



Tarea 2

Esta tarea estará orientada al caso donde se heredan proyectos *a medias* y hay que trabajar sobre ellos tal y como se toman. Fecha de entrega: Sábado 06 de Mayo del 2023, 23:55.

- Repositorio base de la tarea 2: https://github.com/ebadilla10/tarea2_ie0217_i2023
- Realice un *fork* para que tenga una copia exacta de esta versión del repositorio base, pero ahora bajo su nombre. <https://docs.github.com/en/get-started/quickstart/fork-a-repo>

Algoritmos de ordenamiento:

Sobre los algoritmos de ordenamiento, únicamente nos entregaron en el repositorio las definiciones de las funciones, las cuales se encuentran en el archivo headerSort.hpp. Sus tareas consisten en lo siguiente:

1. Documentar, siguiendo las pautas Doxygen, cada una de las funciones declaradas en el header. Debe guiarse en los nombres de las funciones, así como los argumentos que estas reciben.
2. Implementar cada algoritmo de ordenamiento en un único archivo fuente.
3. Analice cada algoritmo y determine la complejidad que estos tienen utilizando la notación Big O (O grande). Su análisis debe documentarlo en el archivo README.md para que sea desplegado en la página web de github al entrar en dicha carpeta.

Templates

Sobre templates, nos dejaron lo que parece ser una implementación completa de una clase pila mediante templates. Así como una pequeña función main donde es utilizada esta plantilla para el caso del tipo entero. Sus tareas consisten en lo siguiente:

1. Verificar si todos los *includes* agregados son necesarios para la compilación del código dado. Esto es necesario realizarlo porque en ocasiones se incluyen headers totalmente innecesarios, y si el compilador no posee instrucciones de ignorar los archivos no utilizados, se agregará código extra innecesario.
2. Comentar el código del template, ya que quien nos heredó el proyecto no colocó ni un sólo comentario. Si nota algún método que no conoce, proceda a investigarlo y comprender su uso.
3. Comentar el código de la función main.
4. Agregar una nueva instancia donde utilice el template dado para otro tipo de dato. Agregue algunas líneas utilizando esta nueva instancia.
5. Agregar en las líneas del paso anterior, pero dejándolo comentado, alguna sentencia, declaración o instrucción que *dispararía* la excepción *out_of_range* presente en el template.



Excepciones

Sobre excepciones, únicamente nos dejaron un archivo donde se utiliza la clase *exception* de la librería estándar para crear una excepción personalizada mediante herencia. Sus tareas consisten en lo siguiente:

1. Documentar el código, el cual es corto, sin embargo, necesita una explicación.
2. Realizar la división en archivos de encabezado y fuente para una correcta estructura de archivos.
3. Investigue sobre tres distintos tipos de excepciones incluidas en la librería estándar, y utilice estas mediante un pequeño código que *dispare* dichas excepciones.

Entrega

- Posterior a realizar el *fork* del repositorio base, se tendrán tres carpetas con los archivos base a trabajar. Se debe crear un branch llamado *tarea2* (así en minúscula) y trabajar todo sobre este branch.
- Debe agregar un README.md dentro de cada carpeta, acá se debe agregar los incisos de investigación, como por ejemplo los análisis de complejidad en la parte de algoritmos de ordenamiento. Así también las instrucciones de ejecución de cada programa.
- Debe crear un makefile para cada parte, tal que, al estar ubicado en dicha carpeta, el correr el comando *make* sea suficiente para la ejecución del programa.
- La carga de los archivos debe ser a través de la línea de comandos, no utilizando la interfaz web de github.
- Al terminar su tarea, investigue qué es un *pull request*. Debe realizar un pull request desde su branch hacia el branch principal (master en este caso).