

CS 101 - Algorithms & Programming I

Fall 2022 - Lab 4

Due: Week of October 31, 2022

*Remember the **honor code** for your programming assignments.*

*For all labs, your solutions must conform to the CS101 style **guidelines**!*

All data and results should be stored in variables (or constants where appropriate) with meaningful names.

The objective of this lab is to learn how to use a while loop to implement automated repetition. Remember that analyzing your problems and designing them on a piece of paper before starting implementation/coding is always a best practice.

In this particular lab, only **use the while loop**, do *not* use the for or do-while loops.

0. Setup Workspace

Start VSC and open the previously created workspace named `labs_ws`. Now, under the `labs` folder, create a new folder named `lab4`.

In this lab, you are to have four Java classes/files (under `labs/lab4` folder) as described below. A fifth and sixth Java file containing the revisions should go under this folder as well. We expect you to submit a total of 6 files including the revisions. Do *not* upload other/previous lab solutions in your submission. The user inputs in the sample runs are shown in [blue](#).

For all parts of this assignment, you may assume that the user enters valid types and number of values unless stated otherwise (e.g. a positive integer when asked for a positive integer).

1. Number Wizarding

Write a program that takes a positive integer n from the user, inverts it and prints the result. Check for non-integer inputs and non-positive integers as well.

Sample run:

```
Enter input n: 12345
54321
```

2. Decimal to Binary Conversion

Write a program that takes an integer n from the user, converts it to binary, stores it as an integer and prints it out. No need for validating input (assume the user enters an integer).

Sample runs:

```
Please input n: 10
1010
```

```
Please input n: 0103
1100111
```

3. Divide/Multiply by Bit Shifting

There are a few ways of achieving division and multiplication operations in math and “bit shifting” is one of them for any number system. We are going to utilize a property of number systems to divide and multiply by 2 in binary, similar to the decimal representation.

Let's take a binary number 10, which is 0b1010 in the binary number system. Notice that the prefix 0b indicates that the number will be interpreted as a binary number. Bit shifting means shifting all the digits either to left or right. When we shift all the digits to the left, we add “0” to the end of the binary number in Java. When we shift right, we discard the least significant bit (lsb), which is the first bit that has a 2^0 coefficient. Let's see this with an example.

The number is decimal 10 and binary 0b1010.

$$0b1010 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = 8 + 0 + 2 + 0 = 10$$

If we right shift the number, it is going to be 0b101.

$$0b101 = 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 4 + 0 + 1 = 5$$

Check these two questions. What happens if you perform a division by an odd number? What happens if you right shift a decimal number?

Same logic is applicable for the left shift. However, adding 1 or 0 to the right side of a number depends on the compiler you use.

Your task is to write a program that takes a positive integer n from the user and divides or multiplies by multiples of 2 with bit-shifting operations. Your program should ask for user input until the user wants to terminate the mathematical operation task. No need to validate input.

Prompts of your program might be such questions: "Do you want to continue?" or "Do you want to calculate again?". But make sure your prompt is clear and carry the exact intention of yours. For example "Do you want to continue?" is not clear. It is open ended.

Computer programs can be deterministic yet people's behaviors will vary. Possibilities, edge cases will come true in real life applications.

Tip1: There are many coding ways of starting a loop and terminating a loop based on user input; some are more elegant than others. In case you get confused as you're writing your loop, check these out: *loop and a half*, *infinite loop*, *break*, *sentinel*. Remember not to use a *break* statement as per coding guidelines!

Sample run:

```
Enter input n: 10
Please choose the operation. Division or Multiplication 'd' or 'm': m
Enter the multiplier or divisor that is a multiple of two: 8
0b1010000
Do you want to do another operation? 'y' (yes) or 'n' (no): y

Enter input n: 16
Please choose the operation. Division or Multiplication 'd' or 'm': d
Enter the multiplier or divisor that is a multiple of two: 4
0b100
Do you want to do another operation? 'y' (yes) or 'n' (no): n

Program finished.
```

4. Busy Euler

Busy Euler was borned in the 20's. At one weekend, her friends called her and now, somehow, she has to go out with her friends for a coffee. Before that, she needs to calculate the area under curves for her math homework. The problem is: she doesn't know how to integrate. But she knows how to write while loops. The homework is about calculating the area under the function $f(x) = x^2$, from $x = 0$ to $x = 1$. Linear or maybe parabolic functions are easy but the next questions of her homework contain complex functions that make it hard to approximate with simple solutions. She planned to generalize the solution and decided to divide the problem into sub-problems. Therefore, she created small rectangles and as she increased the number of rectangles her approximation became better and better (see Figure below).

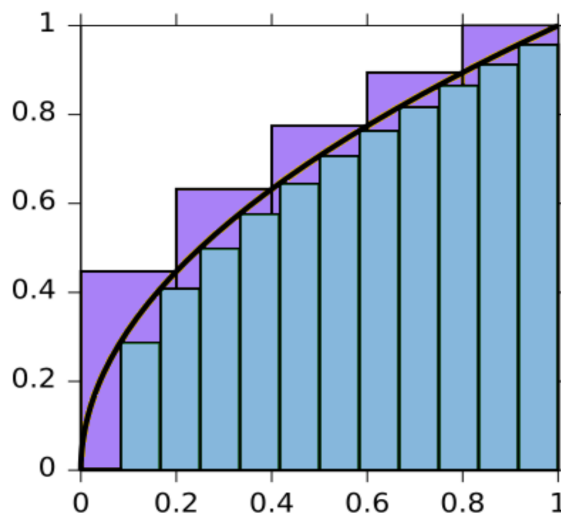


Figure 1 Approximations to integral of \sqrt{x} from 0 to, en.wikipedia.org, [integral approximations](#).

Your task is to calculate the area under the function $f(x) = \sin(\pi * x) + \frac{\sin(3*\pi*x)}{3}$. Below, you can find the initializations she did in her code. You can use them in yours.

```
x_start      : starting point on the x axis
x_end        : ending point on the x axis
step_size    : width of the rectangles
total_area = 0 : initial total area
```

(You are supposed to “hard code” the calculation of your specific function and the string “The function is: x^2 ” argument in your print method.)

Sample run:

```
The function is: x^2
Parameters:
x_start = 0, x_end = 1
step_size = 0.01
Area under "x^2" from 0 to 1: 0.33835000000000004
```

Sample run:

```
The function is: sin(PI*x) + sin(PI*3*x)/3
Parameters:
x_start = 0, x_end = 1
step_size = 0.01
Area under "sin(PI*x) + sin(PI*3*x)/3" from 0 to 1: 0.7072505742610112
```

You can use the integral notation of [Desmos](#) to check your answers by solving them analytically. It will visualize and at the same time calculate the answer. People are also making art out of linear equations $f(x) = ax + b$. You can check an example [here](#).

“It is important to draw wisdom from many different places. If you take it from only one place, it becomes rigid and stale.”
Uncle Iroh

“Boomerang! You do always come back!”
Sokka