

## חלק ב' – שאלות יבשות

1. על פרמטרי התבנית קיימים הדרישות הבאות:

- a. עבור Element צריך להיות קיים בנאי העתקה. הסיבה לכך היא שבעת הכנסת אבוייקט Element למערך הייחודי אנו מעתיקים את האבוייקט שניתן כארגומנט בעזרת בנאי העתקה (copy c'tr) ובהנחה שכזה אכן קיים. לולא קיום כזה – הקומפיילר היה יוצר אחד דיפולטיבי, ובשל כך, היינו עלולים לחוות התנהגות לא מוגדרת.
- b. עבור Compare נדרש מימוש (העמסה) של אופרטור ( ) אשר מקבל שני אבוייקטים מאותו הסוג (Element) ומחזיר ערך בוליאני. לולא מימוש כזה נזכה או לשגיאת קומפילציה או להתנהגות לא מוגדרת (תלוי באופן המימוש).

2. לו היינו מממשים העמסה של אופרטור [ ], אשר מקבל מספר סידורי של איבר ב- UniqueArray ומחזיר את המצביע לאיבר היינו עלולים לחוות שגיאת קומפיילר: בעוד שעבור Element מסוימים (Class מסוימים) הקומפיילר היה מבדיל בין שימוש ב- [ ] כאשר הארגומנט שניתן בתוך הסוגריים הוא Element ובין שהארגומנט שניתן הוא unsigned int, לו היינו בוחרים ש Element יהיה למעשה unsigned int הקומפיילר היה מתריעה על שגיאה שבה כתובות לו שתי הצהרות זהות לפונקציות, וזאת מכיוון שכבר מימשנו אופרטור זה באופן הבא:

```
(const Element& element) [operator. ]
```

מימוש כזה פוגע ביכולת של UniqueArray להיות תבנית גנרית ומגביל את האפשרויות עבור Element. ניתן להתגבר על הבעיה על ידי מימוש אותה הפונקציונליות על ידי העמסת אופרטור פנוי אחר, או על ידי כתיבת מתודה, למשל:

```
const Element* UniqueArray<Element,Compare>::  
get_ptr_to_elem_for_index (unsigned int i) const
```