

## תרגיל בית 3

### סעיף א'

לא ניתן להסתפק בלהגדיר את פעולות האיטרטור הרגיל כ-const כיוון שהדבר יגביל את האיטרטור. לדוגמה הגדרת האיטרטור כ-const מונעת ממנו להצביע על ערכים שונים – כלומר האיטרטור נשאר קבוע ולא יכול להתקדם בלולאה (והרי שזהו תפקידו המקורי). לכן על מנת לאפשר לרוץ על כל איברי התור ועדיין לוודא שלא מתבצע שינוי ערכיהם עלינו להגדיר שימוש ב-ConstIterator בנוסף לאיטרטור הרגיל של התור.

### סעיף ב'

הנחת המוצא שלנו היא ש-T היא מחלקה ולכן קיימים לה:

- Constructor
- Destructor
- Copy constructor
- Assigning operator

בפונקציות הבאות הנחנו כי ההנחות הנ"ל מתקיימות במחלקה T:

- Queue copy constructor
- Queue assigning operator
- Queue push back
- Queue popFront
- Transform
- Filter

בנוסף בכל הפונקציות של Iterator & ConstIterator השתמשנו גם בהנחות אלה.

### סעיף ג'

מימוש המחלקה Queue בקובץ cpp במקום בקובץ h. יגרום לשגיאה בשלב linker. השגיאה שהסטודנט יקבל היא undefine refernce מכיוון שבשלב הקומפילציה הקומפיילר מייצר אובייקטים בהתאם למה שכתוב בקובץ ה-h אך אם המימוש לפונקציות יהיה כתוב בקובץ cpp אזי שהקומפיילר לא חשוף אליו ולכן הם עלולים להיות מאותחלים לא כנדרש. לכן בשלב linker הוא יחפש את המימוש של המחלקות הנ"ל ולא ימצא אותם ולכן תתקבל השגיאה הנ"ל.

### סעיף ד'

על מנת לאפשר סינון של התור עם ארגומנט שמתקבל בזמן ריצת התכנית על הסטודנטית לעשות:  
לקלוט מספר מהמשתמש בזמן ריצה ולוודא תקינות (שונה מ-0).  
ליצור פונקציה בוליאנית אשר שומרת את המספר הנ"ל (אופציה לדוגמה היא משתנה סטטי) ולכל מספר שלם שהיא מקבלת היא מחזירה true אם מתחלק במספר השמור ללא שארית false אחרת.  
לקרוא לפונקציה filter עם הפונקציה משלב 2.  
כך הסטודנטית יכולה לשתמש בפונקציה filter לטובת פרוייקט הצד שלה.