תיעוד מטלה 3 C++:

בניית מחלקת Member.
מבנה הנתונים שבחרנו הינו Map.
Maps מאפשרות איחסון נתונים באינדקס על ידי מפתחות. הראשון הוא משתנה קבוע ואינו ניתן לשינוי (Key) והערך השני הוא בר שינוי (Value). בנוסף, לכל member חדש שנבנה קיים מספר סידורי (id) שניתן על ידי המערכת והוא יהיה ה-key בערך הראשון והאובייקט עצמו יהיה ה-value, בערך השני של ה- Map.
בעזרת זה שיש id שלא משתנה לא ניתנת האפשרות לקיום של כפילויות בין משתמשים - קיים משתמש אחד בעל מספר סידורי שיחודי לו ולא ניתן לשנותו.
ל- Maps ניתן לגשת באמצעות איטרטורים שהם למעשה מצביעים על אובייקטים בתבנית של זוג של סוג הבסיס, הראשון (id) והשני (אובייקט מסוג Member). הראשון מתאים למפתח, השני לערך המשויך למפתח.
קיימות ל- Map פונקציות שמוכנות לשימוש כגון הכנסה/מחיקת איבר, הצגת גודל ה-Map, הצגת הערך הראשון/ האחרון וכאשר גודל ה- Map הוא n, הכנסה/הוצאה/מחיקה הן פעולות בזמן של O(log(n)).

האפשרויות שניתנות לכל Member:

- Follow: ה-Member יכול לעקוב אחרי Member אחר אך לא אחרי עצמו.
- Unfollow: ביטול מעקב אחר Member מסויים.
- get_ID: מחזיר את המספר הסידורי של אותו Member.
- numFollowers: מחזיר את מספר העוקבים אחר אותו Member.
- numFollowing: מחזיר את מספר האנשים שאותו Member עוקב אחריהם.
- count: מחזיר את מספר המשתמשים הפעילים.

בנוסף קיימות פונקציות להקלת הכתיבה לשימוש ב-Map:

- add_followed: כאשר a.follow(b) , נכנס ל Map של העוקבים אחריי של אובייקט b ומכניסים את האובייקט a כעוקב אחרי אובייקט b.
- erase_followed: כמו בפונקציית add_followed רק בשביל מחיקה. שימושי בזמן הפעלת unfollow.
- erase_following: שימושי בזמן Destruct, מוחק ב- Map של מי אני עוקב אחרי של אובייקט a את אובייקט b.

- is private.
+ is public.

Member:
- n_ID : int - followed_by : Map - following : Map - add_followed(Member) : void - erase_followed(int) : void - erase_following(int) : void
+ follow(Member) : void + unfollow(Member) : void + get_ID(void) : int + numFollowers(void) : int + numFollowing(void) : int + count(void) : int