Лабораторная работа 8.

Структуры и объединения в языке С.

Комплект 9.

Задание 1.

1. Создайте структуру, которая описывает объект «книга» и содержит в себе 15 основных элементов «дублинского ядра». Подберите нужные типы данных для этих элементов. Введите эти элементы в экземпляр такой структуры и распечатайте на экране.

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| book | Структура «Книга» | struct |
| title[50] | Title — название | char |
| creator[50] | Creator — создатель | char |
| subject[50] | Subject — тема | char |
| description[50] | Description — описание | char |
| publisher[50] | Publisher — издатель | char |
| contributor[50] | Contributor — внёсший вклад | char |
| date[50] | Дата | char |
| type[50] | Type — тип | char |
| format[50] | Format — формат документа | char |
| identifier | Identifier — идентификатор | int |
| source[50] | Source — источник | char |
| language[50] | Language — язык | char |
| coverage[50] | Coverage — покрытие | char |
| copyright[50] | Rights — авторские права | char |
| print(struct book obj) | Функция, выводящая содержимое объекта структуры | void |
| obj | Объект структуры в функции | struct book |
| MandM | Мастер и Маргарита | struct book |



#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

struct book{

char title[50];

char creator[50];

char subject[50];

char description[50];

char publisher[50];

char contributor[50];

char date[10];

char type[50];

char format[50];

int identifier;

char source[50];

char language[50];

char coverage[50];

char copyright[50];

};

void print(struct book obj){

printf("%s\n%s\n%s\n%s\n%s\n%s\n%s\n%s\n%s\n%d\n%s\n%s\n%s\n%s\n",

obj.title, obj.creator,obj.subject, obj.description,

obj.publisher, obj.contributor, obj.date, obj.type,

obj.format, obj.identifier, obj.source, obj.language,

obj.coverage, obj.copyright);

}

int main(){

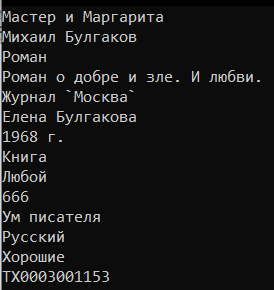
char \*locale = setlocale(LC\_ALL, "Russian");

struct book MandM={"Мастер и Маргарита","Михаил Булгаков","Роман","Роман о добре и зле. И любви.","Журнал `Москва`","Елена Булгакова","1968 г.", "Книга","Любой",666,"Ум писателя","Русский","Хорошие","TX0003001153"};

print(MandM);

return 0;

}



Задание 2.

2. 

|  |  |  |
| --- | --- | --- |
| Имя переменной | Смысл | Тип |
| str | Структура с указателем | struct |
| a | Переменная структуры | int |
| b | Переменная структуры | int |
| \*c | Указатель | int |
| d(str obj) | Функция | int |
| s | Локальная переменная | int |
| objA | Объект структуры | struct str |

#include <stdlib.h>

#include <stdio.h>

#include <locale.h>

struct str

{

int a;

int b;

int \*c;

};

int d(struct str obj)

{

int s;

s = pow(obj.a, obj.b) + 10;

obj.с = &s;

return obj.с;

}

int main ()

{

struct str objA;

objA.a = 3;

objA.b = 7;

objA.c = d(objA);

printf("%d",\*objA.c);

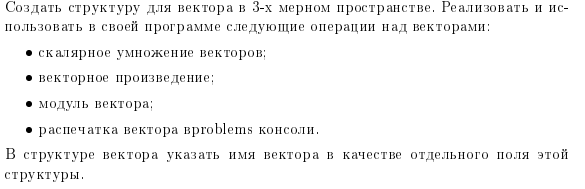
return 0;

}





Задание 3.

1. 

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| vector | Структура обозначающая векторы в 3-х мерном пространстве | struct |
| name | Название вектора | char |
| x | Координата x структуры | int |
| y | Координата y структуры | int |
| z | Координата z структуры | int |
| vlenght | Функция нахождения длины вектора | double |
| skalProd | Функция, находящая скалярное произведение векторов | double |
| objA | Объект структуры функции | struct |
| objB | Объект структуры функции | struct |
| t | Результат функции | double |
| vectProd | Функция, находящая векторное произведение векторов | double |
| lA | Аргумент функции, длина первого вектора | double |
| lB | Аргумент функции, длина второго вектора | double |
| angle | Аргумент функции, угол между двумя векторами | int |
| R | Результат функции | double |
| skalAB | Скалярное произведение векторов | double |
| vectAB | Векторное произведение векторов | double |
| a | Объект a структуры(вектор) | struct vector |
| b | Объект b структуры(вектор) | struct vector |



#include <stdlib.h>

#include <stdio.h>

#include <locale.h>

struct vector{

char name[20];

int x, y, z;

double lenght;

};

double vlenght(struct vector obj){

obj.lenght = sqrt(obj.x \* obj.x + obj.y \* obj.y + obj.z \* obj.z);

return obj.lenght;

}

double skalProd(struct vector objA, struct vector objB){

double t;

t = objA.x \* objB.x + objA.y \* objB.y + objA.z \* objB.z;

return t;

