CSC 120 Midterm

This is an open book exam. What this means is you can refer to online material, book for this exam. However, to get all the credits please make sure that you cite the references for every answer. You can cite the webpage url or the text book name and chapter details as a reference. The aim of this assignment is to help you become self learners. It will test your understanding of basics and how you apply the basics to new scenarios. We are moving away from a learning style which is only memory based and trying to develop skills to learn any new skill by using all the resources available. Make sure you attempt everything as there are points for attempting. Good luck!

**Part A Conceptual Questions**

* (+2) In an 8 bit number 10101000. What is the MSB (Most Significant Bit)? Which bit is the LSB (Least Significant Bit)

In the 8-bit number above, the left most four bits would be the MSB (1010) and the LSB would be the right most four bits (1000).

<https://bit-calculator.com/most-and-least-significant-bit>

* (+2) In the 2's complement system, if there is a number 1000, what does the 1 represent?

The “1” in the four-bit pattern above will represent the “sign” (negative or positive) of the pattern. “1000” would be a negative value, as a “1” in the left most bit is always representative of a negative, while a “0” would be positive.

https://reader.yuzu.com/#/books/9780134875583/cfi/6/44!/4/2/6/8/2@0:1.07(Twos Complement Notation)

* (+6) Convert the following 2's complement to its equivalent base 10 form (decimal)

00011 3

11100 -4

11010 -6

<https://reader.yuzu.com/#/books/9780134875583/cfi/6/44!null@0:0> (Twos Complement Notation)

* (+4) Why is the 2's complement system used?

The two’s complement system simplifies the arithmetical operations between positive and negative (or two negative) bit strings. With this system, positive and negative strings can be computed more easily because the hardware ignores the left most bit and recognizes it as the sign bit.

<https://en.wikipedia.org/wiki/Two's_complement#Why_it_works>

<https://stackoverflow.com/questions/1125304/why-prefer-twos-complement-over-sign-and-magnitude-for-signed-numbers>

* (+6) What is the difference between signed binary integers and unsigned binary integers? When would you use each? If you had a 6 bit binary number how many decimal numbers can be represented with it if you were using a signed binary?

In a signed binary number, the left most bit in the string is going to be representative of whether or not the string is positive or negative, in unsigned, there is nothing that notates whether the number is positive or negative. Unsigned bit patterns are just considered to be typical binary numbers in terms or representation. You would use signed binary if you needed to represent a combination of negative and positive integers (i.e. two’s complement). In a signed binary 6-bit pattern, you can represent up to 62 decimal numbers (different integers), 31 down to -31.

<https://stackoverflow.com/questions/42603639/what-is-the-difference-between-signed-and-unsigned-binary>

<https://www.tutorialspoint.com/unsigned-and-signed-binary-numbers>

* (+5) You have an 8 bit binary number 11110000. What happens when you AND this number with any other 8 bit number. Try out a few examples and try to generalize your observations.

No matter what the second 8 bit number is, the last four digits will always remain “0000” (or false). Unless there are two corresponding “1” (true) in the same positions per string, the AND operation produces an outcome of false.

<https://reader.yuzu.com/#/books/9780134875583/cfi/6/34!/4/2/6/4/6/2/2@0:0>

(Boolean Operations)

* (+5) You have an 8 bit binary number 00001111. What happens when you OR this number with any other 8 bit number. Try out a few examples and try to generalize your observations.

In an OR operation with this string, the last 4 digits will always produce a “1” (true) no matter what they are added two. Unless there are two corresponding “0” in the operation, the outcome produce will be true (1)

<https://reader.yuzu.com/#/books/9780134875583/cfi/6/34!/4/2/6/4/6/2/2@0:0>

(Boolean Operations)

* (+5) You have an 8 bit binary number 11111111. What happens when you XOR this number with any other 8 bit number. Try out a few examples and try to generalize your observations.

XOR produces an outcome of true (1) as long as only one of the inputs is true to begin. With an 8 bit string of “11111111” it truly depends on what the other string is. If you XOR with a string of “00000000” the outcome will be the same as the original (11111111). If you use a string of another 8 ones, the output would be 00000000 as every bit XOR in both strings would be true and would output false.

