Library Management System

You are tasked with managing a library system where you need to handle books, authors, and users. Follow the steps below to complete the exercise.

**Part 1: Define Interfaces**

1. **Book Interface:**
   * Define an interface Book with the following properties:
     + id: string
     + title: string
     + author: string
     + year: number
     + checkedOutBy (optional): string
2. **User Interface:**
   * Define an interface User with the following properties:
     + id: string
     + name: string
     + email: string
     + borrowedBooks: string[] (array of book IDs)

**Part 2: Using Partial**

1. **Book Update:**
   * Create a type BookUpdate using Partial<Book>. This will be used to represent updates to a book where all properties are optional.
2. **User Update:**
   * Create a type UserUpdate using Partial<User>. This will be used to represent updates to a user where all properties are optional.

**Part 3: Using Omit**

1. **Book Data Transfer Object (DTO):**
   * Create a type BookDTO using Omit<Book, 'checkedOutBy'>. This will be used to transfer book data excluding the checkedOutBy property.

**Part 4: Using Record**

1. **Library Collection:**
   * Create a type LibraryCollection using Record<string, Book>. This will represent a collection of books where the key is the book ID and the value is the Book object.

**Part 5: Using Pick**

1. **Contact Information:**
   * Create a type ContactInfo using Pick<User, 'name' | 'email'>. This will represent the contact information of a user.

**Part 6: Using Readonly**

1. **Read-Only User:**
   * Create a type ReadOnlyUser using Readonly<User>. This will represent a user where all properties are read-only.

**Part 7: Implementing Functions**

1. **Update Book Function:**
   * Write a function updateBook that takes a Book and a BookUpdate, and returns an updated Book object.
2. **Update User Function:**
   * Write a function updateUser that takes a User and a UserUpdate, and returns an updated User object.