;
    else if (roman == "L" || roman == 'l')
        cout << 50 << endl;
    else if (roman == "C" || roman == 'c')
        cout << 100 << endl;
    else if (roman == "D" || roman == 'd')
        cout << 500 << endl;
    else if (roman == "M" || roman == 'm')
        cout << 1000 << endl;
    else
        cout << "Invalid input" << endl;
}
```

3. Write a program that reads two numbers and a command: add, subtract, multiply, or divide. If the input is valid, write the result of the operation; otherwise, write an appropriate error message. Sample dialog:

```
Enter the first number: 3
Enter the second number: 7
Enter the command: multiply
Result: 21
```

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

int main()
{
    cout << "Enter the first number: ";
    double operand1;
    cin >> operand1;
    cout << "Enter the second number: ";
    double operand2;
    cin >> operand2;
    cin.ignore(10000, '\n');
    cout << "Enter the command: ";
    string command;
    getline(cin, command);

    if (command == "add")
        cout << "Result: " << operand1 + operand2 << endl;
    else if (command == "subtract")
        cout << "Result: " << operand1 - operand2 << endl;
    else if (command == "multiply")
        cout << "Result: " << operand1 * operand2 << endl;
    else if (command == "divide")
    {
        if (operand2 == 0)
            cout << "Cannot divide by zero!" << endl;
        else
            cout << "Result: " << operand1 / operand2 << endl;
    }
    else
        cout << "Command is not add, subtract,multiply, or divide!"
            << endl;
}
```

4. Write a program that reads three numbers and writes their mean.

```
Number: 6
Number: 2
Number: 5
The mean is 4.3333
```

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

int main()
{
    cout << "Number: ";
    double num1;
    cin >> num1;
    cout << "Number: ";
    double num2;
    cin >> num2;
    cout << "Number: ";
    double num3;
    cin >> num3;
    cout << "The mean is " << (num1 + num2 + num3) / 3.0 << endl;
}
```

5. Write a program that reads three numbers and writes their median.

```
Number: 6
Number: 2
Number: 5
The median is 5
```

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

int main()
{
    cout << "Number: ";
    double num1;
    cin >> num1;
    cout << "Number: ";
    double num2;
    cin >> num2;
    cout << "Number: ";
    double num3;
    cin >> num3;
```

```

cout << "The median is ";
if (num1 <= num2)
{
    if (num2 <= num3)
        cout << num2;
    else if (num1 <= num3)
        cout << num3;
    else
        cout << num1;
}
else if (num1 <= num3)
    cout << num1;
else if (num2 <= num3)
    cout << num3;
else
    cout << num2;
}

```

6. Write a program that determines whether or not a year is a leap year in the Gregorian calendar. A year is a leap year if it is divisible by 4, unless it is divisible by 100 and not by 400. (Ignore the fact that this definition is not correct for years before the Gregorian calendar was adopted.)

```

Year: 2004
2004 is a leap year

```

```

#include <iostream>
using namespace std;

int main()
{
    cout << "Year: ";
    int year;
    cin >> year;
    if (year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) )
        cout << year << " is a leap year" << endl;
    else
        cout << year << " is not a leap year" << endl;
}

```