15-110: Principles of Computing, Spring 2018

Problem Set 8 (PS8)

Due: Friday, March 30 by 2:30PM via Gradescope Hand-in

HANDIN INSTRUCTIONS

Download a copy of this PDF file. You have two ways to fill in your answers:

1. **Just edit (preferred) - Use any PDF editor (e.g., Preview on Mac, iAnnotate on mobile, Acrobat Pro on pretty much anything) to typeset your answers in the given spaces. You can even draw pictures or take a picture of a drawing and import it in the correct place in the document. That's it. (Acrobat Pro is available on all cluster machines.)**

2. **Print and Scan - Alternatively, print this file, write your answers neatly by hand, and then scan it into a PDF file. This is labor-intensive and must be done by the deadline.**

Once you have prepared your submission, submit it on Gradescope. A link to Gradescope is provided in our Canvas course portal.

Fill in your answers ONLY in the spaces provided. Any answers entered outside of the spaces provided may not be graded. Do not add additional pages. We will only score answers in the given answer spaces provided. If we cannot read your answer or it contains ambiguous information, you will not receive credit for that answer.

Be sure to enter your full name below along with your section letter (A, B, C, etc.) and your Andrew ID. Submit your work on Gradescope by 2:30PM on the Friday given above.

REMINDER: Sharing your answers with another student who is completing the assignment, even in another semester, is a violation of the academic integrity policies of this course. Please keep these answers to yourself.

Name (First Last) ____

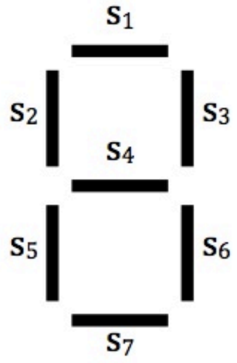
Section __________ Andrew ID ____________________________________

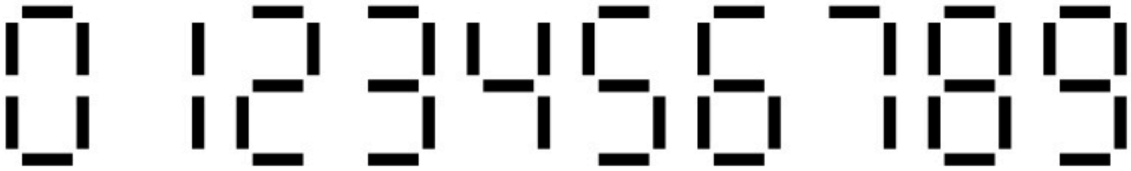
1. (2 pts) Another way to represent unsigned integers is to use a representation called Binary Coded Decimal. In this representation, each decimal digit of an integer is represented using 4 bits as follows:

0	1	2	3	4	5	6	7	8	9
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

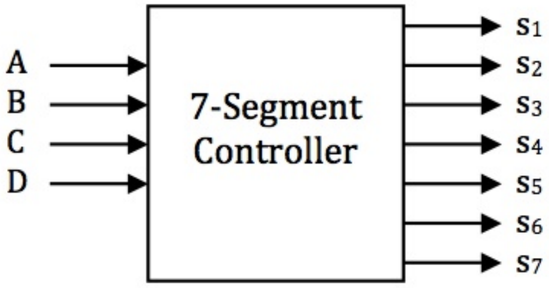
Let ABCD represent a 4-bit binary-coded decimal number as described above. For example, the decimal digit 7 is represented in binary-coded decimal as 0111, so in this case, A=0, B=1, C=1, and D=1.

A seven-segment display can be used to display each of ten decimal digits as shown below.

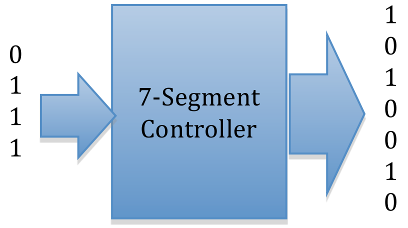




We can define a logic circuit abstractly that requires four Boolean inputs A, B, C and D representing a binary-coded decimal digit, and seven Boolean outputs s_1, s_2, …, s_7 to control whether each segment is on or off to display the number represented by ABCD. A segment s_i is lit if s_i is 1 (true) and not lit if s_i is 0 (false).



For example, if you feed in the binary-coded value for 7 (0111), you would get this output, which controls the display:



(continued on next page)

(a) Derive a Boolean formula for s_2 that is true (1) if and only if segment s_2 is lit given the input ABCD. Your answer should be of the following form:

s_2 = (___________) ∨ (____________) ∨ (_____________) ∨ …

where each missing section is a Boolean expression with all 4 input variables that is true when ABCD represents a decimal digit that results in the segment s_2 of the display being lit. To help you get started, the decimal digit 0 (binary coded 0000) will light segment s_2, so the first expression in the formula is (¬A ∧ ¬B ∧ ¬C ∧ ¬D) since s_2 would be true (i.e. 1) if A=0, B=0, C=0, and D=0. You will need to find all of the other missing expressions for s_2. (You can use the – sign for "not" and the ^ sign for "and" to make typing simpler if you wish.)

