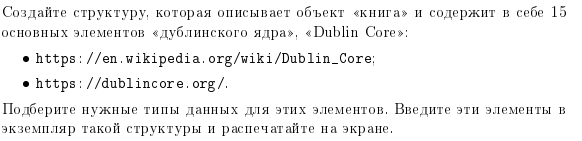
**Лабораторная работа №8**

**Тема лабораторной работы**

Структуры и объединения в языке С

**Задание 1**

**Постановка задачи**



**Математическая модель**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| DublinCore | структура | struct |
| book | экземпляр структуры | struct |
| title | поле структуры | char |
| description | поле структуры |
| date | поле структуры |
| creator | поле структуры |
| contributor | поле структуры |
| rights | поле структуры |
| language | поле структуры |
| relation | поле структуры |
| source | поле структуры |
| type | поле структуры |
| publisher | поле структуры |

**Код программы**

#include <stdio.h>

#include <string.h>

struct DublinCore {

char title[50];

char creator[50];

char subject[100];

char description[300];

char publisher[50];

char contributer[50];

char date[30];

char type[50];

char format[20];

char identifier[100];

char source[50];

char language[100];

char relation[40];

char coverage[50];

char rights[50];

} book;

int main(void) {

strcpy(book.title, "Great Expectations");

strcpy(book.creator, "Charles Dickens");

strcpy(book.subject, "Novel");

strcpy(book.description, "Starting in a graveyard, where the young Pip is accosted by the escaped convict Abel Magwitch.");

strcpy(book.publisher, "Chapman & Hall");

strcpy(book.contributer, "Charles Dickens");

strcpy(book.date, "01.12.1860");

strcpy(book.type, "Novel");

strcpy(book.format, "Book");

strcpy(book.identifier, "ISBN 1-85326-004-5");

strcpy(book.source, "Print");

strcpy(book.language, "English");

strcpy(book.relation, "https://en.wikipedia.org/wiki/Great\_Expectations");

strcpy(book.coverage, "Relevant");

strcpy(book.rights, "Copyright 1860, Charles Dickens. All rights reserved.");

printf("Title: %s\n", book.title);

printf("Creator: %s\n", book.creator);

printf("Subject: %s\n", book.subject);

printf("Description: %s\n", book.description);

printf("Publisher: %s\n", book.publisher);

printf("Contributer: %s\n", book.contributer);

printf("Date: %s\n", book.date);

printf("Type: %s\n", book.type);

printf("Format: %s\n", book.format);

printf("Identifier: %s\n", book.identifier);

printf("Source: %s\n", book.source);

printf("Language: %s\n", book.language);

printf("Relation: %s\n", book.relation);

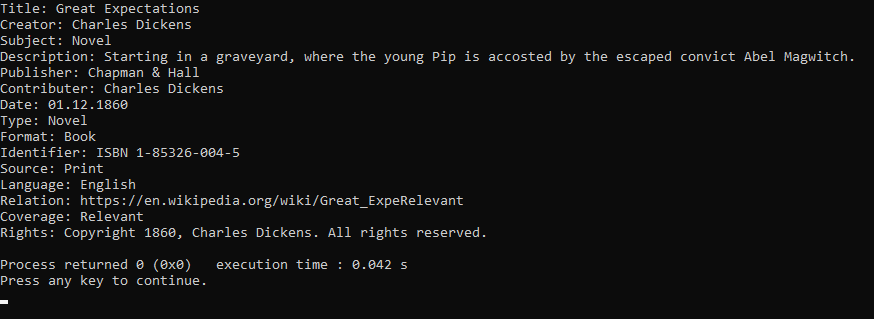
printf("Coverage: %s\n", book.coverage);

printf("Rights: %s\n", book.rights);

return 0;

