CS 101, Spring 2013 — May 14th — Final Exam   Name: ______________________

Question 1. [3 points] What output is printed by the following C code?

```
int x = 20;
printf("%i %i\n", x / 3, x % 3);
```

Question 2. [3 points] Say you are writing a program to keep track of information about YCP students.

(a) What C data type would you use to represent the number of students enrolled at YCP?

(b) What C data type would you use to represent a student's cumulative GPA?

(c) Write variable declarations to define variables that represent the quantities described in parts (a) and (b) above.

Question 3. [3 points] What output is printed by the following code (which begins on the left and continues on the right)?

```
#include <stdio.h>

int main(void) {
  int a, b, c;

  c = a * b;
```

```
  a = 3;
  b = 2;

  printf("c is %i",c);

  return 0;
}
```

a. `c is 5`

b. `c is 6`

c. The program does not compile.

d. It is not possible to predict the output.

Question 4. [3 points] What output is printed by the following C program, which begins on the left and continues on the right?

```
#include <stdio.h>

void f(int p) {
  p = p + 1;
}
```

```
int main() {
  int q = 55;
  f(q);
  printf("%i\n", q);
  return 0;
}
```

a. 0

b. 1

c. 55

d. 56

e. Does not compile.

Question 5. [3 points] What output is printed by the following C program, which begins on the left and continues on the right?

```
#include <stdio.h>

void f(int *p) {
  *p = *p + 1;
}
```

```
int main() {
  int q = 55;
  f(int &q);
  printf("%i\n", q);
  return 0;
}
```

a. 0
b. 1
c. 55
d. 56
e. Does not compile.

Question 6. [3 points] What output is printed by the following program (which begins on the left and continues on the right)?

```
#include <stdio.h>

void double_it(int *p){
  *p = *p * 2;
}
```

```
int main(void) {
  int var = 10;

  double_it(&var);
  printf("%i\n", var);

  return 0;
}
```

a. `10`
b. `20`
c. The program does not compile.
d. It is not possible to predict the output.

Question 7. [3 points] What output is printed by the following C program (which begins on the left and continues on the right)?

```
#include <stdio.h>

void f(int a[], int i, int j) {
  int t = a[i];
  a[i] = a[j];
  a[j] = t;
}
```

```
int main() {
  int v[3] = { 1, 2, 3 };
  f(v, 0, 2);
  printf("%i %i %i\n", v[0], v[1], v[2]);
  return 0;
}
```

a. 1 2 3
b. 3 2 1
c. 0 1 2
d. 2 1 3
e. Does not compile.

Question 8. [3 points] What output is printed by the following program (which begins on the left and continues on the right)?

```
#include <stdio.h>

struct Stuff {
  int hq;
  int lq;
};

double doThis(struct Stuff s) {
  double result;

  result = s.hq/s.lq;
  s.hq = s.lq;
  return result;
}
```

```
int main(void) {
  struct Stuff myStuff;
  myStuff.hq = 6;
  myStuff.lq = 4;
  double qu = 0.0;

  qu = doThis(myStuff);

  printf("%.1lf %i %i",
         qu, myStuff.hq, myStuff.lq);
}
```

a. 0.0 6 4
b. 1.0 6 4
c. 1.0 4 4
d. 1.5 4 4
e. Does not compile.

Question 9. [5 points] What output is printed by the following program (which begins on the left and continues on the right)?

```
#include <stdio.h>

struct MoreStuff {
  double a;
  double b;
};

struct MoreStuff takeStuff(
  struct MoreStuff *s);

int main(void)
{
  struct MoreStuff mine, yours;
  mine.a = 4.0;
  mine.b = 2.0;
  yours.a = 0.0;
  yours.b = 0.0;

  yours = takeStuff(&mine);

  printf("%.1lf %.1lf ", mine.a,
                         mine.b);
  printf("%.1lf %.1lf ", yours.a,
                         yours.b);

  return 0;
}
```

```
struct MoreStuff takeStuff(
  struct MoreStuff *s)
{
  struct MoreStuff mine;

  mine.a = s->b;
  s->b = s->a;
  mine.b = s->b;

  return mine;
}
```

a. 4.0 2.0 0.0 0.0

b. 4.0 2.0 2.0 4.0

c. 2.0 2.0 2.0 2.0

d. 4.0 4.0 2.0 4.0

e. Does not compile.

Question 10. [3 points] Assume that when the following code is run, the user enters `3` as the input (which is then stored in the variable `num`). What output will be printed?

```
int ans = 0;
int num;

printf("Enter an integer: ");
scanf("%i", &num);

for (int i = 1; i <= num; i++) {
  ans *= i;
}
printf("Answer is %i\n", ans);
```

a. `Answer is 0`

b. `Answer is 1`

c. `Answer is 6`

d. It is not possible to predict the output.

