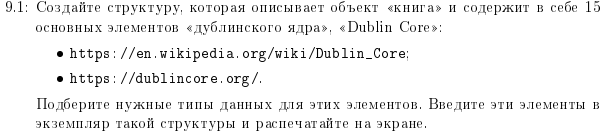
**Лабораторная работа №8 «Структуры и объединения в языке С.**

Задача№1

Постановка задачи:



Списки идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| title | char | Поле структуры |
| creator | char | Поле структуры |
| subject | char | Поле структуры |
| description | char | Поле структуры |
| publisher | char | Поле структуры |
| contributer | char | Поле структуры |
| day | int | Поле структуры |
| month | int | Поле структуры |
| year | int | Поле структуры |
| type | char | Поле структуры |
| format | char | Поле структуры |
| identifier | char | Поле структуры |
| source | char | Поле структуры |
| language | char | Поле структуры |
| relation | char | Поле структуры |
| coverage | char | Поле структуры |
| rights | char | Поле структуры |
| book | struct | Экземпляр структуры |

Код программы:

#include <stdio.h>

#include <string.h>

struct DublinCore {

char title[30];

char creator[30];

char subject[30];

char description[100];

char publisher[30];

char contributer[30];

int day;

int month;

int year;

char type[10];

char format[10];

char identifier[50];

char source[30];

char language[10];

char relation[40];

char coverage[30];

char rights[50];

} book;

int main(void) {

strcpy(book.title, "One Flew Over the Cuckoo's Nest ");

strcpy(book.creator, "Ken Kesey");

strcpy(book.subject, "struggle for freedom");

strcpy(book.description, "one of the main literary works of the movements of hipsters and hippies");

strcpy(book.publisher, "Viking Press и New American Library");

strcpy(book.contributer, "");

book.day = 1;

book.month = 2;

book.year = 1962;

strcpy(book.type, "novel");

strcpy(book.format, "text");

strcpy(book.identifier, "none");

strcpy(book.source, "https://librebook.me/one\_flew\_over\_the\_cuckoo\_s\_nest");

strcpy(book.language, "english");

strcpy(book.relation, "https://www.britannica.com/biography/Ken-Kesey");

strcpy(book.coverage, "Fiction");

strcpy(book.rights, "Copiright belongs to Ken Kesey");

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: %.2d.%.2d.%.4d\n", book.day, book.month, book.year);

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);

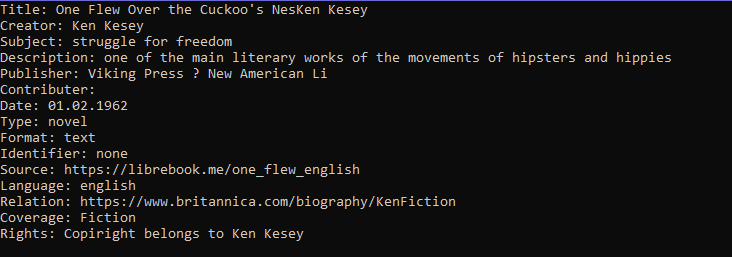
printf("Relation: %s\n", book.relation);

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

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

}

Результат:



Задача№2

Постановка задачи:



Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| factorial | int | Функция факториала |
| K | int | Аргумент функции |
| n | int | Аргумент функции |
| \*f | int | Поле структуры |
| point | struct | Экземпляр структуры |
| a | int | Число |

Код программы:

#include <stdio.h>

#include <stdlib.h>

struct func {

int(\*f)(int);

} point;

int factorial(int n) {

int k = 1;

int i;

for (i=1;i<=n;i++)

k \*= i;

return k;

}

int main(void) {

point.f = factorial;

int a;

printf("Factorial (a):");

scanf("%d",&a);

printf("%d! = %d",a,point.f(a));

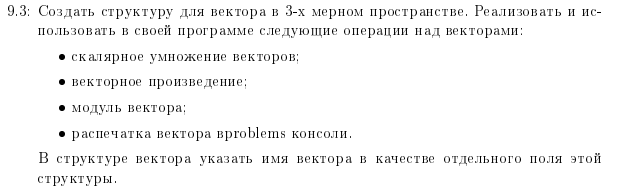
}

Результат:



Задача№3

Постановка задачи:



Математическая модель:

Скалярное произведение векторов А • B = a1b1 + а2b2 + а3b3 представ­ляет собой сумму попарных произведений соответствующих коор­динат исходных векторов.

Векторным произведением векторов АхВ = (а2b3 - b2a3, a3b1 - b3а1, а1b2 - b1a2) является вектор, координаты которого ck вычисляются как ck = ak+1bk+2 — ak+2bk+1 (k=1,2,3) с учетом циклической переста­новки индексов (то есть а4=а1, а5=а2, и так же для компонентов bk).

