

א. כאשר המתרגל האחראי כתב את התרגיל, הוא הגדיר את הממשק הדרוש עבור איטרטורים של התור. תחילה הגדיר את הממשק של המחלקה `Iterator`, וניסה להשתמש במחלקה זו עבור איטרציה על תורים קבועים באמצעות אובייקטים `const` של אותה המחלקה – `const Iterator`. מדוע אובייקטים מסוג זה לא מספיקים כדי לבצע איטרציה על תורים קבועים? כיצד אובייקטים של המחלקה `ConstIterator` פותרים את הבעיה?

על מנת לבצע איטרציה על תור נצטרך להגדיר איטרטור. האיטרטור מורכב מכמה פונקציות שונות שמאפשרות לרוץ על התור (כמו אופרטור * שמשנה את המצביע של האיטרטור ומקדם אותו). לכן נרצה להגדיר איטרטור `const` בשביל תורים קבועים שיאפשר לרוץ על התור מצד אחד אבל מצד שני לא ישנה אותו (כלומר אופרטורים כמו * יהיו גם `const`). במידה ונרצה להגדיר איטרטור רגיל עבור `const` נקבל שגיאה כיוון שהתור לא יודע אם הוא יכול לבטוח באיטרטור שלא ישנה לו ערכים.

ב. ארליך הוא סטודנט בקורס מתמם וכדי לבדוק את המימוש של התור שלו וניסה לבנות תור של אובייקטים מסוג `ofstream`. כאשר ניסה לקרוא לפונקציה `pushBack` על מנת להכניס אובייקט חדש לתור, הקומפילציה נכשלה. מדוע לא ניתן להכניס אובייקט מסוג זה לתור? פרטו.

לא ניתן להכניס לתור מסוג `ofstream` ערכים כיוון שכפונקציה `pushback` ישנה השמה (כלומר שימוש ב `copy constructor`). ההשמה קוראת כאשר אנחנו יוצרים `Node` חדש (כפעולה הבונה של `Node`).

כפי שנלמד בתרגול ל `ofstream` אין `copy constructor`. לכן כאשר נרצה ליצור `Node` מ `ofstream` נקבל שגיאה.

ג. ג'ין יאנג גם הוא סטודנט בקורס מבוא לתכנות מערכות. לחוסר מזלו הוא שכח מהאזהרות שקיבל בתרגול ומימש את המחלקה `Queue` בקובץ `cpp` במקום בקובץ `h`. כאשר ג'ין יאנג ניסה לקמפל את הקוד הקומפילציה נכשלה. באיזה שלב בקומפילציה התהליך נכשל? מדוע הקומפילציה נכשלה דווקא בשלב זה?

ג. `templates` משמשים את המתכנת כאשר יש צורך לבצע תכנות כללי. אם היינו מגדירים את התכנית בקובץ `++c` נפרד. הקומפילר לא היה יכול לגשת גם להגדרת התכנית וגם לשימוש בה בקבצים אחרים ולעשות אינסטנטיאציה. לכן לאחר הידור כל קבצי התוכנית, כאשר הלינקר יקשר בין קבצי ההרצה תתרחש שגיאת `linking` כי לא התרחשה האינסטנטיאציה של סוג מכנה הנתונים.

ד. מוניקה היא סטודנטית שסיימה את הקורס בסמסטר שעבר בהצלחה רבה, והחליטה להשתמש במימוש התור מהתרגיל לפרוייקט צד שהיא מפתחת בשעות הפנאי. במימוש פרוייקט הצד, מוניקה השתמשה בתור של מחזרות המייצגות שמות של סטודנטים אחרים. תארו כיצד ניתן לקבל את האורך של השם הארוך ביותר בתור באמצעות הפונקציה `reduce`.

כשביל לקבל את האורך של השם הארוך ביותר באמצעות `reduce` על מוניקה לשלוח לפונקציה `reduce` תור. משתנה ראשון ופונקציה נוספת. הפונקציה הנוספת תקבל שני ארגומנטים – השם הכי ארוך עד כה (נקרא לו `max name`) והשם שאנחנו רוצים לבדוק (נקרא לו `name`). כל פעם הפונקציה הנוספת תחזיר את השם הארוך ביותר (במידה ו `max name` עדיין יותר ארוך מ `name` אז נחזיר את `max name` אחרת נחזיר את `name`). הפונקציה `reduce` תחזיר את השם הכי ארוך שהתקבל ואז יהיה על מוניקה רק לחשב את האורך של השם. הסיבה שאנחנו לא מחזירים למוניקה ישירות את האורך של השם הכי ארוך (כלומר מספר מסוג `int`) היא שהפונק' `reduce` מחזירה משתנה מסוג `template` ולאו דווקא `int`. לכן לא נוכל להחזיר לה את האורך המקסימלי של המחזרת אלא נחזיר רק את המחזרת הארוכה ביותר. (הסוג של הפונקציה `reduce` נקבע ע"י המשתנה השני שמתקבל הערך הזה הוא לאו דווקא `int`).