

PART - A

Implement the following in Modular Arithmetic: $\mathbb{Z}_n = \{0, 1 \dots n-1\}$

1. Additive inverse of a number
2. Multiplicative inverse of a number
3. Inverse of an $m \times m$ matrix with $m \leq 3$
4. Implement Diffie Hellman Key Exchange (DHKE) Protocol, with small integer values for testing. Hence deliver a man-in-the-middle attack on the DHKE Protocol.

PART - B

1. Implement the Euclidean Algorithm below, to find GCD of two numbers:

EUCLIDEAN ALGORITHM(a, b)

```

 $r_0 \leftarrow a$ 
 $r_1 \leftarrow b$ 
 $m \leftarrow 1$ 
while  $r_m \neq 0$ 
  do  $\begin{cases} q_m \leftarrow \lfloor \frac{r_{m-1}}{r_m} \rfloor \\ r_{m+1} \leftarrow r_{m-1} - q_m r_m \\ m \leftarrow m + 1 \end{cases}$ 
 $m \leftarrow m - 1$ 
return ( $q_1, \dots, q_m; r_m$ )
comment:  $r_m = \gcd(a, b)$ 

```

2. Given two integers a and b , the following algorithm computes GCD (a, b) as well as $b^{-1} \bmod a$, when a and b are co-prime to each other. This is called the Extended Euclidean Algorithm.

EXTENDED EUCLIDEAN ALGORITHM(a, b)

```

 $a_0 \leftarrow a$ 
 $b_0 \leftarrow b$ 
 $t_0 \leftarrow 0$ 
 $t \leftarrow 1$ 
 $s_0 \leftarrow 1$ 
 $s \leftarrow 0$ 
 $q \leftarrow \lfloor \frac{a_0}{b_0} \rfloor$ 
 $r \leftarrow a_0 - qb_0$ 
while  $r > 0$ 
   $\begin{cases} temp \leftarrow t_0 - qt \\ t_0 \leftarrow t \\ t \leftarrow temp \\ temp \leftarrow s_0 - qs \end{cases}$ 
  do  $\begin{cases} s_0 \leftarrow s \\ s \leftarrow temp \\ a_0 \leftarrow b_0 \\ b_0 \leftarrow r \\ q \leftarrow \lfloor \frac{a_0}{b_0} \rfloor \\ r \leftarrow a_0 - qb_0 \end{cases}$ 
 $r \leftarrow b_0$ 
return ( $r, s, t$ )
comment:  $r = \gcd(a, b)$  and  $sa + tb = r$ 

```

This is how the above algorithm works:

Given a and b it computes another two number s and t such that $s \times a + t \times b = r = \text{GCD}(a, b)$.

Now, we aim to find $b^{-1} \bmod a$, which exists iff a and b are co-prime.

Since a and b are co-prime $r = 1$.

Therefore, $s \times a + t \times b = 1$.

Applying $\bmod a$ to both sides, $(s \times a + t \times b) \bmod a = 1 \bmod a$.

Or, $t \times b \bmod a = 1$ [Since $s \times a \bmod a = 0$.]

Or, $b^{-1} \bmod a = t$. (Think why we also output s .)

Implement the Extended Euclidean Algorithm. Hence prove $28^{-1} \bmod 75 = 67$