Модулем вектора является число, равное корню квадратному из скалярного произведения вектора на самого себя: |А| =

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| V | struct | Структура |
| name | char | Поле структуры |
| X | int | Поле структуры |
| Y | int | Поле структуры |
| Z | int | Поле структуры |
| v1 | struct | Экземпляр структуры |
| v2 | struct | Экземпляр структуры |
| Scalar | int | Функция скалярного умножения |
| Module | double | Функция нахождения модуля вектора |
| Prod | struct | Функция векторного умножения |
| v1 | struct | Аргумент функции |
| v2 | struct | Аргумент функции |
| v | struct | Аргумент функции |
| result | struct | Результат векторного произведения |

Код программы:

include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <string.h>

struct V {

char name[1];

int x;

int y;

int z;

}v1,v2;

double module(struct V a) {

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

}

int scalar(struct V a, struct V b) {

return a.x\*b.x+a.y\*b.y+a.z\*b.z;

}

struct V Prod(struct V a, struct V b) {

struct V c;

c.x = a.y\*b.z-a.z\*b.y;

c.y = a.z\*b.x-a.x\*b.z;

c.z = a.x\*b.y-a.y\*b.x;

return c;

}

int main(void) {

printf("Enter the name of 1st vector: ");

scanf("%s",&v1.name);

printf("Enter coordinates of 1st vector:\n");

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

printf("Enter the name of 2st vector: ");

scanf("%s",&v2.name);

printf("Enter coordinates of 2st vector:\n");

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

struct V result;

printf("Scalar production %s \* %s = %d\n",v1.name,v2.name,scalar(v1,v2));

result = Prod(v1,v2);

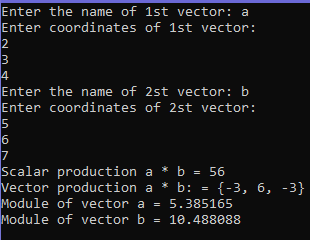
printf("Vector production %s \* %s: = {%d, %d, %d}\n",v1.name,v2.name,result.x,result.y,result.z);

printf("Module of vector %s = %lf\n",v1.name,module(v1));

printf("Module of vector %s = %lf\n",v2.name,module(v2));

}

Результат:



Задача№4

Постановка задачи:



|  |  |  |
| --- | --- | --- |
| Название | Смысл | Тип |
| list | Структура | struct |
| init | Функция | list\* |
| addelem | Функция | list\* |
| deletelem | Функция | list\* |
| listprint | Функция | void |
| listprintr | Функция | void |
| field | поле структуры | int |
| next | поле структуры | \*next |
| prev | поле структуры | \*prev |
| lst | указатель функции/ параметр функции | list\* |
| temp | указатель функции | list\* |
| p | указатель функции | list\* |
| prev | указатель функции | list\* |
| next | указатель функции | list\* |
| head | указатель функции | list\* |
| cur | указатель функции | list\* |
| val | Вводимая переменная | int |

Код программы:

#include <stdio.h>

#include <stdlib.h>

typedef struct list {

int field;

struct list \*next;

struct list \*prev;

}list;

struct list \*init(int a) {

struct list \*lst;

lst = (struct list\*)malloc(sizeof(struct list));

lst->field = a;

lst->next = NULL;

lst->prev = NULL;

return(lst);

}

struct list\*addelem(list \*lst,int number) {

struct list \*temp, \*p;

temp = (struct list\*)malloc(sizeof(list));

p = lst->next;

lst->next = temp;

temp->field = number;

temp->next = p;

temp->prev = lst;

if (p != NULL)

p->prev = temp;

return(temp);

}

struct list \* deletelem(list \*lst) {

struct list \*prev, \*next;

prev = lst->prev;

next = lst->next;

if (prev != NULL)

prev->next = lst->next;

if (next != NULL)

next->prev = lst->prev;

free(lst);

return(prev);

}

void listprint(list \*lst) {

struct list \*p;

p = lst;

do {

printf("%d ", p->field);

p = p->next;

} while (p != NULL);

}

void listprintr(list \*lst) {

struct list \*p;

p = lst;

while (p->next != NULL)

p = p->next;

do {

printf("%d ", p->field);

p = p->prev;

} while (p != NULL);

}

int main() {

int i;

list \*head, \*cur;

int val;

printf("a: ");

scanf("%d", &val);

head = init(val);

cur = head;

for (i = 0; i < 5; i++) {

printf("a: ");

scanf("%d", &val);

cur = addelem(cur, val);

}

printf("Out\n");

listprint(head);

printf("\n");

cur = head->next->next;

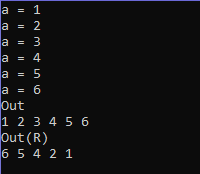
deletelem(cur);

printf("\n");

listprintr(head);

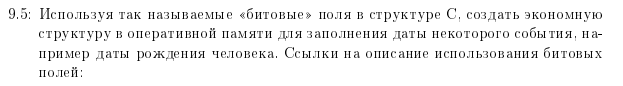
}

Результат:



Задача№5

Постановка задачи:



Код программы:

