Student Id: Name: Signature:

CSE 221 - Principles of Logic Design - 2021 Fall
Homework2

You will use a "**key**" to solve the questions in this homework. The key is the number formed by the last two digits of your student id. **If your key is less than 20, add 20 to it!** (*So for example, if your student id was 201007020**47** your key would be "**47**"; if your student id was 201007020**08** your key would be "**28**".*) *We will assume that the key is 47* while explaining the questions (you should replace it with your own key).

So, first write down your id and key: **Student Id: . . . . Key: . . . . . .**

1) Write your first name in lowercase letters (*Example: ali*). (If you have two names, use only the first one; for Turkish characters use the corresponding English characters: ş->s, ğ->g etc.) Find the ASCII code for your name; use hexadecimal notation. (*ali: 61 6C 69*) Then, do the same with your name in capital letters. (*ALI: 41 4C 49*)

Name (lowercase): . . . . ASCII Code: . . . .
Name (CAPITAL): . . . . ASCII Code: . . . .

2) m = (key mod 6) + 3 (*m=(47 mod 6) + 3 =5 + 3 = 8*) If m is odd, add 1 to it. (*m = 8*)
Find your m. **m = ... mod 6 + 3 = ... (if odd, then add 1 -> m = ...)**
Design a <u>self-complementing</u> coding system to represent the digits in base "*m*" numbering system. For a code system to be "self-complementing", the number obtained by complementing each bit of the original code of a number should be equal to r-1's complement of the original number.
(*For example, for base 4 numbering system, let's say digit 1 has the code "001" and digit 2 has the code "110". Then the code for $(11)_4$ is "001001". When we complement each bit, we obtain "110110" and this is the code for $(22)_4$. 22 is the 3's complement of 11.*)

Write down the codes for your digits on the table below. On the right of the table, give a partial proof that your code is self complementing: Take a 3-digit number in base m as an example, write down its code, then complement this code and show that it is m-1's complement of the original number.

Digit	Code
0	
1	

Example 3-digit number p:

Code of p (C_p):

Complemented Code C_r:

C_r is the code of (r) :

p + r =

3) Write down your key again. **Key: . . . . . .**
Write your key in BCD code: . . . .
Write your key in 84-2-1 code: . . . .

m = 3*key **m: . . . . . .**
Write m in BCD code: . . . .
Write m in Excess-3 code: . . . .

4) Some terms are listed in the table below.

Term #	Term	Term #	Term	Term #	Term	Term #	Term
0 (t_0)	xyz	4	xy	8	x'+y'+z	12	x+y
1	x'yz	5	xz	9	x+y+z'	13	x+z
2	xyz'	6	yz	10	x+y'+z'	14	y+z
3	x'yz'	7	x'y	11	x+y+z	15	x+y'

m = key mod 3 (*m=47 mod 3=2*). Find your m. **m = ... mod 3 = ...**
n = key mod 5 (*n=47 mod 5=2*). Find your n. **n = ... mod 5 = ...**
p = key mod 8 (*p=47 mod 8=7*). Find your p. **p = ... mod 8 = ...**
$F_1 = t_m + t_n + t_p$ ($F_1 = t_2 + t_2 + t_7 = xyz' + x'y$)
Find your F_1. $F_1 = t_{...} + t_{...} + t_{...} =$
Using algebraic methods, simplify F_1 to minimum number of literals.

$F_2 = t_{m+8} \wedge t_{n+8} \wedge t_{p+8}$ ($F_1 = t_{10} \wedge t_{10} \wedge t_{15} = (x+y'+z')(\ x+y')$) --- $\wedge$: AND ---
Find your F_2. $F_2 = t_{...} \wedge t_{...} \wedge t_{...} =$
Using algebraic methods, simplify F_2 to minimum number of literals.

5) m = key mod 8 (*m=47 mod 8=7*). Find your m. **m = ... mod 8 = ...**
n is the binary equivalent of m (*n=7=111_2*) **n: . . . .**
Design a gray code, for the digits of octal (base 8) numbering system. The code for digit "0" should be "n". (*Code of digit "0": 111*) Keep in mind that the codes of "0" and "7" should also have a 1-bit difference (only one bit should change between the two codes).
Write down the codes for your digits on the table below.

Digit	Code
0	
1	
2	

6) a) Find the complement of the simplified version of F_1 in question 4 using DeMorgan's rule and algebraic methods.

b) Draw the truth table of the simplified version of F_1 in question 4. Obtain the truth table of the complement of this function (add a column). Simplify this complement from its truth table using algebraic methods. Did you end up with the same expression in part a? Do these two methods always result in identical expressions? Explain…