Problem 1: Introduction – circle the correct answer [3 marks]

i. Select which of the following answers has the terms from the types of memory in order from Smallest to Largest Capacity? Sample question, modified slightly

   a. Disc Storage, RAM, Cache, CPU Registers

   b. CPU Registers, RAM, Disc Storage, Cache

   c. RAM, Disc Storage, CPU Registers, Cache

   d. RAM, CPU Registers, Cache, Disc Storage

   e. CPU Registers, Cache, RAM, Disc Storage

ii. Microsoft Windows is an example of:

   a. Hardware

   b. Operating System

   c. Application

iii. Computational thinking (the concept, not the course) requires a computer

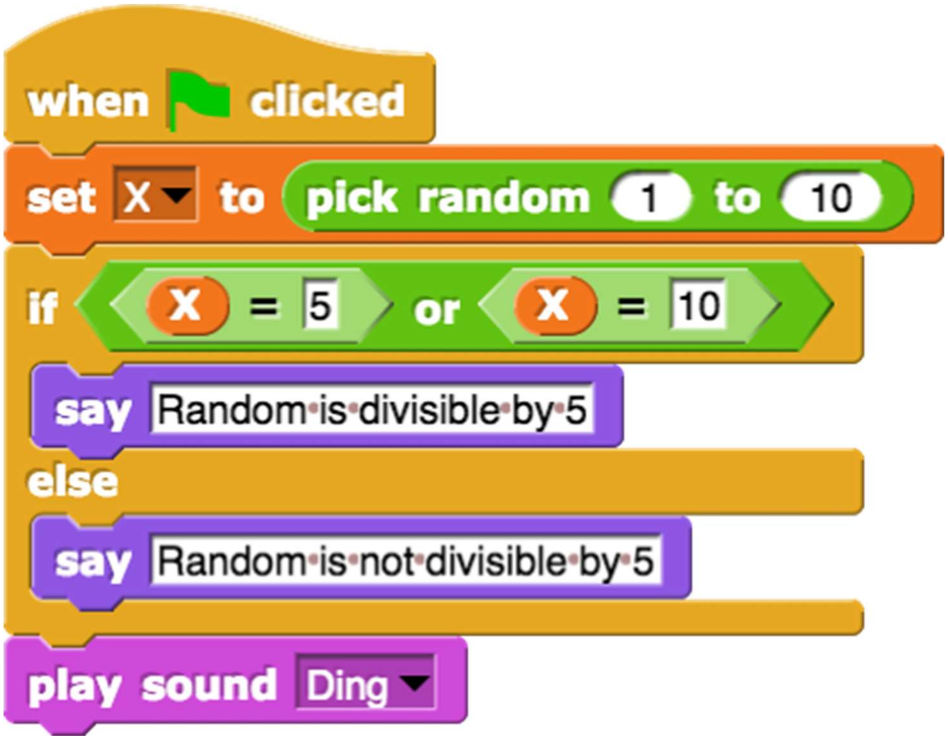
   a. True

   b. False

Problem 2: Programming with Snap! [2 marks] (practice snap question)

Consider the following code: Note that 

chooses a random number from 1 to 10.

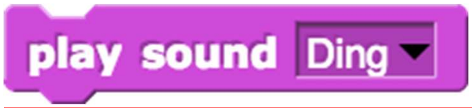


For what values of X does "ding" play? Why?

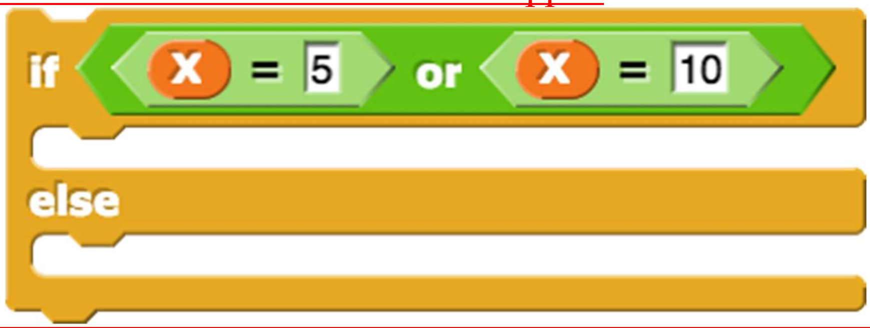
Grading: one point for answer, one point for explanation

