

## חלק יבש

א. הסיבה שלא ניתן להגדיר איטרטור רגיל `const` היא שהגדרה כזו תגרום לאיטרטור עצמו להיות קבוע- דבר שלא יאפשר שינוי של השדות הפנימיים בו, לא תאפשר את פעולת ההתקדמות(++ ) ותשבור את פעולתו. מה שאנו רוצים לעשות זה לא איטרטור שהוא קבוע- אלא איטרטור שמצביע עצמו על קבועים- וכשניגש למקום עליו האיטרטור מצביע נקבל ערך `const`.

ב. מניחים עבור הטיפוס הטמפלייטי שיש אופרטור השמה `=`, `copy`, `c'tor`, בנוסף מניחים שיש אופרטור הריסה כלומר `d'tor`. הפוקנציות בהן יש הנחות הן:

```
מניח העתקה והשמה pushBack  
מניח העתקה והשמה Filter  
מניח העתקה והשמה Queue
```

ג. השגיאה שתתקפל בניסיון הקמפול תהיה שגיאת `undefined reference` שתקרה בשלב הלינקינג. הקומפיילר בשלב הקומפילציה יכתוב את הקוד של ההטמפלייטים לפי הטיפוסים הרלוונטיים, אך מה שבקובץ `cpp` לא יכתב- לכן בקריאה לפונקציה בשלב הלינקינג לא יהיה ניתן למצוא את הפונקציה ונקבל שגיאת `undefined reference`.

ד. הסטודנטית צריכה להשתמש ב `function object`, מחלקה שמממשת אופרטור `()` ולכן קריאתה דומה לשל פונקציה.

## נגדיר מחלקה חדשה:

```
class isDivisible
{
    private:
        int m_divisor;
    public:
        bool operator()(int num) const
        {
            return (num % m_divisor) == 0;
        }
}
```

## ונשתמש כך:

```
Queue<T> queue = filter(queue, isDivisible(num));
```