

## COP 3223 Program #4: Loops

### Program A: Olympics (olympics2.c)

Hopefully you figured out the typo in Program 3...the first modern Summer Olympics were held in Athens, Greece in 1896.

For this program, instead of checking if a year had a Summer Olympics, you'll ask the user to enter a start year and an end year, and print out each year in that range, inclusive on both ends, that the Summer Olympic games took place.

#### Input Specification

Let **s** be the start year entered by the user and **e** be the end year entered by the user. You are guaranteed that both are integers that satisfy  $1800 \leq s \leq e \leq 2025$ .

#### Output Specification

Print out, one year per line, in numeric order, each of the years in the given range when the Summer Olympic Games took place. (Note: This isn't user friendly, but will be easier for the TAs to grade.)

#### Sample Program Run (User Input in Bold and Italics)

What is the starting year for your range?

**1915**

What is the ending year for your range?

**1930**

1920

1924

1928

#### Sample Program Run (User Input in Bold and Italics)

What is the starting year for your range?

**2000**

What is the ending year for your range?

**2021**

2000

2004

2008

2012

2016

2021

### **Program B: Goldfish Tank (goldfish.c)**

You have bought a goldfish tank and are growing your collection of fish. This program is going to be a menu driven program where the tank owner (user) will get to do a series of operations on their tank until they want to quit the program. Here is the menu of choices to present to the user:

1. Buy new fish to add to the tank.
2. A fish dies.
3. A fish has some babies.
4. Print out the current number of fish in the tank.
5. Quit

The tank initially starts with 0 fish.

Present the menu to the user and read in the user's choice. If they enter any integer not in the range of 1 – 5, then reprompt them to enter their choice again.

For choice 1, ask the user to enter how many fish they want to add. Make sure they enter a positive integer. If they don't, reprompt them to enter another integer until they enter a positive integer. Once the user enters a valid integer, print out a message with the following format:

You bought X fish. Now you have a total of Y fish.

where X is the number of fish you bought and Y is the total number of fish in the tank after the purchase.

For choice 2 if there are no fish in the tank, print out the following error message:

There are no fish in the tank to die.

then go back to the main menu. If there is at least one fish in the tank, print out a message of the following format:

One of your fish died. There are X fish left.

where X is the number of fish left.

Choice 3 is very similar to choice 1. Reprompt the user until they enter a positive integer, then output a message with the following format:

Your fish had X new babies. Now you have a total of Y fish.

where X is the total number of new babies and Y is the total number of fish after the babies are born.

For choice 4, print out a message with the following format:

You currently have X fish.

where X is the number of fish you currently have.

For any choice outside of the range 1 – 5, print the following message, followed by reprinting the menu and getting the user's choice again:

Sorry, that is not a valid menu selection.

When the user picks choice 5, print the following message and finish the program:

Thanks for running the fishtank program!

**Sample Program Run (User Input in Bold and Italics)**

Welcome to the fish tank program.

Please select one of these choices.

1. Buy new fish to add to the tank.
2. A fish dies.
3. A fish has some babies.
4. Print out the current number of fish in the tank.
5. Quit

**2**

There are no fish in the tank to die.

Please select one of these choices.

1. Buy new fish to add to the tank.
2. A fish dies.
3. A fish has some babies.
4. Print out the current number of fish in the tank.
5. Quit

**1**

How many fish do you want to buy?

**-2**

Sorry, that is not a valid number. Please try again.

How many fish do you want to buy?

**5**

You bought 5 fish. Now you have a total of 5 fish.

Please select one of these choices.

1. Buy new fish to add to the tank.
2. A fish dies.
3. A fish has some babies.
4. Print out the current number of fish in the tank.
5. Quit

**2**

One of your fish died. There are 4 fish left.

Please select one of these choices.

1. Buy new fish to add to the tank.
2. A fish dies.
3. A fish has some babies.
4. Print out the current number of fish in the tank.
5. Quit

**3**

How many new baby fish were born?

**10**

Your fish had 10 new babies. Now you have a total of 14 fish.