}

double vectorProd(double lA, double lB, int angle)

{

double R;

R = lA \* lB \* sinf(angle);

return R;

}

int main ()

{

char \*locale = setlocale(LC\_ALL, "Russian");

struct vector a, b;

int angle;

printf ("Введите название первого вектора: ");

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

printf("Введите координаты вектора `%s` в формате X Y Z: ", a.name);

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

printf ("Введите название второго вектора:");

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

printf("Введите координаты вектора `%s` в формате X Y Z: ", b.name);

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

printf("Введите значение угла между этими векторами в градусах: ");

scanf ("%f",&angle);

double skalAB,vectAB;

a.lenght = vlenght(a);

b.lenght = vlenght(b);

printf("Длина `%s` = %f\n", a.name, a.lenght);

printf("Длина `%s` = %f\n", b.name, b.lenght);

skalAB = skalProd(a, b);

printf("Scalar product = %f\n",skalAB);

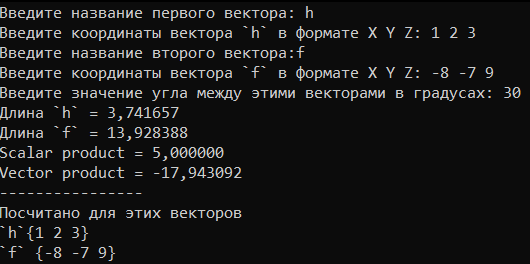
vectAB = vectorProd(a.lenght, b.lenght, angle);

printf("Vector product = %f\n",vectAB);

printf("----------------\nПосчитано для этих векторов\n`%s`{%d %d %d}\n`%s` {%d %d %d}", a.name, a.x, a.y, a.z, b.name, b.x, b.y, b.z);

return 0;

}



Задание 4.







|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| listElem | Структура списка | struct |
| value | Значение элемента списка | int |
| \* prev | Структура предыдущего элемента списка | struct listElem |
| \* next | Структура следующего элемента списка | struct listElem |
| ListElem() | Функция выделения памяти под двунаправленный список | struct listElem\* |
| head | Двунаправленный список | struct listElem\* |
| createRand(list, l) | Функция заполнения списка случайными значениями | struct listElem\* |
| l | Количество элементов в списке | int |
| list | Двунаправленный список, аргумент функции createRand | struct listElem\* |
| i | Параметр цикла | int |
| tmp | Буфер | struct listElem\* |
| \*start | Указатель в функции createRand | struct listElem\* |
| iter | Вспомогательная переменная вывода списка с начала | struct listElem\* |
| biter | Вспомогательная переменная вывода списка с конца | struct listElem\* |



#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

struct listElem

{

int value;

struct listElem \*prev;

struct listElem \*next;

};

struct listElem\* ListElem(){

struct listElem\* head = malloc(sizeof(struct listElem));

head->prev = NULL;

head->next = NULL;

return head;

};

struct listElem\* createRand(struct listElem\* list, int l)

{

struct listElem \*start = list;

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

list->value = rand() % 10 - 5;

list->next = ListElem();

struct listElem\* tmp = list;

list = list->next;

list->prev = tmp;

}

list->next = NULL;

return start;

}

int main()

{

char \*locale = setlocale(LC\_ALL, "Russian");

struct listElem\* head = ListElem();

createRand(head, 10);

printf("Начало -> Конец\n");

struct listElem\* iter = head;

for(int i = 1; iter != NULL; i++)

{

printf(" - Значение элемента %d равно %d\n", i, iter->value);

iter = iter->next;

}

printf ("Конец -> начало\n");

struct listElem\* biter = head;

while(biter->next != NULL){

biter = biter->next;

}

for(int i = 11; biter != NULL; i--){

printf(" - Значение элемента %d равно %d\n", i, biter->value);

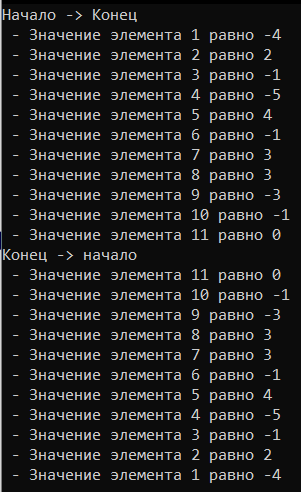
biter = biter->prev;

}

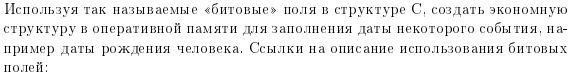
return 0;

}





Задание 5.





|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| birthDate | Структура даты рождения | struct |
| d | День | int |
| m | Месяц | int |
| y | Год | int |
| man | Структура даты рождения конкретного человека | birthDate |



#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <locale.h>

struct birthDate{

int d, m, y;

};

int main()

{

char \*locale = setlocale(LC\_ALL, "Russian");

struct birthDate man;

printf("Введите дату рождения в формате `DD MM YYYY`\n");

scanf("%d %d %d", &man.d, &man.m, &man.y);

printf("Вы ввели дату %d.%d.%d", man.d, man.m, man.y);

return 0;

}