Always

Explanation



is not wrapped within

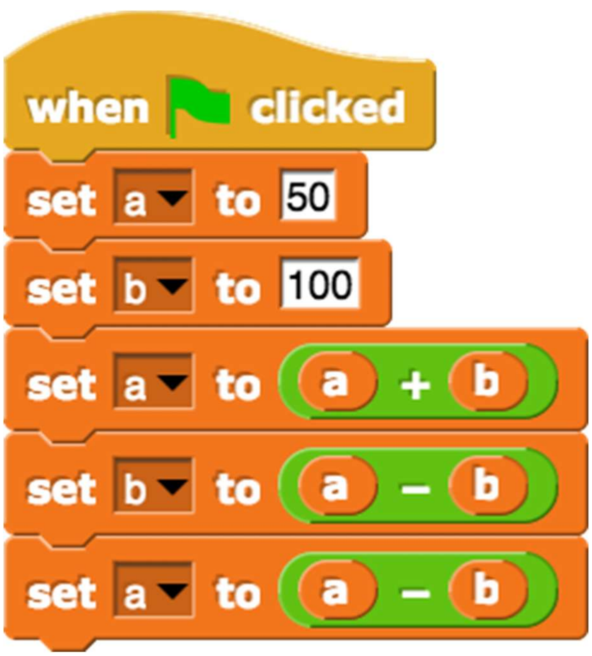


so regardless of whether or not

is true,

will execute.

Problem 3: Fun with variables [4 marks] (code from practice snap questions)



a. [1 mark] What is the value of "a" after this code is run?

100

b. [1 mark] What is the value of “b” after this code is run?

50

c. [2 marks] In general terms, what does this part of the code do:

It swaps the values of two variables without requiring additional swap space
1 point for swap
1 point for mentioning that it doesn’t take up additional space

Problem 4: Sorting Socks [6 marks]

Assume:

- Any two socks of the same colour match (e.g., all black socks match all other black socks, and you have no special socks for the right or left foot)
- You know the number of socks of each colour when you start.
- All socks have a match (no laundry eating socks for you!)
- Your socks initially start out in a big pile. You can only look at one sock at a time.

Algorithm A:

1. Create one pile for each colour of socks
2. For each sock
    a. Look at each pile until you have found the right colour for it
    b. Place the sock in the pile
3. Match up pairs within each pile. Each time you find a match, ball up the pair and throw it in your sock drawer

Algorithm B:

1. Pick up a sock, call it Sock 1
2. Until there are no socks left in the pile
    a. For each sock in the sock pile (call it Sock 2), see if it matches Sock 1
        i. If it matches
            1. Ball up the pair and throw it in your sock drawer
            2. take the next sock on the pile as Sock 1

           else

                Put Sock 2 in the bottom of the pile

For **space** comparison, count the number of distinct spaces that you need to have before the socks go in the sock drawer (the sock drawer is magical and takes no space). Every sock in a pile takes up one space, e.g., a pile of 4 black socks takes up 4 spaces. Be sure to consider your hands as spaces for balling up the socks along with as many swap spaces as you need. You may reuse your hands as needed as long as they are not otherwise occupied.

For **time** comparison, count the number of times that you have to touch each sock.

For example, consider a pile with two black socks on top and then two white socks.

TIME: Algorithm A would result in the following number of times you touch the socks.

1. Pick up the first black sock and put it on the black sock pile
2. Pick up the second black sock and put it on the black sock pile
3. Pick up the first white sock and put it on the white sock pile
4. Pick up the second white sock and put it on the white sock pile
5. Pick up the first black sock from the black sock pile
6. Pick up the second black sock from the black sock pile. Ball up the black socks and throw them in the drawer
7. Pick up the first white sock from the white sock pile
8. Pick up the second white sock from the white sock pile. Ball up the white socks and throw them in the drawer

Thus Algorithm A has a time cost of 8 in this scenario.

