

Question 1. [5 points] Find the errors in the following code:

```
int a[10][10];
for (int i = 0; i <= 10; i++) {
    for (int j = 0; i <= 10; i++) {
        a = i*j;
    }
}
```

Question 2. [5 points] What is the output of the given code?

```
#include <stdio.h>

int main(void) {
    for (int row = 1; row <= 7; row = row+2) {
        for (int col = 1; col <= row; col++) {
            printf("*");
        }
        printf ("\n");
    }
    return 0;
}
```

Question 3. [5 points] Which of the following is most likely to be correct if used in a C program? (Circle one.)

- a. `for (int i = 0; i < 10; i++);`
- b. `for (int i = 0, i < 10, i++)`
- c. `for (int i = 0, i < 10, i++);`
- d. `for (int i = 10; i >= 10; i++)`
- e. None of the above is likely to be correct

Question 4. [5 points] True or False: In order for the `rand` function to return a sequence of values that varies when the program is run multiple times, it is necessary to seed the random number generator using the `srand` function.

Question 5. [5 points] What output is printed by the following program?

```
#include <stdio.h>

int main(void) {
    for (int i = 0; i < 10; i++) {
        printf("-");
        if (i == 5) {
            printf("|");
        }
    }
    return 0;
}
```

Question 6. [5 points] In the space provided below declare a symbolic constant named `HALFPI` which is equal to $3.14/2$.

Question 7. [5 points] Rewrite the following code using a `while` loop instead of a `for` loop.

```
for (int i = 0; i < 100; i++) {
    printf("%i\n", i);
}
```

Question 8. [5 points] What is the result of this print statement?

```
int a[5] = { 4, 35, 99, 12, 7 };
printf("%i", a[5]);
```

Question 9. [5 points] What type of loop is most appropriate if it is difficult or impossible to predict (before the loop starts) how many iterations will be necessary? For example, if you were asking a user if they would like to play again.

Question 10. [5 points] In the space provided below declare a two-dimensional array called `table` representing a 3x3 grid of real numbers.

Programming Questions

Note: For all of the programming questions, you should use `scanf` to read the input value(s) required by the program.

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** (or Linux terminal) and **Notepad++** (or a Linux text editor such as Pluma). (Note: do *not* open any other programs.) Your instructor will give you the name of a zip file. In your terminal, run the following commands:

```
cd h:
mkdir -p CS101
cd CS101
curl -O http://faculty.ycp.edu/~dhovemey/spring2015/cs101/assign/zipfile
unzip zipfile
cd CS101_Exam2
```

Note that in the `curl` command, the `-O` has the letter ‘O’, not the digit ‘0’. Also: skip the `cd h:` step if you are using Linux.

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

Editing code: Use your text editor to open the source file (e.g., `question11.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 11, run the following command in the terminal:

```
make question11.exe
```

Change the number as appropriate for the other questions (e.g., `question12.exe`).

Running: To run the program for Question 11, run the following command in the terminal:

```
./question11.exe
```

Change the number as appropriate for the other questions (e.g., `question12.exe`).

To submit: In Cygwin Terminal, run the command

```
make submit
```

Enter your Marmoset username and password when prompted.

Good luck!

Question 11. [25 points] Complete the program in `question11.cpp` so that it prints out all of the powers of 2 (1, 2, 4, 8, 16, etc.) up to (and if appropriate, including) the integer entered by the user.

Important: All of the powers of 2 should be printed on a single line, and each should be separated by at least one space.

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

```
Enter an integer: 42  
1 2 4 8 16 32
```

Another example run (user input in **bold**):

```
Enter an integer: 211  
1 2 4 8 16 32 64 128
```

Hints:

- Use an accumulator variable to store the current power of two (think about what its initial value should be)
- The next power of two can be found by multiplying the current power of two by 2

Question 12. [25 points] Complete the program in `question12.cpp` as follows:

The program declares an array called `balances` with three rows and two columns. This part of the program is provided: *do not change it*.

The program should read six `double` values and store them in the array such that the first pair of values is stored in the first row, the second pair of values is stored in the second row, and the third pair of values is stored in the third row. Then, the program should use a for loop that calculates the total and average of the values for each of the two columns of the array.

The program should print four lines of output in the format shown in the examples below.

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

```
Enter six values:  
1.2 2.3  
3.4 4.5  
5.6 6.7  
First column total: 10.20  
Second column total: 13.50  
First column average: 3.40  
Second column average: 4.50
```

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

```
Enter six values:  
4.8 2.7  
4.2 8.8  
1.0 9.4  
First column total: 10.00  
Second column total: 20.90  
First column average: 3.33  
Second column average: 6.97
```

Note that it does not matter whether the user enters the input values on a single line or multiple lines (because `scanf` doesn't care). The examples above use multiple lines to emphasize which rows and columns of the array each value should be stored in.

Hints:

- Treat the first dimension of `balances` as rows and the second dimension as columns
- Use `scanf` to read values from the user
- Use a `for` loop or loops to compute the totals of the two columns
- Make sure the four lines of output printed by the program use *exactly* the format shown in the examples