PROBLEM STATEMENT:

Write a c code using the concept of balancing trees Consider a book shelf containing the books that should be sorted year wise if the book publishing year is between 2021 to 2023 assign it as new version and arrange it year wise if the book publishing year is between 2018 to 2020 then assign it as older version and arrange the books year wise if books older than 2018 eliminate them.

CODE:

#include <stdio.h>

#include <stdlib.h>

// Structure for a book

struct Book {

char title[100];

int year;

struct Book \*left;

struct Book \*right;

};

// Function to create a new book node

struct Book \*createBook(char title[], int year) {

struct Book \*newBook = (struct Book \*)malloc(sizeof(struct Book));

strcpy(newBook->title, title);

newBook->year = year;

newBook->left = NULL;

newBook->right = NULL;

return newBook;

}

// Function to insert a book into the tree

struct Book \*insertBook(struct Book \*root, char title[], int year) {

if (root == NULL)

return createBook(title, year);

if (year > 2023)

return root; // Discard books published in the future

if (year >= 2021 && year <= 2023) {

// New version book

root->right = insertBook(root->right, title, year);

} else if (year >= 2018 && year <= 2020) {

// Older version book

root->left = insertBook(root->left, title, year);

} else {

return root; // Discard books published before 2018

}

return root;

}

// Function to print the books in ascending order of their publishing year

void printBooks(struct Book \*root) {

if (root == NULL)

return;

printBooks(root->left);

printf("%s (%d)\n", root->title, root->year);

printBooks(root->right);

}

// Function to free the memory allocated for the tree

void freeTree(struct Book \*root) {

if (root == NULL)

return;

freeTree(root->left);

freeTree(root->right);

free(root);

}

int main() {

struct Book \*bookshelf = NULL;

// Insert books into the bookshelf

bookshelf = insertBook(bookshelf, "Book A", 2022);

bookshelf = insertBook(bookshelf, "Book B", 2020);

bookshelf = insertBook(bookshelf, "Book C", 2019);

bookshelf = insertBook(bookshelf, "Book D", 2021);

bookshelf = insertBook(bookshelf, "Book E", 2017); // Will be discarded

// Print the books in ascending order of publishing year

printf("Books on the bookshelf:\n");

printBooks(bookshelf);

// Free the memory allocated for the bookshelf

freeTree(bookshelf);

}