#include <stdio.h>

#include <stdlib.h>

struct date {

unsigned int day: 5;

unsigned int month: 4;

unsigned int year: 11;

};

int main(void) {

int day, month, year;

printf("Enter day: ");

scanf("%d", &day);

printf("Enter month: ");

scanf("%d", &month);

printf("Enter year: ");

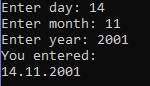
scanf("%d", &year);

struct date point = {day, month, year};

printf("You entered:\n%.2d.%.2d.%d", point.day, point.month, point.year);

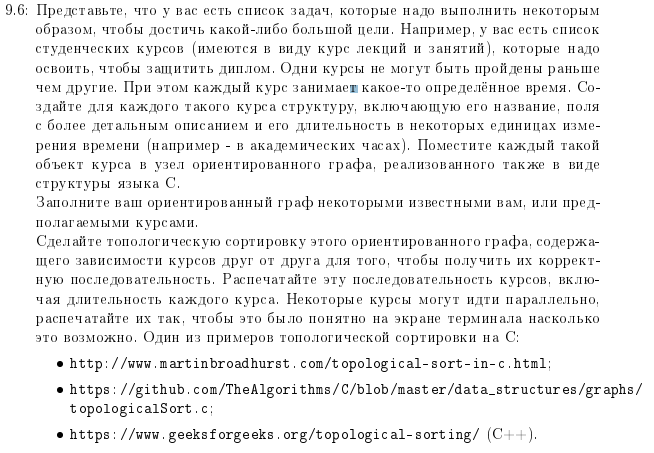
}

Результат:



Задача№6

Постановка задачи:



|  |  |  |
| --- | --- | --- |
| Название | Смысл | Тип |
| x | Массив | int |
| a | Массив | int |
| f | Массив | int |
| e | Массив | int |
| i | Параметр цикла | int |
| k | Параметр цикла | int |
| j | Параметр цикла | int |

Код программы:

#include <stdio.h>

#include <stdlib.h>

int main() {

int k,i,j;

int x[2][2] = { 1,2,3,0 };

int a[2][2] = { 0,0,0,0 };

int b[2][2] = { 0,0,0,0 };

int c[2][2] = { 1,0,0,1 };

for (k=0;k<2;k++) {

for (i=0;i<2;i++) {

for (j=0;j<2;j++) {

a[k][i]=a[k][i]+x[k][j]\*x[j][i];

}

printf("[%d][%d]= %d ",k,i,a[k][i]);

}

}

printf("\n");

for (i=0;i<2;i++) {

for (j=0;j<2;j++) {

f[i][j]=-2\*a[i][j]+5\*x[i][j]+c[i][j]\*9;

printf("[%d][%d]: %d ",i,j,b[i][j]);

}

}

}

Результат:



Задача№7

Постановка задачи:



|  |  |  |
| --- | --- | --- |
| Название | Смысл | Тип |
| Point | Объединение | union |
| Number | Поле объединения | int |
| A | Экземпляр объединения | Pass |

Код программы:

#include <stdio.h>

#include <stdlib.h>

union Point

{

int number;

};

int main() {

union Point a = {10};

union Point \*b = &a;

printf("%d\n", b->number);

a.number \*= 30;

printf("%d\n", b->number);

}

Результат:



Задача№8

Постановка задачи:



Математическая модель:

Код программы:

#include <stdio.h>

#include <stdlib.h>

union Un

{

unsigned long num;

unsigned char s[3];

};

int i;

int main(void) {

union Un point = { 4294967295 };

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

printf("%d ", (int)point.s[i]);

}

Результат:



Задача№9

Постановка задачи:



Математическая модель:

Код программы:

#include <stdio.h>

#include <stdlib.h>

enum week {

Sunday,

Monday,

Tuesday,

Wednesday,

Thursday,

Friday,

Saturday

};

int main(void) {

enum week day1,day2,day3,day4,day5,day6,day7;

day1 = Sunday;

day2 = Monday;

day3 = Tuesday;

day4 = Wednesday;

day5 = Thursday;

day6 = Friday;

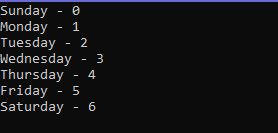
day7 = Saturday;

printf("Sunday - %d\nMonday - %d\nTuesday - %d\nWednesday - %d\n"

"Thursday - %d\nFriday - %d\nSaturday - %d\n",day1,day2,day3,day4,day5,day6,day7);

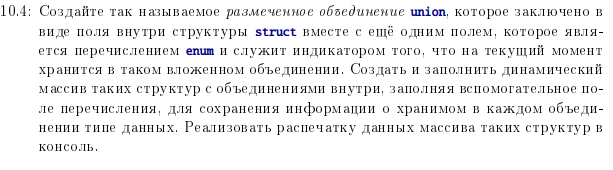
}

Результат:



Задача№10

Постановка задачи:



Математическая модель:

Код программы:

Результат: