Honor Code Acknowledgment (signature): ________________________________

MAKE SURE YOUR NAME IS ON THE TOP OF EACH SHEET.

All functions should be properly commented. You do not need to do any error checking within your functions. A sample string.h file is included at the back of the exam. Where it is important, spaces are printed with the underscore character.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problem 1 -- 10 points

Define 5 of the 6 terms below. You do not need a formal definition, just a good description. Limit your answers to less than 50 words per items. You may use an example if it would be helpful. Clearly mark the term you do not wish to include; otherwise I will discard the final term.

a. Iteration:

b. Predicate:

c. Algorithm:

d. Associativity:

e. Boolean Expression:

f. Concatenate:
Problem 2 -- 10 points

Write a function, FlipNumber, which takes a two digit integer and returns a two digit integer with the digits flipped. For example:

\[
\begin{align*}
\text{FlipNumber}(52) & = 25 \\
\text{FlipNumber}(16) & = 61 \\
\end{align*}
\]

The function prototype for FlipNumber is:

\[
\text{int FlipNumber(int num)}
\]
Problem 3 -- 10 points

What is the output of this program when the user enters 5 at the Enter a number prompt? The function prototype of the pow function is:

```cpp
int pow(int x, int y)
// purpose: return x to the y power
// preconditions: y>=0
```

```
#include <iostream.h>
#include <math.h>       // for the pow function
```

```
int main()
{
    int k, j;
    int number;

    cout << "Enter a number: ";
    cin >> number;
    k = 0;
    while (k < number)
    {
        cout << k << "::";
        for (j = (number - k); j > 0; j -= 1)
        {
            cout << pow(2, j) << "_";
        }
        cout << endl;
        k++;
    }
    return 0;
}
```

Output:
Problem 4 -- 10 points

The following program, trig.cc, has several semantic errors. Carefully compare the outputs to make sure you catch everything. The author intended the output to be as follows:

Desired 1:

```
enter an angle (in degrees): 45
45 degrees = 0.785398 radians.
sin(0.785398) = 0.707107
cos(0.785398) = 0.707107
tan(0.785398) = 1
```

Desired 2:

```
enter an angle (in degrees): 90
90 degrees = 1.5708 radians.
sin(1.5708) = 1
cos(1.5708) = 0
tan(90) is undefined
```

But instead, the actual output is:

Actual 1:

```
enter an angle (in degrees): 45
45 degrees = 0.785398 radians.
sin(0.785398) = 0.707107
cos(0.785398) = 0.707107
tan(90) is undefined
1
```

Actual 2:

```
enter an angle (in degrees): 90
90 degrees = 1.5708 radians.
sin(1.5708) = 1
cos(1.5708) = 1
tan(90) is undefined
3.73205e+07
```
Identify each error directly on the program and show how to correct the error.

```cpp
#include <iostream.h>
#include <math.h>         // for sin(), cos(), tan()

const double PI = 3.1415926;

int main()
{
    double degrees;
    double radians;

    cout << "enter an angle (in degrees): ";
    cin >> degrees;

    radians = PI / 180 * degrees;

    cout << degrees << " degrees = ";
    cout << radians << " radians." << endl;

    cout << "sin(" << radians << ") = ";
    cout << sin(radians) << endl;

    cout << "cos(" << radians << ") = ";
    cout << cos(radians) << endl;

    if (degrees == 90)
        cout << "tan(90) is undefined" << endl;
    else
        cout << "tan(" << radians << ") = ";
        cout << tan(radians) << endl;

    return 0;
}
```
Problem 5 -- 15 points
What is the output for each of the following code fragments. Assume each statement is part of a program that compiles and runs. Use the underscore character instead of a space. You must show work in order to receive partial credit.

```cpp
string where = "Chapel_Hill";
string who = "Grant_Hill";
int number = 7;
int x = 2;

a) cout << where + who.Upcase() + "_" << where.length() << endl;

b) cout << number / x << "&_" << number % x << endl;

c) cout << static_cast<double>(number) / x + x << endl;

d) cout << 3 * (x - 1) << "_" << 3 * x - 1 << endl;

e) cout << who.find(where.substr(6, 4)) << "+5=" << endl;
```
Problem 6 -- 10 points

Write a function `MaxInt3`. Which takes three integer parameters and returns the largest of the three. The prototype for `MaxInt3` is:

```c
int MaxInt3(int n1, int n2, int n3);
// purpose: determine the largest of 3 integers.
```

So, the function call `MaxInt3(2, 11, 6)` will return 11.

