



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

## 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](#)

Посилання на мій коміт на GitHub: [Посилання](#)

## 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

```
--- Harry Potter Book Series (Linked List) ---
-----
Book #1
  Title:      Harry Potter and the Sorcerer's Stone
  Publication Year: 1997
  Price:      17.99 USD
  Pages:      309
  Language:   English
  Weight:     0.320 kg
-----
Book #2
  Title:      Harry Potter and the Chamber of Secrets
  Publication Year: 1998
  Price:      18.50 USD
  Pages:      341
  Language:   English
  Weight:     0.350 kg
-----
Book #3
  Title:      Harry Potter and the Prisoner of Azkaban
  Publication Year: 1999
  Price:      19.99 USD
  Pages:      435
  Language:   English
  Weight:     0.450 kg
-----
Book #4
  Title:      Harry Potter and the Goblet of Fire
  Publication Year: 2000
  Price:      25.99 USD
  Pages:      734
  Language:   English
  Weight:     0.850 kg
-----
Book #5
  Title:      Harry Potter and the Order of the Phoenix
  Publication Year: 2003
  Price:      35.00 USD
  Pages:      870
  Language:   English
  Weight:     1.050 kg
-----
```

```
Book #6
  Title:      Harry Potter and the Half-Blood Prince
  Publication Year: 2005
  Price:      28.50 USD
  Pages:      652
  Language:   English
  Weight:     0.780 kg
-----
Book #7
  Title:      Harry Potter and the Deathly Hallows
  Publication Year: 2007
  Price:      30.99 USD
  Pages:      759
  Language:   English
  Weight:     0.750 kg
-----

Linked list memory successfully freed.
```

## Висновок

У ході виконання роботи було успішно реалізовано зв'язаний список (Linked List) на мові С 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.