

HW3 DRY

סעיף (א):

נרצה במקרים מסוימים לאפשר קריאה של הערכים ולא שינוי שלהם, לכן איטרטור רגיל לא מספיק, מאחר ויש לו יכולת לשנות את ערכי התור עליו הוא נע. לשם כך נגדיר `ConstIterator`, שמחזיר `const reference` במקום `reference` רגיל לתור הנתון.

סעיף (ב):

| הנחות על הטיפוס הטמפלייטי | פונקציה בממשק של התור |
|------------------------------------|---|
| קיום אופרטור השמה ל <code>T</code> | <code>Queue(const Queue& other)</code> |
| קיום אופרטור השמה ל <code>T</code> | <code>Queue& operator=(const Queue& other)</code> |
| קיום אופרטור השמה ל <code>T</code> | <code>void pushback(const T& object)</code> |
| קיום אופרטור השמה ל <code>T</code> | <code>void popFront()</code> |

סעיף (ג):

השגיאה תתרחש בשלב הקומפילציה, הקומפיילר לא יזהה את הקריאות לפונקציות שמוגדרות על ידי תבנית, בפרט לא יזהו המשתנים של התבניות כמו `T`.

סעיף (ד):

ראשית, הפונקציה תמומש כחיצונית למחלקה `Queue` ולא כמתודה של המחלקה. מאחר והפונקציה לא מתודה, נדרש איטרטור כדי לעבור על אברי התור הנתון. הפונקציה `filter` ממומשת על ידי תבנית, בעזרת שני משתני תבנית `S, T` כך ש:-

(1) `T` הוא הטיפוס שמשמש לזיהוי הקלט `Queue<T>`

(2) `S` הוא טיפוס הפרדיקט (התנאי) (`function object/ function pointer`) הבוליאני שמניחים שיש לו אופרטור `()` כדי לשלוח אליו את אברי התור הנתון.

(3) נקצה תור חדש (שנחזיר אותו מהפונקציה) ונעבור בעזרת האיטרטור על אברי התור הנתון ונשלח אותם לפרדיקט הנתון, אם יוחזר `true` בעזרת המתודה `pushback` שמימשנו למחלקה, נכניס את האיבר הנוכחי לתור שנחזיר. אחרת נדלג לאבר הבא.

מאחר ואנחנו משתמשים בפרדיקט, לא נדרש מאיתנו לדעת את המספר שנרצה לבדוק חלוקה בו.