<https://reader.yuzu.com/#/books/9780134875583/cfi/6/34!/4/2/6/4/6/2/2@0:0>

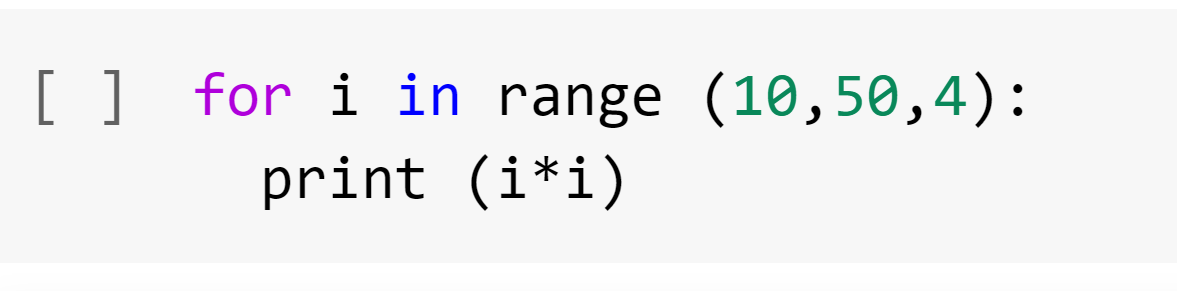
(Boolean Operations)

* (+5) Write a Python program to add two numbers by getting inputs from the user. Paste a screenshot of the code from Google colab.

Graphical user interface, text, application

Description automatically generated

* (+10) In Python, consider the following code sample,



1. What does 10 represent?
2. What does 50 represent?
3. What does 4 represent?
4. What gets printed?

1 the number 10 represents the initialization number, or the starting point integer.

2 50 represents the stopping point integer, the loop will finish printing before it hits this number.

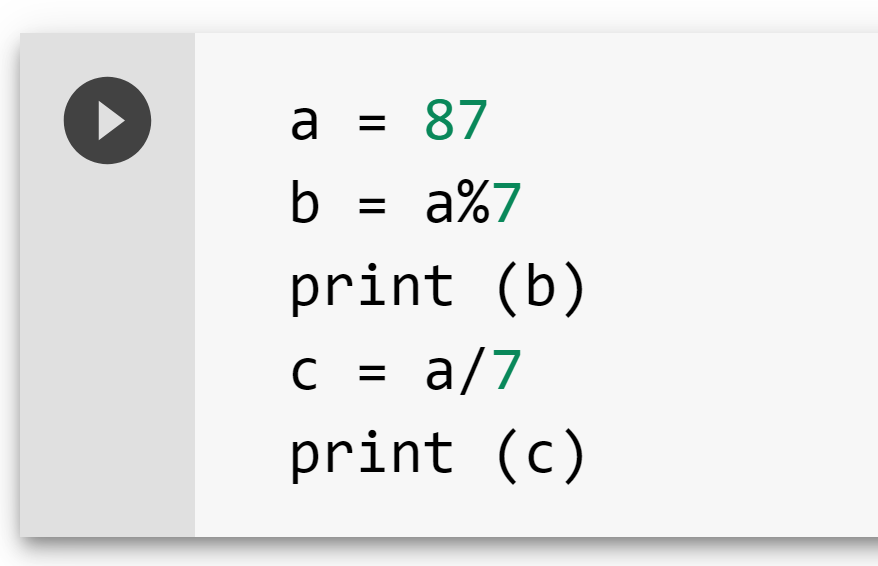
3 the 4 represents the specific increment in which the loop is printed. In this code, every fourth number is printed.

4 when this is run, the code prints each integer squared (10x10, 14x14, 18x18 etc.)

<https://www.w3schools.com/python/ref_func_range.asp>

<https://www.w3schools.com/python/ref_func_range.asp>

* (+5) In Python consider the code below what value would be printed by the following operation. Why?



When the code is run altogether, it will print two distinct values based on the specific syntax defined. The modulator (%) in b will force print the remainder 87 divided by 7, which is 3. Underneath this, it will print a floating-point number of ﻿12.428571428571429 since the syntax used (/) calls for division without the use of a remainder. The differentiation in the syntax (% opposed to /) is what causes the difference between the float and the integer.

<https://www.programiz.com/python-programming/operators> (Arithmetic Operators)

<https://www.w3schools.com/python/python_operators.asp>

* (+5) In the VOLE architecture describe a sample instruction in terms of an opcode and an operand with an example.

0x1 0x7 0xB 0x4 17B4

The red highlighted portion is the operation code (op-code) that contains the instruction for the operation to be performed (i.e. LOAD, STORE, MOVE, JUMP). The green and blue portions are the operand, which provides further details on the operation being performed. In this example, the green section contains the register in which the contents of the memory cell highlighted in blue will be loaded into. This example states that the contents of memory cell B4 will be loaded into register 7. The pink portion is a shorthand expression of the same instruction.

