A3. [**40**] Hyperplane classification is an important classification task in data analysis. Given a hyperplane classifier model $W = \{W_i\}_{i=1}^{\ell}$ with ℓ vectors in $\mathbb{R}^d$, an input $x = (x_1, \cdots, x_d)$ is classified as

$$t = \text{argmax}_{i\in[\ell]}\Big\{W_i \cdot x\Big\} = \text{argmax}_{i\in[\ell]}\Big\{\sum_{j=1}^{d} w_{ij}x_j\Big\}$$

where $W_i = (w_{i1}, w_{i2}, \cdots, w_{id})$, $W_i \cdot x = \sum_{j=1}^{d} w_{ij}x_j$ is the dot-product between W_i and x, and *argmax* is the function that returns the argument/index that gives the maximum value.

In this question, we consider implementing a private hyperplane decision classifier computation protocol given in Figure 1 where a cloud holds a private hyperplane classifier model W and a user holds a private input x. The protocol is based on the Paillier homomorphic encryption scheme. Let E and D denote the encryption and decryption algorithms for Paillier. You can use the encryption and decryption functions implemented in task **A2**. To perform the protocol, the cloud and the user perform the following steps:

- **User:** The user first generates the private and public key pair of the Paillier encryption scheme, denoted by (sk, pk). It encrypts its private input vector x as $E(x) = (E(x_1), E(x_2), \cdots, E(x_d))$ using the public key pk, and sends $E(x)$ to the cloud.
- **Cloud:** As the cloud holds W, for each W_i, the cloud computes the encrypted inner product $E(W_i \cdot x) = \prod_{j=1}^{d}\Big(E(x_j)\Big)^{w_{ij}} = E\Big(\sum_{j=1}^{d} x_j w_{ij}\Big)$ from $E(x)$ and W_i for $i = 1, 2, \cdots, \ell$. Then, it randomly generates a non-zero number r from $\mathbb{Z}_n^*$ and computes $C_i \leftarrow E(W_i \cdot x) \cdot E(r) = E(W_i \cdot x + r)$ for $i = 1, 2, \cdots, \ell$. The ciphertexts $(C_1, C_2, \cdots, C_\ell)$ are sent to the user.
- **User:** After receiving $(C_1, C_2, \cdots, C_\ell)$, it decrypts each C_i using the private key sk as $v_i \leftarrow D(sk, C_i)$ and then computes the argmax function to get the index t as $t = \text{argmax}_{i\in[\ell]}\{v_1, v_2, \cdots v_\ell\}$.

Please implement the protocol (Figure 1) in the same code where you don't need to implement the message exchange protocol. For simplicity, consider only the (positive) integer values of W and x, and the value of the random number $r < 2^{256}$. Randomly generate the values of W and x where $0 \le w_{ij} \le 100$ and $0 \le x_i \le 100$. Example parameters $\ell = 30$ and $d = 10$.

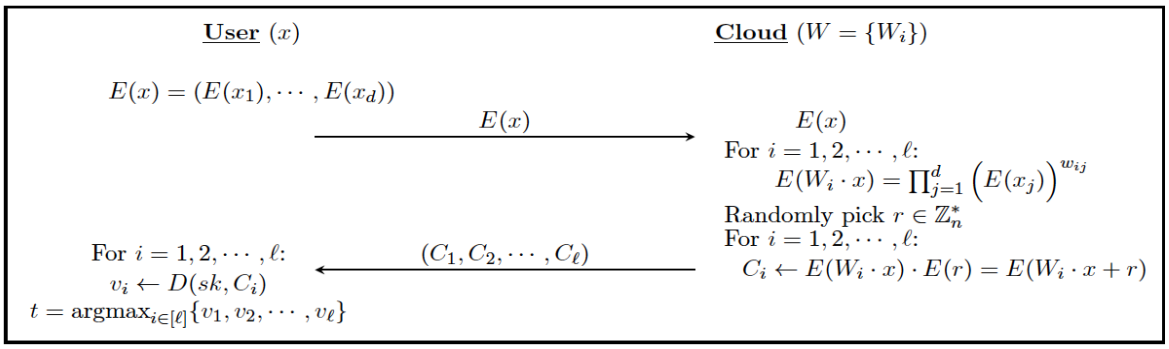

Figure 1: Private hyperplane classification protocol

Sample I/O:

```
----------------------------
Please enter ℓ (≥ 2): _____
Please enter d (≥ 2): _____
----------------------------
Encrypted vector E(x): _____
Encrypted result (C_1, ··· , C_ℓ) : _____
----------------------------
The input x belongs to the class: _____
----------------------------
```