

מת"מ תרגיל 3 – חלק יבש

מגישים: אלכס פרומן – 941180150,
עומר מאושר – 208953463.

תשובות:

- א. לא ניתן כי `operator*` של `ConstIterator` מחזיר ערך גנרי `T const`. כלומר כאשר נבצע * דרך איטרטור קבוע בתור משתמש, לא נוכל לערוך את הערך שקיבלנו.
- ב. עבור הטיפוס הגנרי `T` קיימות ההנחות הבאות בממשק התור:
- a. בפונקציה `pushback` – הנחה על `operator=()` של `T`.
 - b. בבנאי ההעתקה של התור – הנחה על `operator=()` של `T`.
 - c. ב `operator=()` של התור – הנחה על `operator=()` של `T`.
- הנחות נוספות:
- d. בפונקציה `filter` – הנחה על `operator=()` של התנאי המסופק כפרמטר, ובנוסף הנחות בסיסיות על איטרטור על `data` גנרי שיאפשר ריצה על מערך (אצלנו מומש כמעטפת - `Element` המכיל את `T`). הדרישות המינימליות לאיטרטור: `++()`, `!=`, `<` וכמו כן פונקציות `begin()`, `end()` של התור הגנרי.
 - e. בפונקציה `transform` – הנחות בסיסיות על איטרטור על `data` גנרי שיאפשר ריצה על מערך (אצלנו מומש כמעטפת - `Element` המכיל את `T`). הדרישות המינימליות לאיטרטור: `++()`, `!=`, `<` וכמו כן פונקציות `begin()`, `end()` של התור הגנרי.
- ג. הסטודנט יקבל שגיאת קומפילציה שתתקבל בזמן קומפילציה (תהליך `instatiation` של הקומפיילר שיכשל). במימוש מחלקה/פונקציה גנרית הקומפיילר יחפש באותו קובץ (h) את המימוש עצמו כדי שיוכל לבצע בעצמו יצירה של המחלקה/פונקציה הגנרית עם ה `type` הספציפי הנדרש. היות והמימוש של הסטודנט נמצא ב `cpp`, הקומפיילר לא יצליח לבצע את תהליך `instatiation` ויחזיר שגיאת קומפילציה.
- ד. ניתן ליצור `function object` שפועל כמו `devidedBy` כפי שראינו בהרצאות ובתרגולים. בזמן ריצה ניתן ליצור את האובייקט `devidedByN` לפי המספר `N` הנדרש (באמצעות הבנאי של `devidedBy`). כך נוכל בהעברת הארגומנט `devidedByN` כתנאי לפונקציה `filter` לסנן את האובייקטים המתחלקים במספר הנדרש.