Please select one of these choices.

1. Buy new fish to add to the tank.
2. A fish dies.
3. A fish has some babies.
4. Print out the current number of fish in the tank.
5. Quit

**4**

You currently have 14 fish.

Please select one of these choices.

1. Buy new fish to add to the tank.
2. A fish dies.
3. A fish has some babies.
4. Print out the current number of fish in the tank.
5. Quit

**6**

Sorry, that is not a valid menu selection.

Please select one of these choices.

1. Buy new fish to add to the tank.
2. A fish dies.
3. A fish has some babies.
4. Print out the current number of fish in the tank.
5. Quit

**5**

Thanks for running the fishtank program!

### **Program C: Jumping Frog (frog.c)**

There's a classic puzzle where a frog is stuck in a well below the ground. When the frog jumps up, he jumps up 4 feet, but then slips back down 2 foot. If he starts 20 feet below the top of the well, how many jumps does he take to get out?

Most people answer 10 to this riddle since the frog's progress for one jump is effectively 2 feet. But, if you think carefully, after 8 jumps, the frog has moved 16 feet from the bottom of the well, so he's only 4 feet from the top. When it makes its 9<sup>th</sup> jump, he gets to the top of the well **so it doesn't slip back down 1 foot.**

For this problem, you'll ask the user to enter 3 pieces of input:

1. How far below the top of the well the frog is (in feet).
2. How far the frog jumps up on a single jump (in feet).
3. How far the frog slips down right after a jump (in feet).

Based on these three pieces of input, you'll print out a log of the frog's position (relative to the top of the well.)

#### **Input Specification**

Let **x** be the number of feet the frog starts below the top of the well. **x** is guaranteed to be a positive integer less than 1000.

Let **y** be the number of feet the frog jumps up on a single jump. **y** is guaranteed to be a positive integer less than 1000. Note: **y** could be greater than **x**, in which case the frog gets out in one jump.

Let **z** be the number of feet the frog slips down right after jumping (if the frog hasn't jumped out of the well or to the top of it). **z** is guaranteed to be a positive integer less than **y**.

#### **Output Specification**

For each jump where the frog doesn't get out of the well, print two lines with the following formats:

The frog jumps up **Y** feet and is now **Z** feet from the top.  
The frog slips after jumping and is now **W** feet from the top.

If the frog gets out of the well on a jump, two statements with the following formats:

With a jump of **Y** feet, the frog escapes from the well!  
The frog got out with a total of **J** jumps.

### **Sample Program Run (User Input in Bold and Italics)**

How far, in feet, below the top of the well does the frog start?

**20**

How far, in feet, does the frog jump up on a single jump?

**4**

How far, in feet, does the frog slip after a jump?

**2**

The frog jumps up 4 feet and is now 16 feet from the top.

The frog slips after jumping and is now 18 feet from the top.

The frog jumps up 4 feet and is now 14 feet from the top.

The frog slips after jumping and is now 16 feet from the top.

The frog jumps up 4 feet and is now 12 feet from the top.

The frog slips after jumping and is now 14 feet from the top.

The frog jumps up 4 feet and is now 10 feet from the top.

The frog slips after jumping and is now 12 feet from the top.

The frog jumps up 4 feet and is now 8 feet from the top.

The frog slips after jumping and is now 10 feet from the top.

The frog jumps up 4 feet and is now 6 feet from the top.

The frog slips after jumping and is now 8 feet from the top.

The frog jumps up 4 feet and is now 4 feet from the top.

The frog slips after jumping and is now 6 feet from the top.

The frog jumps up 4 feet and is now 2 feet from the top.

The frog slips after jumping and is now 4 feet from the top.

With a jump of 4 feet, the frog escapes from the well!

The frog got out with a total of 9 jumps.

### **Deliverables**

**Please submit three separate .c files** for your solutions to these problems via WebCourses by the designated due date:

Program A: **olympics2.c**

Program B: **goldfish.c**

Program C: **frog.c**

**Please make sure to include a header comment and internal comments in your code and indent when appropriate.**