**Applications of Cryptography CSCE 4050/5050 (Spring 2025)**

**Homework 1**

1. [**Modular arithmetic**] Compute the following by hand (without calculator/computer)   
   and show your computation:
2. 25 + 15 = 14 (mod 26) which is in the form of a = b (mod N)

Here, 25 + 15 = 40 (mod 26)

40 – 14 = 26 = 1 ⋅ 26 which is in the form of a – b = r ⋅ N

I.e. a = 40, b= 14, r = 1, N = 26

1. 8 – 20 = 14 (mod 26) which is in the form of a = b (mod N)

Here, 8 – 20 = -12 (mod 26)

-12 – 14 = -26 = -1 ⋅ 26 which is in the form of a – b = r ⋅ N

i.e. a = -12, b = 14, r = -1, N=26

1. [**Modular arithmetic: implementation**] Write a program which implements a modular addition and a modular subtraction. For simplicity, fix the modulus to be 26. A modular addition will be implemented as a function m\_add(a,b) which take integers “a” and “b” as inputs and outputs   
   a + b (mod 26). For simplicity, you may assume that a,b ∈ {0, 1, …, 25}. Similarly, a function m\_sub(a,b) will implement a - b (mod 26). Your submission must include the code of your program (enclosed as a separate file) as well as screenshots (included into the homework report) showing test runs for the values from Question 1 items (b) and (c).

**Here is the screenshot of the output of the program:**

A screenshot of a computer program

Description automatically generated

1. [**Caesar cipher: implementation**] Write a program which implements the Caesar cipher. It is required that the encryption and decryption algorithms in your implementation will use the functions implemented in Question 2. Your submission must include the code of your program as well as screenshots showing three test runs for the plaintext “back” and the values of key 3, 7, 10.

**Here is the screenshot of output of the program:**

A screenshot of a computer program

Description automatically generated

1. [**Exclusive-OR using modular arithmetic**] Let us consider a representation of the Exclusive-OR (XOR) function using modular arithmetic. Compute the following:
2. 1 + 1 = 0 (mod 2)
3. 1 + 0 = 1 (mod 2)
4. 0 + 1 = 1 (mod 2)
5. 0 + 0 = 0 (mod 2)
6. 2 + 4 = 0 (mod 2)

Truth table for XOR function

|  |  |  |
| --- | --- | --- |
| a | b | a + b |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

The above results match the XOR truth table which confirms that XOR is equivalent to addition modulo 2.

**Note:** Compare the items (a)-(d) with the truth table of XOR to confirm that the latter is actually an addition modulo 2.

1. [**More** **modular arithmetic**] Compute the following by hand (without calculator/computer)   
   and show your computation:
2. 5 ⋅ 8 = 7 (mod 11)

5 ⋅ 8 mod 11 = 40 mod 11 = 7 + 3 ⋅ 11 mod 11 = 7 mod 11

**Answer**: 7

1. 28 ⋅ 35 = 1 (mod 11)

Or, 28 ⋅ 35 mod 11

Or, (28 mod 11) ⋅ (35 mod 11) mod 11

Or, 6 ⋅ 2 mod 11

Or, 12 mod 11 = 1

**Answer**: 1

1. 90 ⋅ 102 = 6 (mod 11)

Or, 90 ⋅ 102 mod 11

Or, (90 mod 11) ⋅ (102 mod 11) mod 11

Or, 2 ⋅ 3 mod 11

Or, 6 mod 11 = 6

**Answer:** 6

1. [**Random variables**] Suppose that a 4-bit string is chosen uniformly at random.   
   What is the probability that the sum of its bits is exactly 3?

Explain your answer.

Tips: A uniform distribution implies that the probability for each 4-bit string to appear is 1/16.   
As a few examples, we have:  
The string (0000): the sum of its bits is 0. The string (0001): the sum of its bits is 1.  
The string (0010): the sum of its bits is 1. The string (1010): the sum of its bits is 2.  
The string (0111): the sum of its bits is 3. The string (1111): the sum of its bits is 4.

Here,

The total number of 4-bit strings is 2^4 = 16

For the number of 4-bit string that has sum of exactly 3, the string must have three 1s and one 0.

Which of them are (0111), (1011), (1101) and (1110) = 4

We can also find them by choosing 3 positions of 1s from 4 in 4-bit string.

i.e. Number of 4-bit strings that has sum of 3 is 4C3 = 4! / (3! \* (4-3)!) = 4

Thus, the probability that the sum of its bits is exactly 3 = 4/16 = 1/4