PRUEBA TÉCNICA
DESARROLLADOR C++

Introducción

El presente documento reúne la información de la prueba técnica que deberá realizar el candidato que opte a Desarrollador C++ del equipo T2MC como parte de la primera fase del proceso de selección. El resultado final de la prueba realizada deberá entregarse de acuerdo a las directrices indicadas más abajo y en los plazos convenidos entre ambas partes.

T2MC se comunicará con el candidato (una vez valorado el trabajo presentado), para indicar los siguientes pasos del proceso de selección en el supuesto que haya aprobado satisfactoriamente la prueba.

El propósito de esta prueba es evaluar las habilidades, conocimientos del lenguaje, conocimientos técnicos del sector, capacidad de resolución de problemas y la correcta utilización de herramientas complementarias de acuerdo a las necesidades.

Enunciado del Problema

Las claves privadas de los wallets de criptomonedas son datos alfanuméricos de más de 100 caracteres. Si un usuario tiene que teclear esta clave, o peor aún, si pretende memorizar o almacenar en algún medio físico, el trabajo será complejo y con un alto margen de error humano. Conscientes de ello, dentro del estándar existe una propuesta de mejora denominada BIP39 que define el uso de “frases semilla”.

Para ello BIP39 establece un sistema mnemónico, con 12 a 24 palabras, que posibilita restaurar billeteras a partir de esa “frase” (con palabras fáciles de recordar y manejar). En síntesis, a partir de un diccionario, el algoritmo encargado de formar la frase, irá eligiendo palabras de forma aleatoria de acuerdo al grado de entropía definido en el propio algoritmo. En el siguiente enlace se encuentra la espeicificación oficial del BIP39: https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki

El problema de esto es que la seguridad recae en el propio algoritmo que establece la entropía. Un algoritmo basado en software puede generar números aleatorios, pero lo hace a través de mecanismos que podrían ser replicables o tener un menor grado de entropía real. La solución a esto es delegar la entropía a la propia naturaleza (dados y monedas).

En el siguiente enlace se encuentra una implementación en JavaScript que establece como obtener una frase de 24 palabras utilizando únicamente cuatro dados y una moneda: https://blockmit.com/guias/variadas/generar-semillas-bip39-con-dados/

La prueba técnica consiste en replicar la solución descrita en el enlace anterior (solución taelfrinn), utilizando lenguaje C++ y de acuerdo a las siguientes especificaciones:

- Como input, desde consola se solicitará al usuario introducir cada combinación de dados y moneda siguiendo el formato propuesto por taelfrinn. Debe pedir en bucle una combinación por iteración, hasta tener las 23 necesarias. Ejemplos de entradas válidas: primera iteración => T1342, segunda iteración => H3211. tercera iteración => H1111, cuarta iteración T1121, etc.
- El algoritmo deberá calcular la palabra 24, haciendo el checksum y utilizando como entropía el input (ver especificación del BIP39).
- Teniendo las 24 claves, el algoritmo deberá ser capaz de ir a buscar la palabra correspondiente a cada una de ellas, dentro del diccionario de palabras EN de la especificación BIP39.
- Como output, el algoritmo deberá imprimir en consola la frase resultante, incluida la palabra 24, cada palabra deberá ocupar una fila de la pantalla.

Requisitos

Obligatorio

1. Código fuente (archivos .cpp y .h)
2. Modularización del software y aplicación de la POO
3. Incluir dentro del entregable, todas las dependencias necesarias.
4. Especificar la configuración de entorno requerida.
5. Comentarios en código y archivo txt “Readme” con indicaciones de arranque
6. Ejemplos de inputs y outputs válidos.

Deseable

1. Implementación propia de funciones para operaciones cheksum y hashing.
2. Separación de código por librerías
3. Manejo de archivos: Diccionario de Mnemonics fuera del fuente

Entrega

Para realizar la entrega de la solución, deberá responder al correo desde el que se le envió este documento, adjuntando en una única carpeta comprimida, todos los documentos de la solución (Muchos clientes de correo no permiten el envío de archivos .exe o código fuente).