**Added Twist:**

Here is the `MaxInt2` function; which returns the largest of 2 integers. You can use `MaxInt2` in your function, but *you may not use an if statement in your function!!*

```c
int MaxInt2(int n1, int n2)
// purpose: To determine the maximum of two integers
// precondition: none
// postcondition: none
{
    if (n1 > n2)
    {
        return n1;
    }
    return n2;
}
```
Problem 7 -- 20 points
The library has noted a rash of books not being returned on time. With the popularity of email, they have decided to send patrons email messages to notify them of overdue books.

Write a function named OverDueBook to accomplish this task. The following two function calls produce the output shown:

```
OverDueBook("Mike","Tapestry",2,10,1.55);
OverDueBook("Wei","Operating Systems",12,11,3.25);
```

Corresponding Output:

**Dear Mike,**
The book you checked out, Tapestry, was
due on February 10 and is now OVERDUE.
Your current fine is: $1.55
Thank you for returning it promptly.

**Dear Wei,**
The book you checked out, Operating Systems, was
due on December 11 and is now OVERDUE.
Your current fine is: $3.25
Thank you for returning it promptly.

MonthFromInt:
You may use the following function in your function:

```
string MonthFromInt(int month);
// purpose: Return the string form of a month, i.e. January,
// given a integer between 1-12.
// precondition: 1 <= month <= 12
// postcondition: none.
```

Please write the function on the back of THIS page.
Problem 8 -- 15 points

The ChannelIterator class is designed to process information about shows currently on a local cable TV system. A ChannelIterator variable can be used to process all of the available channels, once the member function First is called. The channels will not be processed in numeric order; i.e. channel 5 may be processed before channel 3.

ChannelIterator iterator("feb20");
iterator.First();
cout << iterator.Channel() << "_";
cout << iterator.Show() << ":_";
cout << iterator.Category() << endl;
iterator.Next();
cout << iterator.Channel() << "_";
cout << iterator.Show() << ":_";
cout << iterator.Category() << endl;

The code fragment show above produces output such as:

4_Blues Brothers:_Movie
9_Curling:_Sports

Write the function PrintShowByCategory whose prototype is given below. You do not need to know what data a ChannelIterator contains; that is, the channels, shows and categories are a mystery to the programmer.

void PrintShowByCategory(string FileName, string Category)

Calling PrintShowByCategory("feb21","Sports") might yield the following:

Category:_Sports  ←  A similar line is required.
8_Superbowl
10_The Masters
4_Duke vs UNC Basketball
7_Curling

The channeliterator.h file is included in the back of the exam.

Please write the function on the back of THIS page. Note that your function must output the "Category" line as shown in the sample output.
CPstring.h

// string class
//
// string() -- construct a string
//
// int length() -- The number of characters in the string
//
// string substr(int pos, int len) -- Returns a
//        string of len characters beginning at pos.
//        The first character of a string in position zero.
//
// int find(string s) -- Position at which s begins within
//                      the string (or NPOS if the string
//                      doesn’t exist). First character
//                      has position 0.
//
// string Downcase() -- Lower-case equivalent. Non-
//                      alphabetic characters are left
//                      unchanged.
//
// string Upcase() -- Upper-case equivalent. Non-
//                      alphabetic characters are left
//                      unchanged.

class string
{
    public:
        string();       // Constructor
        int length();
        string substr(int pos, int len);
        int find(string s);
        string Downcase();
        string Upcase();
    private:
        // omitted
}
channeliterator.h

// ChannelIterator class
//
// ChannelIterator(string FileName) -- construct an iterator
//   The FileName contains the name of a file which
//   contains all of the TV data. The constructor
//   handles all of the file operations. You can not
//   "use" the iterator until First() is called.
//
// void First() -- Starts the iterator at the first channel
//
// int Channel() -- Returns the number of the current channel
//
// string Show() -- returns the name of the show on the
//                  current channel.
//
// string Category() -- returns the category of the show on
//                      the current channel.
//                      i.e. comedy, soap opera, etc.
//
// void Next() -- advances to the next channel
//
// bool IsDone() -- returns true if there are no more states
//                  to process (in which case the iterator
//                  has no current channel). returns false
//                  if there are more states to process.
//

Class ChannelIterator
{
    public:
        ChannelIterator();
        void First();
        int Channel();
        string Show();
        string Category();
        void Next();
        bool IsDone();
    private:
        // omitted
    }