NOTAS

========

Estas son las tareas que creo que hay que realizar junto con las dudas que me han surgido. Para que podáis poner y completar todo esto vosotros se me ha ocurrido usar un código de colores. Cada uno escribe en un color diferente lo que ha entendido como sus dudas para así tener algo consistente y sin ellas al final y poder ir a objetivo claro y fijo tanto en el reparto como en la realización.

*Código de colores*

Luis-> rojo

Miguel-> verde

Samuel->

Alberto

Ítems a realizar

=================

**.)Proceso encargado de la E/S**

(¿proceso con id 0?) que puede calcular claves o no (creo que se puede mantener independiente que solo haga eso porque sino habría que interrumpirlo muchas veces en su labor de calcular por lo que lo dejaría solo como “coordinador”) . No sé si recoge las claves metidas por el usuario o las lee de una lista de claves a desencriptar. Creo que este proceso tiene una lista de claves encriptadas y su correspondiente original y le manda las encriptadas a los calculadores.

***Labores de este proceso:***

->Generar lista de claves a repartir y repartirlas al resto de procesos ( aquí no sé si a cada proceso le da un rango en el que probar o cómo. No lo llego a entender muy bien) O si a cada proceso le da una clave diferente a desencriptar(creo que es esto).(Creo que a cada proceso le manda la clave encriptada y la original para que pueda comparar y saber cuándo la ha desencriptado)

->Si hay menos claves que procesos asignar la misma clave a desencriptar a diferentes procesos.

->De alguna manera que cuando un proceso desencripte la clave, anotarla en una lista y detener todos aquellos procesos que también estuviesen intentando desencriptar esa clave. Asignar a todos ellos una nueva clave a desencriptar (que ayuden a los demás)

->Asignar a los procesos calculadores que hayan desencriptado su clave una nueva clave a desencriptar (que ayuden a los demás)

->Recolectar datos (clave encontrada, quién la ha encontrado, número de intentos, tiempo)

->A partir de ahí hacer un cálculo del número de claves comprobadas por segundo.

->Mostrar estadísticas finales (¿Una vez desencriptadas todas las claves? Imagino que sí)

->Calcular el rango válido para la generación de números aleatorios por parte de los procesos calculadores en función de la clave que les mande desencriptar

->Mantener un contador de cuántas claves se han desencriptado para cuando llegue al número de claves que tenía para desencriptar acabar el programa generando previamente las estadísticas.

**.)Procesos calculadores.**

***Labores:***

->Esperar a que el proceso de E/S (“coordinador”) les mande una clave a desencriptar.

->Desencriptar la clave generando números aleatorios en un rango válido que les pase el proceso coordinador. Cada proceso tiene que tener una semilla de generación diferente. (El código para ver si han conseguido desencriptar creo que es el que viene en el enunciado)

->Mandar cuando desencripten una clave el resultado al proceso coordinador con el número de veces que lo intentaron, el tiempo, su id, la clave desencriptada con el resultado.

**,) Objetivos encontrados**

* El **proceso con id 0** (sacado a través de la función MPI\_Comm\_rank) es el único proceso que conoce todas las claves encriptadas.
* **Proceso E/S** (puede ser calculador o no) se encarga de generar la lista de claves encriptadas (especifica más adelante que será random ,es decir, cifras de 7 digitos aleatorias) y de comunicarlas de una en una al resto de procesos para que las vayan desencriptando. Si algún proceso acaba antes que el resto de desencriptar su clave, ayuda a otro proceso que aún este buscando la suya (se le asigna la clave de un proceso que no haya acabado 🡪 asignar alguna especie de flag para los procesos que hayan acabado su tarea para saber que claves descartar y avisar a los procesos con algún tipo de interrupción o mensaje). **IMPORTANTE:** los procesos que buscan la misma clave tienen que conocerse para poder mandarse mensajes

Algo asi como que todos los procesos tengan un MPI\_Recv y mientras no les llegue ningún mensaje no se bloquean y revisan esa función en ¿cada iteración? Y cuando les llegue se quedan bloqueados y terminan indicando que esa clave que estaban buscando ya ha sido encontrada por otro proceso (y este proceso que la encuentra envia un MPI\_Send)

* Cada proceso muestra quien es y la clave que ha encontrado así como otros parámetros meramente informativos
* Estadísticas finales de cual ha sido el proceso mas rápido y cuantas claves ha desencriptado cada uno y esas mierdas
* **Proceso Calculador** esperan a que el E/S les de la clave (¿a través de un MPI\_Recv o algo así y así nos aseguramos de que se quedan esperando a que les llegue la clave a cada uno? y ¿luego con un MPI\_Send indicamos al proceso 0 que hemos acabado y este al resto para que el resto de procesos comprueben si el proceso que ha acabado va a buscar la misma clave que están buscando ellos?.). La desencriptan con números aleatorios (OJOOOOOOOOOOOO -> cada proceso utiliza una semilla diferente para generar los números aleatorios)
* **Todo pasa por el proceso 0.**
* Al ppio el contador de todas las claves está inicializado a -1 como se ve en la prueba que han hecho ellos, y según la encuentren los procesos se pone a la derecha el número del proceso que la ha encontrado
* Debajo del todo de cada ejecución poner los procesos que están calculando – la clave que están buscando – y el numero identificativo de esa clave (del 00 al 19)