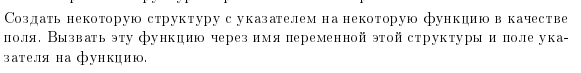
}

**Результат выполненной работы**



**Задание 2**

**Постановка задачи**



**Математическая модель**

s = a + b

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| s | накопительная переменная | int |
| a | аргумент функции | int |
| b | аргумент функции | int |
| x | вводимое с клавиатуры число | int |
| y | вводимое с клавиатуры число | int |
| F | структура | struct |
| \*f | поле структуры | int |

**Код программы**

#include <stdio.h>

int s(int a, int b) {

return a + b;

}

struct F{

int(\*f)(int, int);

}asd;

int main(void) {

asd.f = s;

int x, y;

printf("Write two numbers:\n");

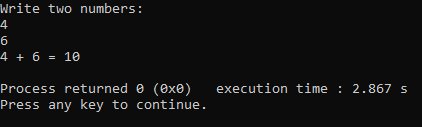
scanf("%d %d", &x, &y);

printf("%d + %d = %d\n", x, y, asd.f(x, y));

return 0;

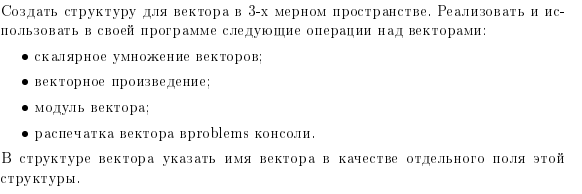
}

**Результат выполненной работы**



**Задание 3**

**Постановка задачи**



**Математическая модель**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| Vector | структура | struct |
| v1 | аргумент функции | struct |
| v2 | аргумент функции | struct |
| x | поле структуры | int |
| y | поле структуры | int |
| z | поле структуры | int |
| scalar | функция | int |

**Код программы**

#include <stdio.h>

#include <math.h>

#include <string.h>

struct Vector {

char name[1];

int x;

int y;

int z;

} v1, v2;

int scalar(struct Vector v1, struct Vector v2) {

return v1.x \* v2.x + v1.y \* v2.y + v1.z \* v2.z;

}

double moduleVector(struct Vector v) {

return sqrt(v.x \* v.x + v.y \* v.y + v.z \* v.z);

}

struct Vector vectorProd(struct Vector v1, struct Vector v2) {

struct Vector v;

v.x = v1.y \* v2.z - v1.z \* v2.y;

v.y = v1.z \* v2.x - v1.x \* v2.z;

v.z = v1.x \* v2.y - v1.y \* v2.x;

return v;

}

int main(void) {

printf("Name and coordinates of the 1st vecror: ");

scanf("%s %d %d %d", &v1.name, &v1.x, &v1.y, &v1.z);

printf("Name and coordinates of the 2nd vecror: ");

scanf("%s %d %d %d", &v2.name, &v2.x, &v2.y, &v2.z);

struct Vector result;

printf("Result of scalar multiplication of vectors: %d\n", scalar(v1, v2));

result = vectorProd(v1, v2);

printf("The result of a vector multiplication of vectors: {%d, %d, %d}\n", result.x, result.y, result.z);

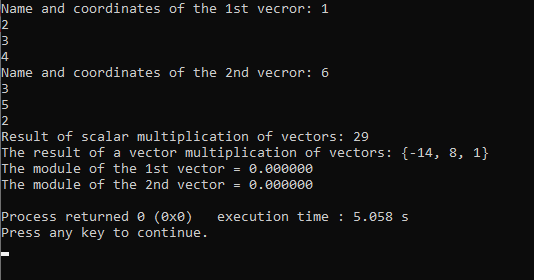
printf("The module of the 1st vector = %lf\n", v1.name, moduleVector(v1));

printf("The module of the 2nd vector = %lf\n", v2.name, moduleVector(v2));

return 0;

}

**Результат выполненной работы**



**Задание 4**

**Постановка задачи**



**Математическая модель**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| ListNode | структура | struct |
| LinkedList | структура | struct |
| \*head | экземпляр структуры | struct |
| \*tail | экземпляр структуры | struct |
| size | размер списка | unsigned int |
| create\_list | функция | struct |
| my\_list | связный список | struct |
| \*prev | экземпляр структуры | struct |

