תור <Queue<T>

תור הוא מבנה נתונים שמיועד לאחסן אובייקטים בסדר שבו הם נכנסים. (FIFO - First In, First Out), כלומר, האובייקט שנכנס ראשון הוא גם זה שיוצא ראשון.

תכונות עיקריות של תור:

1. **FIFO (First In, First Out)**: זהו המבנה הבסיסי של תור, כלומר, האיבר הראשון שנכנס לתור יוצא ראשון.
2. דינמי-אפשר להסיר אלמנטים מראש התור ולהכניס איברים לסופו.
3. גנרי-האות T מציינת שה-Queue יכול להכיל אלמנטים מכל סוג (הסוג מצוין על ידי המשתמש).

פעולות במחלקת <Queue<T:

1. ()Queue-פעולה הבונה תור חדש. (ריק)
2. (T value)Insert -הכנסה לתור. מקבלת איבר ומכניסה ל"סוף" התור
3. ()Remove-הסרה מתור. הפעולה מסירה את האיבר מראש התור ומחזירה אותו.
4. ()Head-הצצה לראש התור. הפעולה מחזירה את האיבר מראש התור.
5. ()IsEmpty-פעולה המחזירה true אם התור ריק, ו false אחרת.

שימוש בתור

כדי לקבל את האלמנט הבא בתור-יש להוציא את האיבר הראשון. ז"א, בתור ניתן לקבל בו זמנית איבר אחד בלבד (בלי לשנות את התור)

סריקת תור-בד"כ לא ידוע מס' האיברים בתור, ולכן סריקת תור תתבצע בעזרת לולאת while. תנאי הלולאה יהיה while(!q.IsEmpty()) כאשר q מייצג את שם התור.

למה התנאי הוא כל עוד התור לא ריק? מכיוון שכדי לגשת לאיבר הבא יש להסיר את הקודם.

על מנת לשמור על שלמות מבנה הנתונים, נשתמש בתור עזר בו נשמור את האיברים טרם הסרתם.

לאחר ריקון התור המקורי-נרוקן את תור העזר חזרה לתור המקורי. שימו לב: האיברים יחזרו לתור הקורי בסדר המקורי, בתנאי שרוקנו לחלוטין את התור, ולאחר מכן החזרנו את איבריו. אם נסיר חלק מהאיברים ונחזיר חזרה, הסדר ישתנה. (לכן גם אם מצאנו את מבוקשנו נסיים לרוקן את התור, ונחזיר את כל האיברים יחד).

לדו':

ריקון תור לתור עזר:

```
while(!q.IsEmpty())
        temp.Insert(q.Remove());
```

החזרת האיברים לתור המקורי:

```
while(!temp.IsEmpty())
        q.Insert(temp.Remove());
```