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

[0] [1] [2] [3]

```
int arr[4] =  {13, 1, 9, 18};
printf("%i\n", arr[3]);
```

18

Question 2. [3 points] Specify a function prototype for the following function:

```
int AreaOfRectangle(int L, int W) {
    return L * W;
}
```

Prototype: _____ int AreaOfRectangle(int L, int W); _____

Question 3. [4 points] Specify a function (prototype and definition) that accepts one `double` value as an argument and returns a `double` value representing the argument cubed.

Prototype: double cubeVal(double val);

Definition:
```
double cubeVal(double val){
    double result;
    result = val * val * val;
    return result;
}
```

Question 4. [3 points] You are writing a program which calls for the repeated use of a constant value, 9.81, representing gravity. Show how to declare a symbolic constant called `GRAVITY` to represent this value.

#define GRAVITY 9.81

Question 5. [4 points] What output is printed by the following program (which begins on the left and continues on the right)?

```c
#include <stdio.h>

int divideByTwo(int x);

int main(void) {
    int input = 5, output;
    output = divideByTwo(input);
    printf("%i\n", output);
    return 0;
}
```

```c
int divideByTwo(int x) {
    int result = x/2;
    return result;
}
```

[handwritten: 5/2 = 2 (integer division)]
[handwritten: 5 next to output; 2 near return result]
[handwritten answer: 2]

Question 6. [4 points] What output is printed by the following program? Explain briefly.
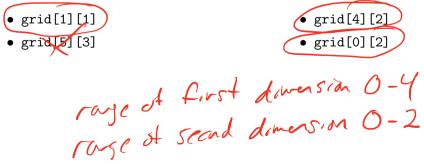
```c
#include <stdio.h>

int main(void) {
    int arr[2] = {3, 8};
    int x = arr[2];
    printf("%i\n", x);
    return 0;
}
```

[handwritten: Undetermined output since arr[2] is out-of-bounds program may print random output or crash]

Question 7. [4 points] Consider the following array declaration: `int grid[5][3];`

For the element references below, circle the ones that are legal (in-bounds), and cross out the ones that are not legal (out-of-bounds).

- grid[1][1] *(circled)*
- grid[5][3] *(crossed out)*
- grid[4][2] *(circled)*
- grid[0][2] *(circled)*

[handwritten: range of first dimension 0-4]
[handwritten: range of second dimension 0-2]

Question 8. [6 points] What output is printed by the following program?

```c
#include <stdio.h>

int main(void) {
   int a[4] = { 0, 3, 1, 0 };
   int b[4] = { 1, 8, 4, 7 };

   for (int i = 0; i < 4; i++) {
     printf("%i ", b[a[i]]);
   }
   return 0;
}
```

1   7   8   1

(b[0]   b[3]   b[1]   b[0])

Question 9. [6 points] Consider the following program (which begins on the left and continues on the right):

```c
#include <stdio.h>

int main(void) {
   int arr[10];
   int count = 0;

   printf("Enter 10 integers: ");
   for (int i = 0; i < 10; i++) {
     scanf("%i", &arr[i]);
   }
```
```c
   for (int i = 0; i < 10; i++) {
     if (arr[i] + arr[i+1] == 7) {
        count++;
     }
   }

   printf("%i pairs sum to 7\n",
          count);
   return 0;
}
```

Briefly explain the bug in this code.

In second loop, the last iteration i=9 but
the if condition references arr[i+1] which is
arr[10] that is out-of-bounds

Question 10. [8 points] Consider the following partially-specified program (on the left), and expected output (on the right):

```
#include <stdio.h>

char choose(int x, int y);

int main(void) {
  for (int j = 0; j < 6; j++) {
    for (int i = 0; i < 6; i++) {
      printf("%c", choose(i, j));
    }
    printf("\n");
  }
  return 0;
}
```

```
******
*    *
*    *
*    *
*    *
******
```

Show a definition of the **choose** function so that the program will produce the output shown.

Hint: a literal character can be specified using single quotes (e.g., `'A'` is the literal character `A`.)

```
char choose(int x, int y){
    char result = ' ';
    // if first or last row, j=0 or 5 passed into y
    if (y==0 || y==5){
        result = '*';
    }
    // if first or last column, i=0 or i=5 passed into x
    else if (x==0 || x==5){
        result = '*';
    }
    return result;
}
```

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** and **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/spring2014/cs101/assign/zipfile
unzip zipfile
cd CS101_Exam2
```

Note that in the `curl` command, the `-O` has the letter 'O', not the digit '0'.

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

Editing code: Use Notepad++ 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 Cygwin 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 Cygwin 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. [30 points] Complete the program in `question11.cpp` so that it prompts the user to enter 10 integer values. After the values have been entered, it should print a line with the text `Rotated values:`, followed by a line with the values "rotated" one position to the right, meaning that the last value entered should be printed first, and then the other values should be printed in the order in which they were entered.

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

```
Enter 10 integer values:
8 7 45 49 42 20 8 44 1 40
Rotated values:
40 8 7 45 49 42 20 8 44 1
```

Hints:

- Use an array to store the values entered by the user
- Print all values on a single line
- Make sure that there is a space after each value

Handwritten annotation (right side):

```
int arr[10];
printf("Enter 10 values: \n");
for (int i=0; i<10; i++){
    scanf("%i", &arr[i]);
}
printf("Rotated values: \n");
printf("%i ", arr[9]);
for (int i=0; i<9; i++){
    printf("%i ", arr[i]);
}
```

Question 12. [25 points] Modify the program in `question12.cpp` so that it prints the following output:

```
-----X----
------X---
-------X--
--------X-
---------X
X---------
-X--------
--X-------
---X------
----X-----
```

Important requirement: Do *not* modify the `main` function in any way. Instead, write a prototype and definition for the `choose` function as indicated by the comments in `question12.cpp`.

Hints:

- A literal character can be specified using single quotes (e.g., `'A'` is the literal character `A`.)

Handwritten annotation (bottom):

```
notice the pattern is
    y=0    X is at x=5
    y=1    X is at x=6    }  X = (y+5) % 10
    ⋮
    y=5    X is at x=0
```

```
char choose (int x, int y) {
    char result;
    result = '-';
    if ( x == (y+5)%10) {
        result = 'X';
    }
    return result;
}
```