

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки Кафедра обчислювальної техніки

ЗВІТ ДО ПРАКТИЧНОЇ РОБОТИ

з дисципліни

«Технології програмування на C/Embedded»

на тему «Строки»

|  |  |
| --- | --- |
| Перевірив:  Каплунов Артем Володимирович | Виконав:  студент групи ІК-34  Могильний Микола |

Київ 2025

**Строки**

**Мета роботи:** отримати початкові навички роботи з С.

**Послідовність виконання:**

1. Створити додаток який виводить на екран інформацію про серію книг про Гарі Поттера. Інформація о книгах повинна міститися в зв'язаному списку.

# Хід роботи

1. Створити додаток який виводить на екран інформацію про серію книг про Гарі Поттера. Інформація о книгах повинна міститися в зв'язаному списку.

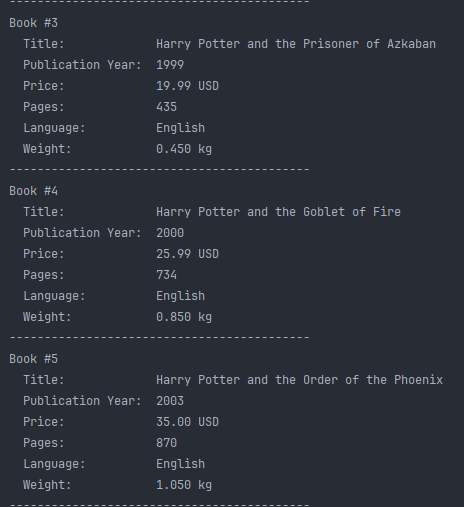
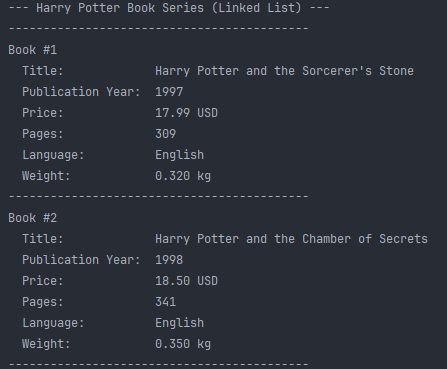
Посилання на GitHub: [Посилання](https://github.com/quantum200?tab=repositories)

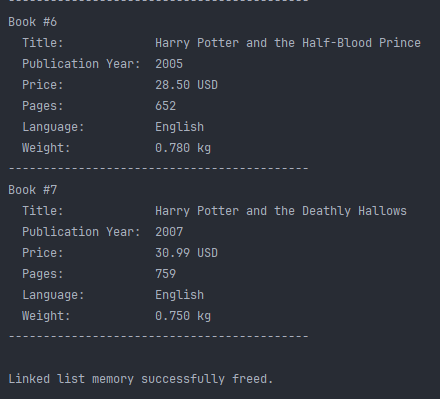
Посилання на мій коміт на GitHub: [Посилання](https://github.com/quantum200/C-Embedden/tree/f95146a8937a5a9e331a7ad0c4d97215c3f9c06d/%D0%9A%D0%9F4)

**Код для розрахунку строки**

#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
#define MAX\_STR\_LEN 50  
  
// --- 1. Book Data Structure Declaration ---  
typedef struct BookData {  
 char title[MAX\_STR\_LEN];  
 float price;  
 int pages;  
 char language[MAX\_STR\_LEN];  
 float weight\_kg;  
 int publication\_year;  
} BookData;  
  
// --- 2. Linked List Node Structure Declaration ---  
typedef struct BookNode {  
 BookData data;  
 struct BookNode \*next;  
} BookNode;  
  
// Global pointer to the start of the list (the head)  
BookNode \*head = NULL;  
  
// --- 3. Linked List Functions ---  
BookNode\* create\_node(BookData book) {  
 BookNode \*newNode = (BookNode\*)malloc(sizeof(BookNode));  
  
 if (newNode == NULL) {  
 perror("Error: Failed to allocate memory for the new node");  
 return NULL;  
 }  
  
 strcpy(newNode->data.title, book.title);  
 newNode->data.price = book.price;  
 newNode->data.pages = book.pages;  
 strcpy(newNode->data.language, book.language);  
 newNode->data.weight\_kg = book.weight\_kg;  
 newNode->data.publication\_year = book.publication\_year;  
 newNode->next = NULL;  
 return newNode;  
}  
  
void insert\_at\_beginning(BookData book) {  
 BookNode \*newNode = create\_node(book);  
 if (newNode == NULL) {  
 return;  
 }  
 newNode->next = head;  
 head = newNode;  
}  
  
/\* Displays the information of all books in the list.\*/  
void display\_list() {  
 BookNode \*current = head;  
 int count = 1;  
  
 printf("\n--- Harry Potter Book Series (Linked List) ---\n");  
  
 if (current == NULL) {  
 printf("The list is empty.\n");  
 return;  
 }  
  
 while (current != NULL) {  
 printf("-------------------------------------------\n");  
 printf("Book #%d\n", count++);  
 printf(" Title: %s\n", current->data.title);  
 printf(" Publication Year: %d\n", current->data.publication\_year);  
 printf(" Price: %.2f USD\n", current->data.price);  
 printf(" Pages: %d\n", current->data.pages);  
 printf(" Language: %s\n", current->data.language);  
 printf(" Weight: %.3f kg\n", current->data.weight\_kg);  
  
 current = current->next; // Move to the next node  
 }  
 printf("-------------------------------------------\n");  
}  
  
/\*\*  
 \* Clears the memory allocated for the entire linked list.  
 \* This is crucial to prevent memory leaks.  
 \*/  
void free\_list() {  
 BookNode \*current = head;  
 BookNode \*next\_node;  
  
 while (current != NULL) {  
 next\_node = current->next;  
 free(current);  
 current = next\_node;  
 }  
 head = NULL;  
 printf("\nLinked list memory successfully freed.\n");  
}  
  
// --- 4. Main Function ---  
int main() {  
 BookData book7 = {"Harry Potter and the Deathly Hallows", 30.99, 759, "English", 0.750, 2007};  
 BookData book6 = {"Harry Potter and the Half-Blood Prince", 28.50, 652, "English", 0.780, 2005};  
 BookData book5 = {"Harry Potter and the Order of the Phoenix", 35.00, 870, "English", 1.050, 2003};  
 BookData book4 = {"Harry Potter and the Goblet of Fire", 25.99, 734, "English", 0.850, 2000};  
 BookData book3 = {"Harry Potter and the Prisoner of Azkaban", 19.99, 435, "English", 0.450, 1999};  
 BookData book2 = {"Harry Potter and the Chamber of Secrets", 18.50, 341, "English", 0.350, 1998};  
 BookData book1 = {"Harry Potter and the Sorcerer's Stone", 17.99, 309, "English", 0.320, 1997};  
  
 insert\_at\_beginning(book7);  
 insert\_at\_beginning(book6);  
 insert\_at\_beginning(book5);  
 insert\_at\_beginning(book4);  
 insert\_at\_beginning(book3);  
 insert\_at\_beginning(book2);  
 insert\_at\_beginning(book1);  
 display\_list();  
 free\_list();  
 return 0;  
}

Результат виконання





**Висновок**

У ході виконання роботи було успішно реалізовано зв'язаний список (Linked List) на мові C для зберігання інформації про серію книг про Гаррі Поттера. Створено структури BookNode та BookData, що є основою зв'язаного списку. Використано динамічне виділення пам'яті (malloc) для гнучкого додавання елементів. Реалізовано функції для додавання та виведення даних списку.