S_2 = (¬A ∧ ¬B ∧ ¬C ∧ ¬D) ∨ ____

____

(b) Derive a very simple Boolean formula for s_3 that is true (i.e. 1) if and only if the segment s_3 is lit. (HINT: Segment s_3 is lit for all decimal digits EXCEPT when the decimal digit is 5 or 6.)

S_3 = ____

2. (1 pt) When a programmer writes a program in a high-level language (e.g. Python), an interpreter or compiler is needed in order to execute the program on a computer. Explain why in one or two sentences.

3. (2 pts) This question deals with generating random numbers using a linear congruential generator.

(a) A linear congruential generator is created with a = 11, c = 5 and m = 16. What is the period of this random number generator if it starts with a seed (initial value) of 1? What are the "random" values in this period?

(b) Consider another random number generator with a = 81, c = 337, and m = 1000. Here is a short program that can generate a "random" list of integers.

```
def prng():
     seq = [344]
     for i in range(1, 20):
          seq.append( (81*seq[-1] + 337) % 1000 )
     return seq
```

When we use the generator to create a list of 20 numbers, we get this list:

```
[344, 201, 618, 395, 332, 229, 886, 103, 680, 417, 114, 571,
588, 965, 502, 999, 256, 73, 250, 587]
```

What is the pattern in this sequence that would make you question whether you want to use this generator for random numbers? (HINT: Look at the last digit of each number across the sequence. Your answer should explain how you can predict exactly the last digit of the next value after this sequence.)

4. (2 pts) Use **only** the Python random function `randrange` to solve each of the following tasks. Do not use any other random number functions in your solutions. You may assume `random` is imported.

(a) Write a Python function `spin1()` that returns an integer and simulates the wheel spinner from the Milton Bradley game of Life. The wheel contains the numbers from 1 to 10.

```
def spin1():
```

(b) Write a Python function `spin2()` that returns an integer and simulates the Showcase Showdown wheel from The Price Is Right game show. The wheel contains the multiples of 5 from 5 to 100. Inclusive.

```
def spin2():
```

(c) Write a Python function `roll1()` that returns an integer and simulates a backgammon doubling cube. The 6-sided cube contains the values 2, 4, 8, 16, 32, and 64.

```
def roll1():
```

(d) Write a Python function `roll2()` that returns a string and simulates a poker die. The 6-sided cube contains the values "A", "K", "Q", "J", "10" and "9". (You should simulate just one die.)

```
def roll2():
```

5. (1.5 pts) Consider the following code written for a simple dice game. In this game, if the player does not roll "doubles", then the player adds the sum of the dice to their total. If the player rolls "doubles", the player's total gets reset back to 0. At the end of the game, the player wins the number of total points accumulated after 10 rolls. There is a logical error in the loop of this program below. Explain what is wrong in this computation clearly and concisely, and then rewrite the loop to correct this error.

```
from random import *

def roll1():
    return randint(1,6)     # simulate a roll of a six-sided die

def roll2():
    return randint(1,6)     # simulate a roll of a six-sided die

def simple_game():
    total = 0
    for i in range(1,11):
        if roll1() != roll2():
            total = total + roll1() + roll2()
        else:
            total = 0
    return total
```

Logical error in loop:

Corrected loop:

```
for i in range(1,11):
```

6. (1.5 pts) Recall that a deck of cards is represented as a list of lists. Each card is a list of two strings, the rank followed by the suit. A deck is made up of 52 cards. For this problem, you may assume that the deck is shuffled.

(a) Complete the function below that returns (but does not remove) a random card from the second half of the deck. Your function should be only one line of code.

```
from random import *

def random_card(deck):

    return ______________________________________________
```

(b) Let `hand` be a list of five cards that are sorted by rank. A player has a "four of a kind" if the player has four cards that have the same rank. Complete the following function that returns `True` if the player has a four-of-a-kind or `False` if not. Think carefully, there are only two cases where a four-of-a-kind can happen. For this solution, do not use any helper function like `get_rank`. You can solve the problem directly with the given hand.

HINT FOR A SHORTER ANSWER:
In Python, if you want to test if `a == b and b == c`, you can write the shortcut test `a == b == c` to do the same thing.

```
def four_of_a_kind(hand):

    if ________________________________________________________

    __________________________________________________________:

        return True

    if ________________________________________________________

    __________________________________________________________:

        return True

    return False
```