Question 11. [3 points] Briefly describe the error in the following code.

```
int row, col;

printf("How many rows do you want in the table: ");
scanf("%i", &row);

printf("How many columns do you want in the table: ");
scanf("%i", &col);

for (row = 0; row <= row; row = row + 1) {
  for (col = 0; col <= col; col = col + 1) {
    printf("%3i ", row * col);
  }
  printf("\n");
}
```

Programming Questions

Note: For all of the programming questions, you should **not** need to write any code to obtain user input.

Note: Make sure your programs produce the output in **exactly** the format described, including capitalization and punctuation. You may not receive credit for programs that produce incorrectly formatted output.

Getting started: Start **Cygwin Terminal** and **Notepad++** closing any open tabs in **Notepad++**. (Note: Do *not* open **ANY** other programs.) Your instructor will give you the name of a zip file. In Cygwin Terminal, run the following commands:

```
cd h:
mkdir -p CS101
cd CS101
curl -O http://faculty.ycp.edu/~dhovemey/spring2013/cs101/assign/zipfile
unzip zipfile
cd CS101_Final
```

Substitute the name of the zip file for *zipfile*.

Editing code: Use Notepad++ to open the source file (e.g., `question12.cpp`) referred to in the question. Do not open any files other than the ones for the exam.

Compiling: To compile the program for Question 12, run the following command in Cygwin Terminal:

```
make question12.exe
```

Change the number as appropriate for the other questions (e.g., `question13.exe`).

Running: To run the program for Question 12, run the following command in Cygwin Terminal:

```
./question12.exe
```

Change the number as appropriate for the other questions (e.g., `question13.exe`).

To submit: In Cygwin Terminal, run the command

```
make submit
```

Enter your Marmoset username and password when prompted.

Good luck!

Question 12. [15 points] The program in `question12.cpp` computes the daily balance of a bank account with an initial value of $100.

The program should repeat the following steps using a loop:

1. Output to the user the current account balance, using the format `There are $`*N*` in your account`, where *N* is the user's current balance with two decimal places of precision. The account balance is stored in the `account` variable.
2. Prompt the user as to whether or not they would like to make a purchase, 1 for yes, 0 for no. If the user answers 0 for no, then the loop should terminate and the program should exit.
3. Prompt the user to enter an amount for his/her purchase. Then, call the `makePurchase` function described in the next step, passing a pointer to the `account` variable and the purchase amount as arguments.
4. Add a prototype and definition for a `makePurchase` function. The prototype should look like this:

   ```
   void makePurchase(double *pAccount, double purchaseAmt);
   ```

   The `makePurchase` function should check to see if there is enough money in the account to make the purchase. If so, it should deduct the purchase amount from the account balance (`pAccount` is a pointer which contains the address of the variable used to store the account balance). If not, it should print the message `Insufficient funds`.

The steps above should continue until the user enters `0` when prompted to make a purchase.

Make sure your program's output *exactly* matches the examples shown below.

Example run 1 (user input in **bold**):

```
There are $100.00 in your account
Make a purchase (1=yes, 0=no)? 1
Cost of your purchase: 25.50
There are $74.50 in your account
Make a purchase (1=yes, 0=no)? 1
Cost of your purchase: 75.00
Insufficient funds
There are $74.50 in your account
Make a purchase (1=yes, 0=no)? 0
Thank you for using YCP National Bank
```

Example run 2 (user input in **bold**):

```
There are $100.00 in your account
Make a purchase (1=yes, 0=no)? 1
Cost of your purchase: 35.00
There are $65.00 in your account
Make a purchase (1=yes, 0=no)? 0
Thank you for using YCP National Bank
```

Question 13. [15 points] The program in `question13.cpp` determines if a user input value is a multiple of another user input value (you may assume the user enters two integers).

Follow the steps below to complete the program:

1. Write a *function prototype* for a function named `isMultiple` that has two `int` parameters called `a` and `b` and returns an `int`.
2. Inside the `main` function, write a *function call* to call the `isMultiple` function, passing `num1` and `num2` as arguments and storing the return value in `ans`.
3. Now write the complete function definition for the `isMultiple` function that takes two `int` parameters `a` and `b` *by value* and returns an `int` result. The return value should be **1** if `a` is *evenly divisible* by `b` (i.e. the first number is a multiple of the second number) and **0** otherwise.