<https://reader.yuzu.com/#/books/9780134875583/cfi/6/64!/4/2/14/6/2@0:15.6> (Vole: An Illustrative Machine Language)

* (+10) What is an IO bound process vs a compute bound process? In case an IO bound process is waiting on an operation what happens? What is the role of the kernel and the scheduler in this event? Write your own interpretation by reading the book/online. Cite references.

A compute bound process is a process run by and determined by the speed of the central processor. An IO (input/output) bound processes’ speed is determined on the time spent waiting for the IO operations to complete. If an IO process is waiting on an operation, they will sit idle and ready to be run until the other process is finished. The scheduler, located within the internal portion of the operating system called the kernel, will make note of the processes waiting to start as well as the process already active, and once the active process is complete, will remove the completed process from the queue and being the next

<https://reader.yuzu.com/#/books/9780134875583/cfi/6/86!/4/2/8/6/2@0:100>

<https://reader.yuzu.com/#/books/9780134875583/cfi/6/88!/4/2/8/4/2@0:100>

<https://en.wikipedia.org/wiki/I/O_bound> (Comparison with CPU bound)

* (+5) How slow is main memory compared to CPU register? How slow is fetching from a hard drive as compared to main memory? Research online and provide references.

The CPU register is faster than main memory. CPU registers are smaller, faster storage units located within the CPU itself. Fetching from a hard drive will be slower than fetching from main memory although the storage capacity of a hard drive will be far greater than that of main memory.

<https://www.enterprisestorageforum.com/hardware/memory-vs-storage/>

<https://pediaa.com/difference-between-register-and-main-memory/>

**Part B Exploration (15 points)**

In Python, when do you use a for loop vs a while loop? Provide examples of each. Research online on when a while loop is more suitable to use than a for loop. Cite your sources as references.

Deciding between using a for loop and a while loop is ultimately dependent on what seems more appropriate for the task. Often times, for loops can be written as while loops, and vice versa.

The general rule of thumb for choosing between the two is whether or not the the number of iterations to loop through is known ahead of time. If the number of iterations is known ahead of time, then use the for loop, else, use the while loop to iterate until a given condition is broken.

For example, use a **for loop** when you know the loop should be executed *n* times. Use a **while loop** that requires user input, like a game of tic-tac-toe, that will break until the program is over (board is filled or player wins)

<https://betterprogramming.pub/how-to-pick-between-a-while-and-for-loop-14ef217c3776>

**Part C General Awareness (10 points)**

**Read the following article.**

**https://uhs.princeton.edu/health-resources/ergonomics-computer-use**

What guidelines are you currently following? What guidelines are you not following. List all changes that you plan to make over the next few weeks to improve your posture.

For years, I worked as a personal trainer, working closely with physical therapists and physicians (as clients and colleagues) and learned a great deal about much of the postural queues mentioned in this article. While seated or standing, I always make sure that I am not slouching, that my shoulders are back, and my neck and spine are in a neutral position as much as possible. Most of this is done through engagement of one’s core muscles and is a technique I have been practicing for years and working on with clients who sit at desks for long periods. I also try to sleep in a neutral position, using somewhat firm pillows between my arms and legs to keep my spine in alignment Unfortunately, because of all of the weightlifting I did as a trainer, I already have various RSI symptoms in my neck, upper back/shoulders, and wrists, so it is important that I do my best to take breaks every thirty minutes and do some basic stretches and movement exercises that fit my areas of issue (also giving my eyes needed rest, though I try to wear my blue light filtering glasses as well). I also try to do mild strength training as much as possible (3-4x per week) focusing especially on more core and posterior chain exercises, muscles that directly affect posture.

One thing I know I need to fix quickly is my work chair. Right now, I use a backless stool that is non-adjustable and too high for my desk. I plan to buy something more ergonomic as soon as possible. I also need a more comfortable keyboard and mouse combination as the laptop I am using is not ideal for correct typing posture. I have not done much typing in the last ten years, I am slow, therefore, I spend more time actually typing than the average person with some experience. If I do not make some of these simple typing related changes soon, I worry that issues I already have will become exacerbated quickly. I have been practicing typing with better form to unlearn bad habits that I have developed over the years. A lot of these things are very new to me, but many are not. I am optimistic that once I make some of the changes above, I will be able to work more comfortably and efficiently than ever before.

Instructions: Upload the file with your firstname\_lastname on Blackboard.