## Практические домашние задания для Урока 12 Модуля 2

1. Напишите программу для создания списка книг. Создайте структуру для хранения информации о книге (автор, количество). Создайте структуру “список” с раздельным хранением данных и ссылки на следующий элемент списка. Программа должна запрашивать информацию о 5 книгах и каждый раз добавлять элемент в конец списка. Информация вводится с клавиатуры в формате:

Автор

Количество

Программа должна определять, есть ли искомая книга в списке. Название книги вводится с клавиатуры.

Примеры работы программы:

Пример 1:

Введите фамилию автора: Пушкин

Введите количество: 10

Введите фамилию автора: Лермонтов

Введите количество: 20

Введите фамилию автора: Блок

Введите количество: 5

Введите фамилию автора: Бунин

Введите количество: 12

Введите фамилию автора: Гоголь

Введите количество: 45

Введите название книги для поиска:Блок

Такая книга есть в магазине

Пример 2:

Введите фамилию автора: Пушкин

Введите количество: 10

Введите фамилию автора: Лермонтов

Введите количество: 20

Введите фамилию автора: Блок

Введите количество: 5

Введите фамилию автора: Бунин

Введите количество: 12

Введите фамилию автора: Гоголь

Введите количество: 45

Введите название книги для поиска:Толстой

Такой книги нет в магазине

Для сравнения строковых значений используйте функцию strcmp (const char \*str1, const char \*str2); из модуля string.h  
Функция побайтно сравнивает коды символов двух строк, на которые указывают аргументы функции. Сравнение продолжается до встречи первого отличающегося символа или пока не будут проверены все символы строк.

Если все символы строк совпали, то возвращается 0.

## Решение

1.

#include <stdlib.h>

#include <malloc.h>

#include <cstring>

typedef struct

{ // шаблон данных элемента списка

char name[20]; // название книги

int kol; // количество книг

} InfoBook;

struct List

{ // шаблон элемента списка

InfoBook data;

struct List \*next;

};

struct List \*head; // глобальный указатель – голова списка

// вставка нового элемента в конец списка

struct List \*findNameList()

{

char s[80];

printf("Введите название книги для поиска:");

scanf("%s",s);

struct List \*p = head; // стали на первый элемент списка

while (p != NULL)

{ // пока не дошли до конца списка

if (strcmp(p->data.name, s) == 0) // нашли совпадение

break; // закончили поиск

p = p->next; // перешли к следующему элементу списка

}

if (p != NULL) // элемент найден

printf("Такая книга есть в магазине\n");

else

printf("Такой книги нет в магазине\n");

return p;

}

struct List \*addEndList(InfoBook a)

{

struct List \*add, \*p;

add = (struct List \*)malloc(sizeof(struct List));

if (add)

{ // память выделилась

add->data = a; // информационная часть

add->next = NULL; // новый элемент будет последним

if (head == NULL) // в списке нет элементов

head = add; // вставка первого элемента

else

{

p = head; // становимся на начало списка

while(p->next != NULL) // пока не последний элемент

p = p->next; // переходим к следующему элементу

p->next = add; // вставка нового следом за последним

}

}

return add; // адрес нового элемента или NULL

}

int main()

{

InfoBook a;

int i, n = 5;

struct List \*p;

// создание списка из 5 элементов

head = NULL;

for (i=0; i<n; i++)

{

printf("Введите фамилию автора: ");

scanf("%s", a.name);

printf("Введите количество: ");

scanf("%d",&a.kol);

if (!addEndList(a))

{ // addEndList(a) == NULL

printf("Ошибка создания списка\n"); break;

}

}

findNameList();

// удаление списка с освобождением всей занятой им памяти

while(head != NULL)

{

p = head->next; free(head); head = p;

}

return 0;

}