Four example runs are shown below (user input in **bold**):

```
Enter two integers: 12 5
Result is 0

Enter two integers: 2 4
Result is 0
```

```
Enter two integers: 12 4
Result is 1

Enter two integers: 4 2
Result is 1
```

Important: You may not change any code in the `main` function except to add a function call to your newly created `isMultiple` function.

Hint: the `%` operator computes the integer remainder from dividing one integer value by another integer value. For example, `12 % 5` equals 2 which is the remainder from dividing `12` by `5`.

Question 14. [15 points] The program in `question14.cpp` defines the following struct data type:

```
struct Rectangle {
  float x1, y1, x2, y2;
};
```

This `struct` defines a rectangle by storing the x and y coordinate values of two corner points `(x1,y1)` and `(x2,y2)`. You may assume that `x1` < `x2` and `y1` < `y2`.

The program reads four `float` values, which are the x and y coordinate values of two corner points, and stores them in a `Rectangle` variable called `r`. It then calls a function called `area` and prints the final output. **Important**: You may not change any code in the `main` function except to uncomment the following line: **result = area(&r);**

Your task is to complete the program according to the following specification:

- you **must** add a prototype and definition for the `area()` function which has a single *pointer* to a `struct Rectangle` paramater and returns a `float` value representing the area of the rectangle
- the function should print the initial values of the points from the parameter
- the function should compute and print the width of the rectangle (with two decimal places of precision)
- the function should compute and print the height of the rectangle (with two decimal places of precision)
- the function should compute the area of the rectangle and return this value
- **DO NOT MODIFY** `main()` except to uncomment the function call

Hint: The width and height of a rectangle can be defined as `x2 - x1` and `y2 - y1`, respectively. The area is defined as the width of the rectangle times the height.

Example run (user input in **bold**):

```
Enter x1: 5.0
Enter y1: 8.0
Enter x2: 15.0
Enter y2: 20.0
x1 is 5.000000
y1 is 8.000000
x2 is 15.000000
y2 is 20.000000
Width is 10.00
Height is 12.00
Area is 120.00
```

Question 15. [20 points] Finish writing the program in the source file `question15.cpp`. The program prompts the user for an integer value representing the length of an array (up to a maximum of 10) and then inputs the integer array values. The program will then call a function named `arrayExtents()` which sets the first element of the array to its minimum and the last element to its maximum and then prints these extents.

Your task is to complete the program according to the following specification:

- you **must** write the `arrayExtents()` function and function prototype which has an integer array parameter and an integer length parameter, and return an integer value indicating the number of swaps that were performed
- the function should print the initial values in the array
- the function should print the minimum, i.e. smallest value of the array and the *index* where it occurs
- the function should print the maximum, i.e. largest value of the array and the *index* where it occurs
- the function should *swap* the *first* element in the array with the *minimum*, i.e. smallest, element in the array *only* if it is not currently the first element
- the function should *swap* the *last* element in the array with the *maximum*, i.e. largest, element in the array *only* if it is not currently the last element
- the function should count the number of swaps that were performed and return this value
- **DO NOT MODIFY** `main()`

An example run of the program is below (user input in bold):

```
Enter array length: 7
Enter array values: 4 1 5 6 3 7 2
Initial values 4 1 5 6 3 7 2
Min value 1 at index 1
Max value 7 at index 5
1 4 5 6 3 2 7 - 2 swaps done
```

Another example run of the program is below (user input in bold):

```
Enter array length: 6
Enter array values: 3 8 13 9 11 6
Initial values 3 8 13 9 11 6
Min value 3 at index 0
Max value 13 at index 2
3 8 6 9 11 13 - 1 swaps done
```

Question 16. [10 points] [Bonus Question] In the program `question16.cpp`, write the prototype and definition for a function called `furthestDistance()` that takes four parameters: an array of `struct Point` elements called `p`, an `int` parameter called `num_points`, which specifies how many points the array has, and two `int` parameter pointers called `p1` and `p2` specifying the array indices for the two points with the greatest distance. The function should return a double value, indicating the distance between the furthest points.

The function should evaluate all pairs of points to determine which two are separated by the furthest distance. Note the distance between two points can be computed as

$$\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

The `struct Point` structure contains two `int` fields `x` and `y`.

The `main()` function (which is provided for you) reads the initial array as input, stores them in the array, calls the `furthestDistance()` function, and then prints the two furthest points along with the distance. You will not need to change any of the code in `main()`.

For example, if the input points are

(1,2) (5,3) (7,1) (2,4)

The output would be

```
Furthest distance 6.083 between (1, 2) and (7, 1)
```

Hints:

- Use the `sqrt` function to find the square root of a `double` value
- You will most likely need nested loops to check each *pair* of elements