UNIVERSIDAD EL BOSQUE

MATEMÁTICAS APLICADAS

Taller 1c

Desarrollar los siguientes puntos bajo el esquema cliente-servidor mediante un menú creado en Python. Se indica lo que debe hacer el cliente y lo que debe hacer el servidor.

1. ***Cliente***: Lee un número entero positivo que indica una posición y se lo envía al servidor.
   Servidor: Imprime el número primo de la posición que recibió.
   Tamaño de envío: Sin restricción

2. ***Cliente***: Lee una palabra y un número entero positivo que representa un desplazamiento.
   Servidor: Imprime la palabra cifrada con el algoritmo Cesar y con el desplazamiento que se le indicó.
   Tamaño de envío: Sin restricción

3. ***Cliente***: Pide el nombre del archivo (jpg, pdf, xlsx, etc.), lo codifica en base64 y se lo envía al servidor.
   Servidor: Restaura el archivo correspondiente con el formato original (jpg, pdf, etc.)
   Tamaño de envío: 1 byte

4. ***Cliente***: Lee el nombre de un archivo donde está la llave que se usará para encriptar un mensaje. Encripta un mensaje usando el algoritmo de Hill y lo envía al servidor.
   Servidor: El servidor también tiene la llave guardada en un archivo de texto, por lo que debe leer el archivo para poder desencriptar el mensaje.
   Tamaño de envío: 1 byte

5. ***Cliente***: Lee un mensaje para luego encriptarlo usando el algoritmo XOR y luego enviarlo al servidor.
   Servidor: Desencripta lo que recibe y luego lo Imprime.
   Tamaño de envío: 1 byte

6. ***Cliente***: Lee el nombre de un archivo el cual debe convertir a texto con formato binario y enviarlo al servidor. También debe enviar el valor hash del archivo.
   Servidor: Restaura el archivo original y verifica si el valor hash sí corresponde al archivo restaurado.
   Tamaño de envío: 1 byte

7. ***Cliente***: Genera una llave pública y una llave privada para el cliente y para el servidor las cuales se usarán en el siguiente punto.

8. ***Cliente:*** Lee un mensaje y lo encripta con la llave pública del servidor. Luego lo envía al servidor.
   Servidor: Descifra el mensaje y lo imprime. Se supone que el servidor conoce su llave pública generada en el punto anterior.
   Tamaño de envío: Sin restricción

9. ***Cliente***: Lee un número de cédula y su dígito de verificación de acuerdo con el algoritmo 1 para luego enviarlo al servidor.
   Servidor: Verifica si el número de celular concuerda con el dígito de verificación.
   Tamaño de envío: Sin restricción

10. ***Cliente***: Lee una palabra, y la transforma en una secuencia binaria para transmitir. Aleatoriamente altere uno de los bits que se van a transmitir.
    Servidor: Mediante el método de Hamming, el programa detecta el bit alterado y lo corrige. Luego de recibir toda la información debe imprimir la palabra original.
    Tamaño de envío: 1 byte.

Nota: Si tiene alguna duda de si puede o no usar una función o librería por favor hágamela saber. En principio pueden usar solamente lo que hemos visto. Cualquier similitud con otro trabajo tendrá que explicarlo o sustentarlo oralmente. Por favor enviar solamente un único archivo de Python.