$\qquad$
March 18, 2011
Score $\qquad$

Read each problem and follow the directions carefully. When details are omitted, such as which data type to use, make any appropriate choice of your own. Except in cases when directed to write a program, you need only turn in your function.

1. (20 points) The $n$th harmonic number $H_{n}$ is defined as follows:

$$
H_{n}=1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\cdots+\frac{1}{n}
$$

or equivalently,

$$
H_{1}=1 ; \quad H_{n}=H_{n-1}+\frac{1}{n}, n=2,3,4, \ldots
$$

For example, the 5 th harmonic number is

$$
H_{5}=1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}=H_{4}+\frac{1}{5}=2.28 \overline{3}
$$

(a) Write a complete function with function header double H ( int n ) that uses a do-while loop to compute $H_{n}$.
(b) Write a complete function with function header double H ( int n ) that uses recursion (i.e. H calls itself) to compute $H_{n}$.
(c) Suppose we define the generalized harmonic numbers as follows:

$$
G H(n, p)=1+\frac{1}{2^{p}}+\frac{1}{3^{p}}+\cdots+\frac{1}{n^{p}}
$$

where $n$ and $p$ are positive integers. Write a complete function that uses a for loop to compute $G H(n, p)$. In your function, let $p$ be an optional argument with default value 1 .
2. ( 7 points) Consider the function $f$ defined as follows:

$$
f(x)=\left\{\begin{array}{crrr}
\cos x^{2}, & & -1 \leq x<0 \\
e^{x^{2}}, & & 0 \leq x<1 \\
\sqrt{x}+\frac{\ln x}{x}, & & 1 \leq x \leq 2
\end{array}\right.
$$

(a) Write a function that returns the value of $f(x)$.
(b) Use our Monte Carlo integration program (available on the Class Examples web page) to approximate $\int_{-1}^{2} f(x) d x$.
3. (6 points) Use the Bisection program (available on the Class Examples web page) to approximate all three solutions of $x^{3}-5 x^{2}+2 x+3=0$ correct to 6 decimal places. For each of the solutions, state your solution and the user input that generated that solution.
4. (6 points) For this problem you will need the program template available on the Class Examples web page (click on the Test Problem link).
(a) Starting with the program template, add the following components to the main function:

- Declare and initialize an 8 -element float array named $x$ containing the elements $0.0,1.0,2.0,3.0,4.0,5.0,6.0$, and 7.0.
- Call the bitrev function defined in the template with the command bitrev ( $\mathrm{x}, 8$ ). This function call is already included in the template.
- After the function call, use a while loop to output the elements of array $x$, one per line. Use the field width specifier setw() to output each number right justified in a field of width 4.
(b) (3 points extra credit) The function bitrev applies a process called the bit-reversal permutation. Explain what the function did. Do not try to make sense of the function, instead do some Google research.

5. (8 points) In this problem, you will write a program that displays the Collatz sequence associated with a given positive integer. Basically, to generate the sequence, you start with a positive integer $n$. If $n$ is even, divide it by 2 . Otherwise, multiply it by 3 and then add 1. Repeat the process indefinitely or until you get to 1 . Here is the pseudo-code:
```
Input positive integer n
Input maximum number of iterations M
Do until n=1
    If n is even
        n = n / 2
    Else
        n = 3n+1
    Increment your count by 1
    Output n
    If your count exceeds M
        Stop
```

Write and test the corresponding C++ program. Use a do-while loop, and inside the loop use a break statement if your max number of iterations is exceeded. Recall that to test if a number is even you can use if ( $\mathrm{n} \% 2==0$ ). (It is not known whether the Collatz sequence for every positive integer terminates at 1, but every known Collatz sequence ends with $4,2,1$.)
6. (3 points) Refer back to problem 1 and your function that computes the harmonic numbers. Explain why the following main function would be a terribly inefficient way to list the first 100 harmonic numbers.

```
int main()
{
    for ( short i = 1; i <= 100; i++ ) {
            cout << "H(" << i << ") = " << H(i) << endl;
    }
    cout << "\n\n"; system( "PAUSE" );
    return( EXIT_SUCCESS );
}
```