Algorithm B would result in the following number of times you touch the socks

1. Pick up the first black sock and call it Sock 1
2. Pick up the second black sock and call it Sock 2. Ball up the block socks and throw them in the drawer
3. Pick up the first white sock and call it Sock 1.
4. Pick up the second white sock and call it Sock 2. Ball up the white socks and throw them in the drawer

Thus Algorithm B has a time cost of 4 in this scenario

a. [2 marks] What is the minimum amount of space needed for algorithm A to do the above scenario where there is a pile of two black socks followed by two white socks? Why?

You would need to have 4 slots for each sock in the pile plus four slots for the socks on the black and white socks pile plus 2 slots for holding the socks in your hands. = 10 slots

b. [2 marks] What is the minimum amount of space needed for algorithm B to do the above scenario where there is a pile of two black socks followed by two white socks? Why?

You would need to have 4 slots for the socks in the pile plus 2 slots for the socks in your hand. You can get away without swap space for the pile because you never have two socks in your hands at the same time that you need to reorder the pile.

c. [2 marks] In general, which algorithm would you expect to take the least amount of *time*? Why?

Algorithm A would in general take the least amount of time because you may have to cycle through the socks a large number of times in order to find all the matches.

Note: sorting the socks once is much, much more time efficient than repeatedly going through a loop of many socks. So this line of reasoning was incorrect.

Problem 5: Internet [2 marks]

a. [1 mark] Which of the following webpages is more likely to be related to
http://www.ugrad.cs.ubc.ca/~cs100/slides/index.html?

a. http://www.ugrad.cs.ubc.ca/~cs100/slides/final/index.html

b. http://www.ugrad.cs.ubc.ca/~cs110/slides/index.html

b. [1 mark] Consider the following webpage: http://cs.ubc.ca/students/undergrad/courses/core-curriculum
[practice question]

Rearrange both the domain name and file directory information from the highest (most general) level to lowest (most specific) level.

a. cs → ubc → ca → students → undergrad → courses → core-curriculum

b. core-curriculum → courses → undergrad → students → ca → ubc → cs

c. ca → ubc → cs → students → undergrad → courses → core-curriculum

d. cs → ubc → ca → core-curriculm → courses → undergrad → students

e. students → undergrad → courses → core-curriculum → cs → ubc → ca

Problem 6: Algorithms and fairness [5 marks]

i. [1 mark] When building a classifier, if the training data is biased in some way then

   a. The test data is biased in the opposite way

   b. The test data is biased in the same way

   c. The test data is unbiased

   d. The biasing of the test data and the training data is not related

ii. [2 marks] Consider the case where Google allows companies to pay to be advertised. List one way that this could cause conscious bias. List one way where this could cause unconscious bias.

A way that this could cause conscious bias is that Google could choose to only pay companies that they deem to be suitable for advertising, ones that share the same values that Google does. A way where this could cause unconscious bias is that Google may be unconsciously charging higher prices or denying advertising requests just based on the name of the advertiser or company, for example, if they are of an ethnic minority.

iii. [2 marks] Consider programmers at Amazon writing code to decide where to offer one day shipping. Using terms discussed in class and in the readings, state why the programmers may want to consider the racial demographics of those who are in areas they are considering.

Programmers may want to consider the racial demographics of those who are in areas they are considering as they may be affected by unconscious bias, and may design a biased algorithm, which for example, would only offer one day shipping in areas with fewer ethnic minorities. By considering the racial demographics, they could ensure that their algorithm is fair.

Note that saying that people may want to avoid delivering to risky areas because people of certain racial minorities tend to commit more crime is NOT a valid answer. That is NOT something that it is appropriate for the programmers to consider as it would disproportionally cause harm. Other examples of why one shouldn't advertise the service to racial minorities are similarly incorrect.

For more on the subject, read: https://www.bloomberg.com/graphics/2016-amazon-same-day/