תרגיל בית 2 – חלק יבש:

1. **השאלה:** במבנה ה$sorted\ list$ הגנרי, מה הדרישות **ההכרחיות** שעל הטיפוס T לקיים? הסבירו
תשובה: נתון שלT אין בנאי חסר פרמטרים, וכתוב שעל המימוש לא להיות תלוי במשתמש מבחינת זיכרון, כלומר עליו ליצור עותקים משל עצמו לכל האובייקטים שהוא מקבל. לפיכך יש צורך בבנאי העתקה כדי שנוכל להעתיק את העצם לשם יצירת עותק משלנו. בנוסף צריך הורס כי כשעושים $remove$ מסירים את העצם, ובגלל שהוא גנרי יכול להיות שהוא טיפוס מורכב והוא צריך הורס כדי לא לפספס שחרור של זיכרון. יתר על כן צריך את אופרטור $<$ כי נתון שאיברי הרשימה ממוינים בכל רגע נתון ופעולות ה$insert$ וה$remove$ שומרות על הסדר בין איברי הרשימה בעזרת האופרטור $<$ של הטיפוס T.

2. **השאלה:** נניח כי היינו רוצים לממש איטרטור $non-const$ עבור $SortedList$. כלומר, עבור איטרטור זה האופרטור $*$ היה מחזיר $T\&$. איזו בעיה עלולה להיווצר במימוש?
תשובה: הייתה נוצרת בעיה שהיינו מאפשרים למשתמש לשנות את הערכים בתוך הרשימה (את העצם עצמו) וזה היה פוגע במיון של הרשימה. כשרצינו להכניס או להסיר איבר השתמשנו בפונקציות שמיועדות לכך, וכשהן עשו את זה הן דאגו לא לפגוע בסדר הפנימי של האיברים ברשימה, אבל אם האיטרטור לא יהיה קבוע המשתמש יוכל לשנות ערך של איבר בעצמו ולבטל את הסדר שנעשה בעזרת הפונקציות. כלומר ננסה לנהל מיון בזמן שהמשתמש יוכל לפגוע בזה ובעצם נוכל להיכשל במיון בגלל זה.

*שאלה 3 בעמוד הבא

3. **השאלה:** במתודה $filter$, המימוש כולל שימוש ב$template$ נוסף כדי להעביר את הפרדיקט. אילו שתי דרכים שונות קיימות בשפת $C++$ למימוש והעברת הפרדיקט? מה ההבדל ביניהן? מדוע אנחנו לא צריכים לספק שני מימושים שונים כדי לתמוך בשתי הדרכים הללו?
תשובה: הדרך הראשונה היא להעמיס את $operator()$ עבור הטיפוס T (כלומר במבנה שלו) ככה שהוא יבדוק אם התנאי על איברי הרשימה מתקיים ויחזיר ערך בוליאני בהתאם. הדרך השנייה היא, בדומה למה שראינו בc, בעזרת מצביע לפונקציה בוליאנית, דרך זו קיימת גם ב$c++$.
ההבדל בין הדרכים הוא שבמצביע לפונקציה לא צריך עצם. בשפה $c++$ מעדיפים לעבוד עם עצמים ולא עם מצביעים ולכן זו אופציה קיימת אך פחות עדיפה. בדרך הראשונה האופרטור הוא רק חלק ממבנה (חלק ממה שאפשר לעשות על העצם) ולכן אנחנו יכולים לנהל עוד דברים בנוסף לאופרטור זה וליצור עוד מתודות למשל, ובעצם לנהל את הטיפוס באופן שלם.
לא צריך ליצור שני מימושים שונים כדי לתמוך בדרכים אלו כי אנחנו עשינו את המימוש גנרי בעזרת $template$ שיכול לקבל כל טיפוס שהוא או מצביע וכדומה, בתנאי שמה שהוא מקבל יכול לעבוד עם האופרטור $()$ בו אנחנו נשתמש לצורך הבדיקה ובשתי הדרכים זה מתאפשר.