CS 31 Worksheet 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.
Solutions are written in red. The solutions for **programming** problems are not absolute, it is okay if your code looks different; this is just one way to solve the specific problem.

1. What is the output of the following code?

```
int a = 10;
int b = 22;
while (a / 2 >= 1) {
  a--;
  cout << a << endl;
  if ((a + b) % 2 == 0) {
    a--;
    cout << a << endl;
    b /= 2;
  }
}
```

9
8
7
6
5
4
3
2
1

2. This code snippet tries to print all prime numbers between 3 and a given input *n*. Find the 3 bugs contained in the code and fix them.

```
int n;
cin >> n;
int candidate = 3
while (candidate < n) {
        bool isPrime = true;
        for (int x = 2; x < candidate; x++) {
                if (candidate % x == 0){
```

```
                    isPrime = false;
                }
        }

        if (isPrime) {
                cout << candidate << " ";
        }
        candidate = candidate + 1;
}
```

3) What does the following code snippet output?

```
void mystery(int a, int b) {
      int count = 0;
      while (count < 2) {
            a = a + b/2;
            b = a + 5;
            cout << "a: " << a << " b: " << b << endl;
            count++;
      }
}

int main() {
      int a = 5, b = 10;
      cout << "a: " << a << " b: " << b << endl;
      mystery(a, b);
      cout << "a: " << a << " b: " << b << endl;
}
```

```
a: 5 b: 10
a: 10 b: 15
a: 17 b: 22
a: 5 b: 10
```

4. What does the following code snippet output?

```
      int mystery(char code) {
        switch(code) {
          case 'a':
           case 'b':
           case 'c':
             cout << "spooky";
             break;
           case 'd':
```

```
            cout << "feeling";
            break;
          case '1':
            cout << " ";
            break;
          case '2':
            cout << "?";
          default:
            cout << endl;
            break;
        }
        return( 0 );
      }

      int main() {
        mystery( '2' );
        mystery( 'a' );
        mystery( 'Z' );
        mystery( 'd' );
        mystery( '1' );
        mystery( 'c' );

}
```

```
?
spooky
feeling spooky
```

5. Consider the two programs shown below. If there are no errors in the program, show what will be printed by each of the following programs. If there are any errors in the program explain what is wrong.

a.
```
#include <iostream>
using namespace std;
int three(int,int);
int main()
{
   int a,b;
   a = 3;
   b = 4;
   cout << three(a,b);
   return( 0 );
}
int three(int x, int y)
{
```

b.
```
#include <iostream>
using namespace std;
int three(int,int);
int main()
{
   int f;
   f = 1;
   int i = 1;
   while( i < 5 ) {
      f = three(i,f);
      cout << f << endl;
      i = i + 1;
   }
```

```
    int a;
    a = x + y;
    return a;
}
```

a. 7

```
    return( 0 );
}
int three(int a, int b)
{
    int z;
    z = a + a * b;
    return z;
}
```

b. 2
6
21
88

Programming Problems

1) Write a program that takes in an integer N where N > 0, and outputs a comma-separated list of all the factors of N.

Sample input:
12
Sample output:
1,2,3,4,6,12

```
int main() {
        int n;
        cin >> n;

        cout << "1";      // 1 is always a factor
        for (int i = 2; i <= n; i++) {
                if (n % i == 0) {
                        cout << "," << i;
                }
        }
        cout << endl;
        return 0;
}
```

2) Write a function that returns whether or not two integers are palindrome number. A palindrome number is a value that reads the same forwards and backwards. HINT: Use % and / to break the value into its different digits. For example:

intPalindrome(62, 26) should return true
intPalindrome(154, 451) should return true
intPalindrome(25, 56) should return false

```
bool intPalindrome (int first, int second) {
      int reverse = 0; // calculate the reverse of first
      int digit = 0;
      do {
            digit = first % 10;
            reverse = (reverse * 10) + digit;
            first = first / 10;
      } while (first != 0);

      return reverse == second;
}
```

3) Write a function checkeven which accepts 3 integer parameters and prints YES if all three numbers are even. Otherwise the function prints NO. Then write a main program with the statements to read in 3 integers. Then call your function to determine whether the data entered was all even.

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

void checkeven(int num1,int num2,int num3)
{
   if (num1 % 2 == 0 && num2 % 2 == 0 && num3 % 2 == 0)
      cout << "YES\n";
   else
      cout << "NO\n";
}


int main( )
{
   int n1, n2, n3;
   cout << "Enter three numbers: ";
   cin >> n1 >> n2 >> n3;
   checkeven(n1,n2,n3);
}
```

4) Write a function that returns the cost of mailing a package, given the weight of the package in pounds and ounces, and a cost per ounce are supplied as arguments to the function. Recall that there are 16 ounces in a pound. Then write a main program with the statements to read in the weight of a package (in pounds and ounces), and the cost per ounce for mailing. Then call your function to calculate the mailing cost, and print the mailing cost.

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

float postage (int ounces, int pounds, int costperounce)
{
    ounces = ounces + pounds * 16;
    return (ounces * costperounce);
}

int main( )
{
  int pounds, ounces;
  double ouncecost, cost;
  cout << "Enter package weight in pounds and ounces: ";
  cin >> pounds >> ounces;
  cout << "Enter mailing cost per ounce: ";
  cin >> ouncecost;
  cost = postage(ounces,pounds,ouncecost);
  cout << "The cost to mail this package is: " << cost << endl;
  return( 0 );
}
```

5) Write a function that does integer division without using the division operator (/). Return -1 if second number is 0. Your main driver code should recognize that -1 and print an error statement as shown below. Then write a main program with the statements to read in 2 integers. Then call your function.

integerDivide(6, 2) should return 3
integerDivide(2, 0) should return -1 and then your main program should print:
Error: Cannot divide by 0

This is a solution assumes that the parameters will be positive.

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

int integerDivide(int x, int y) {
  int count = 0;
  int iterator = y;
  if ( y == 0 )   return( -1 );
  while (x >= y) {
```

```
      y += iterator;
      count++;
   }
   return count;
}



int main( )
{
   int a, b;
   cin >> a >> b;
   int answer = integerDivide( a, b );
   if (answer == -1)
   {
      cout << "Error: cannot divide by 0" << endl;
   }
   else
   {
      cout << "answer = " << answer << endl;
   }
}
```

6) Write a function that does integer multiplication without using the multiplication operator (*). Return true if the multiplied value equals the third argument, false otherwise. Then write a main program with the statements to read in 3 integers. Then call your function to determine whether the values entered were correct for use by integerMultiply.
HINT: Perform the multiplication by repetitively using addition.

integerMultiply(6, 2, 12) should return true
integerMultiply(2, 0, 7) should false

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

bool integerMultiply(int x, int y, int answer) {
  int total = 0;
  while (x >= 1) {
      total += y;
      x = x - 1;
   }
   return (answer==total);
}
```

```
int main( )
{
   int a, b, c;
   cin >> a >> b >> c;
   bool answer = integerMultiply( a, b, c );
   if (answer)
   {
      cout << a << "*" << b << " = " << c << endl;
   }
   else
   {
      cout << a << "*" << b << " != " << c << endl;
   }
   return( 0 );
}
```