**Код программы**

#include <stdio.h>

#include <string.h>

typedef unsigned int a;

struct ListNode {

char text[50];

struct ListNode \*prev;

struct ListNode \*next;

};

struct LinkedList {

struct ListNode \*head;

struct ListNode \*tail;

a size;

};

struct LinkedList create\_list() {

struct LinkedList list = {

.head = NULL,

.tail = NULL,

.size = 0

};

return list;

}

void push\_to\_end(struct LinkedList \*list, char \*text) {

struct ListNode \*new\_node = (struct ListNode \*)malloc(sizeof(struct ListNode));

strcpy(new\_node->text, text);

new\_node->prev = list->tail;

new\_node->next = NULL;

if (list-> tail == NULL)

list->head = new\_node;

else

list->tail->next=new\_node;

list->tail = new\_node;

list->size += 1;

}

int main(void) {

struct LinkedList my\_list = create\_list();

push\_to\_end(&my\_list, "First element");

push\_to\_end(&my\_list, "Second element");

push\_to\_end(&my\_list, "Third element");

push\_to\_end(&my\_list, "Forth element");

push\_to\_end(&my\_list, "Fifth element");

struct ListNode \*iterator = my\_list.head;

while(iterator != NULL) {

printf("%s: previous = %p; current = %p; next = %p\n", iterator->text, iterator->prev, iterator, iterator->next);

iterator = iterator->next;

}

printf("\n\n");

struct ListNode \*iterator2 = my\_list.tail;

while(iterator2 != NULL) {

printf("%s: previous = %p; current = %p; next = %p\n", iterator2->text, iterator2->prev, iterator2, iterator2->next);

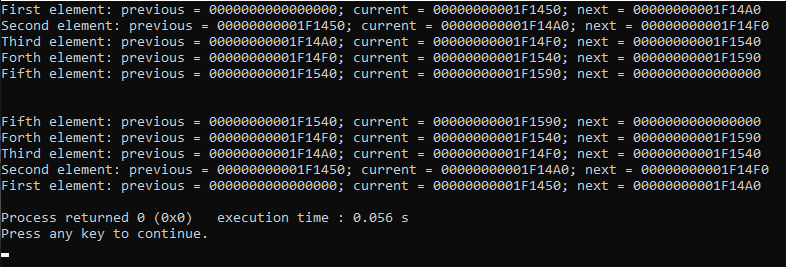
iterator2 = iterator2->prev;

}

return 0;

}

**Результат выполненной работы**



**Часть 2**

**Задание 1**

**Постановка задачи**



**Математическая модель**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| a | поле объединения | int |
| s | поле объединения | char |
| u | объединение | union |

**Код программы**

#include <stdio.h>

#include <string.h>

union u

{

int a;

char s;

};

int main(void) {

union u obj = {'A'};

union u \*point = &obj;

for (int i = 0; i < 10; i++) {

printf("Number = %d, s = %c\n", obj.a, obj.s);

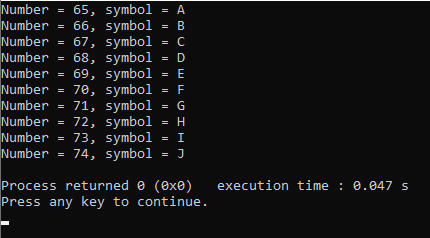
point->a += 1;

}

return 0;

}

**Результат выполненной работы**



**Задание 2**

**Постановка задачи**



**Математическая модель**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| p | объединение | union |
| num | поле объединения | unsigned long |
| str[3] | поле объединения | unsigned char |
| i | параметр цикла | int |

**Код программы**

#include <stdio.h>

#include <string.h>

union p

{

unsigned long num;

unsigned char str[3];

};

int main(void) {

union p obj = { 4294967295 };

for (int i = 0; i < 4; i++)

printf("%d\t", (int)obj.str[i]);

return 0;

}

**Результат выполненной работы**

