

Question 1. [5 points] Circle and briefly explain the error(s) in the following code:

```
double x, y;  
result = (x)(x) + y;  
printf("%lf\n", result);
```

should be $(x * x)$ for multiplication
x and y are uninitialized

variable must be declared before being referred to, e.g.
double result;

Question 2. [10 points] Show code to (a) prompt the user to enter a height in meters, (b) read the user's height value and store it in a variable called `height`, (c) compute the equivalent height in feet, and (d) print the height in both meters and feet. There are approximately 3.28 feet per meter. For example, if the user input is 3.21, then the code should print

3.21 meters = 10.52 feet

```
double height;  
printf("Enter height in meters: ");  
scanf("%lf", &height);  
  
double height_ft = height * 3.28;  
  
printf("%.2lf meters = %.2lf feet\n", height, height_ft);
```

Question 3. [5 points] What output is printed by the following code?

```
int a, b;  
a = 3;  
b = a * 4;  
a = 6;  
printf("a=%i, b=%i\n", a, b);
```

$a = 6, b = 12$

Question 4. [5 points] What output is printed by the following code?

```
int a, b, c, d;  
a = 10;  
b = 4;  
c = a / b;  
d = a % b;  
printf("c=%i, d=%i\n", c, d);
```

$c = 2, d = 2$

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

```
int a, b;  
double c;  
a = 10;  
b = 4;  
c = a / b;  
printf("c=%lf\n", c);
```

$c = 2.000000$

this is an integer division because both a and b have type int

Question 6. [5 points] Refer to the following code:

```
int val;  
scanf("%i", &val);  
if (val > 20) {  
    printf("X");  
}  
printf("Y");
```

no else, so this executes unconditionally

(a) What output is printed if the user enters 20?

Y

(b) What output is printed if the user enters 30?

XY

Question 7. [5 points] Refer to the following code:

```
int a = 30;  
int val;  
scanf("%i", &val);  
if (a = val) {  
    printf("X");  
} else {  
    printf("Y");  
}
```

(a) When the user enters 30, the output is "X" and when the user enters 20, the output is also "X". Why?

this is an assignment, not an equality comparison.
The "value" of the assignment (20) is considered to be true because 20 is nonzero.

(b) When the user enters 0, the output is "Y". Why?

That would make the value of the assignment 0, which is considered false in a boolean context.

(c) Explain how to fix the code so that when the user enters 30 the output is "X", and when she enters anything other than 30, the output is "Y".

use the equality comparison operator == rather than the assignment operator

Question 8. [10 points] Refer to the following code:

```
int temperature;
scanf("%i", &temperature);
if (temperature > 90) {
    printf("That's too hot\n");
} else if (temperature >= 70 && temperature < 90) {
    printf("That's just right\n");
} else {
    printf("That's too cold\n");
}
```

(a) When the user enters 100, what is the output?

That's too hot

(b) When the user enters 90, what is the output?

That's too cold

(c) When the user enters 70 what is the output?

That's just right

(d) Fix the code so that "That's too cold" is only printed when the temperature is less than 70.

make this \leq (less than or equal to)
rather than $<$ (strictly less than)

Bonus Question. [4 points] Refer to the following code:

```
int val;  
scanf("%i", &val);  
if (val <= 20);  
    printf("X");  
    printf("Y");
```

SEMILOLON OF DEATH

(a) When the user enters 30, the output is "XY". Why?

The semicolon after the condition of the if statement counts as the body of the if statement, so the statement `printf("X");` executes unconditionally.

(b) Fix the code so that it outputs "X" when the user inputs a number less than or equal to 20, and "Y" when she enters a number greater than 20.

Should be:

```
if (val <= 20) {  
    printf("X");  
} else {  
    printf("Y");  
}
```

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++** and make sure ALL TABS are closed. (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/spring2017/cs101/zipfile
unzip zipfile
cd CS101_Exam01
```

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 your text editor to open the source file (e.g., `question9.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 9, run the following command in the terminal:

```
make question9.exe
```

Change the number as appropriate for the other questions (e.g., `question10.exe`).

Running: To run the program for Question 9, run the following command in the terminal:

```
./question9.exe
```

Change the number as appropriate for the other questions (e.g., `question10.exe`).

To submit: In Cygwin Terminal, run the command

```
make submit
```

Enter your Marmoset username and password when prompted.

Good luck!

Question 9. [25 points] In a quadratic equation of the form

$$ax^2 + bx + c = 0$$

the roots of the equation (values of x which make the equation true) can be found using the quadratic formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Complete the program in `question9.cpp` so that it prompts the user for values of a , b , and c , and then prints two lines of output of the form

```
Root = X.YY
```

where $X.YY$ is one of the two roots of the quadratic equation where a , b , and c are the coefficients. You may assume that $b^2 - 4ac$ will not be negative (since the square root of a negative number can't be represented as a real number.)

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

```
Enter a: 5
Enter b: 6
Enter c: 1
Root = -0.20
Root = -1.00
```

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

```
Enter a: 1
Enter b: -3
Enter c: -4
Root = 4.00
Root = -1.00
```

Hints:

- Use an appropriate data type for each variable in your program
- Make sure each root is printed on a separate line, using *exactly* the form specified above
- It does not matter what order the roots are printed in
- Be sure to test your program on the example inputs shown above
- Use the `sqrt` function to compute a square root: for example, `sqrt(q)` computes the square root of the value in the variable `q`
- Note that \pm means “plus or minus”, so you’ll need to apply the quadratic formula twice (once for plus and once for minus) in order to find both roots

Question 10. [25 points] Complete the program in `question10.cpp` as follows.

For any integer $2 \leq k \leq 12$ entered by the user, determine which of the integers from 2 to 6 is a factor of k . For example, 2 is a factor of 12 because 12 is evenly divisible by 2. 5 is *not* a factor of 12 because 12 is not evenly divisible by 5.

For each factor, the program should print a single line of output of the form

```
X is a factor of k
```

where X is a factor of the input value, and k is the input value.

If k does not have any factors in the range 2 through 6, issue a message saying so. If the user enters a value for k outside of the valid range, issue an error message, and do not determine the factors. Make sure to include an appropriate prompt to the user to enter a value for k .

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

```
Enter an integer (2 <= k <= 12): 12
2 is a factor of 12
3 is a factor of 12
4 is a factor of 12
6 is a factor of 12
```

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

```
Enter an integer (2 <= k <= 12): 7
7 has no factors
```

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

```
Enter an integer (2 <= k <= 12): 27
27 is an invalid input
```

Hints:

- Use an appropriate data type for each variable in your program
- You can use the modulo operator to determine whether an integer is evenly divisible by another integer
- Possible strategy: first check to see if the input is out of range; if the input is in range, check whether it's a multiple of 2, then check if it's a multiple of 3, etc.
- Be sure to test your program on the example inputs shown above