Лабораторная работа № **15**

«**Разработка алгоритмов и программ с использованием структуры данных struct**»

От учащегося требуется изучить теорию, примеры и выполнить 5 заданий в соответствии с его вариантом по списку группы в журнале по предмету ОАиП.

Структуры данных struct

Структурные переменные (часто называемые просто «структуры») – это объединение одной или более переменных, возможно, разных типов, в одну область памяти, имеющую одно имя.

Структура – упорядоченное множество данных различного типа, которые называются **полями** или **членами** структуры.

Доступ к полю структуры осуществляется по имени переменной типа структуры и имени поля, между которыми ставится точка или «стрелка» **–>**.

Поля структуры могут иметь любой тип кроме void и типа этой же структуры (но поле типа указателя на такую же структуру разрешено).

Структура может содержать только такие поля, длина которых известна компилятору в момент определения структуры.

Структурным типом данных или просто структурой называется тип, описывающий структуру struct.

Синтаксис объявления структурного типа:

```
struct имяСтруктуры
{
    типПоля имяПоля;
    типПоля0 имяПоля0;
    типПоля1 имяПоля1;
    типПоля2 имяПоля2;
    //и так далее создаем (декларируем) поля нужных нам типов
};//в конце декларации структуры ставится фигурная скобка и точка с запятой ;
```

Объявление переменных структурного типа

имяСтруктуры имяПеременной;//переменную типа структуры продекларировали, но еще не проинициализировали
имяСтруктуры имяПеременной = { список значений };//переменную типа структуры продекларировали и сразу проинициализировали строкой значений. Значения в инициализирующей строке должны располагаться **в том же порядке** в каком продекларированы типы и имена полей при декларации соответствующей структуры.

Пример:

```
#include <iostream>
#include <Windows.h>
using namespace std;

int main()
{
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);
    struct emp//объявим новый тип данных - структуру emp (от англ. employee - наёмный работник, служащий)
    {
        int empNo;//поле для хранения табельного номера работника (это целое число)
        char name[80];//поле для хранения имени работника (это символьный массив до 80 символов)
        double salary;//поле для хранения заработка работника за месяц (число рублей и копеек - вещественное число)
    };//конец объявления структуры emp
    emp engineer, teacher, professor;//объявление трех переменных типа структуры emp
    emp professor0 = { 121, "Складовская", 710000.95 };//декларация переменной professor0 типа структуры emp и одновременная инициализация этой переменной строкой значений,
    //что равнозначно коду:
    professor0.empNo = 121;//полю empNo переменной professor0 присвоить значение 121
    strcpy_s(professor0.name, 80, "Складовская");//правильная инициализация символьного массива функцией strcpy_s(), а не professor0.name = "Складовская";//ошибка
    professor0.salary = 710000.95;//полю salary переменной professor0 присвоить значение 710000.95
    emp eng = { 123, "Иванов", 65000 }, teach = { 124, "Петров", 450000 }, prof = { 127, "Сидоров", 790000 };//объявление трех переменных структурного типа emp с одновременной
    //инициализацией каждой переменной строкой значений. Значения в инициализирующей строке значений идут в порядке, как продекларированы поля при декларации структуры emp
    struct myStruct//совмещение декларации структуры myStruct и одновременной декларации трех переменных типа этой структуры
    {
        double re;
        int im;
    } c1, c2, c3;//декларация переменных c1, c2 и c3 типа структуры myStruct

    struct enyStruct//совмещение объявления структуры enyStruct и декларации и одновременной инициализации строками значений трех переменных типа этой структуры enyStruct
    {
        double re;
        int im;
    } x1 = { 0.5, 0 }, x2 = { 1.99, -2 }, x3 = { -6.09, 1 };//вещественное поле переменной x1 инициализируется значением 0.5, а целочисленное поле переменной x1 - нулем
    //аналогично для переменных x2 и x3
    system("pause");
    return 0;
}
```

Доступ к элементам структуры

Прямой доступ к элементам структуры осуществляется через оператор доступа **.** (точка). Это оператор прямого доступа к полям переменной типа структуры. Если у нас указатель на переменную типа структуры или динамический объект, то может понадобиться косвенная адресация, то есть обращение к полю структуры через оператор доступа

косвенной адресации «стрелка вправо» –> (состоит из двух символов – «тире» и «знак больше»).

имяПеременной.имяПоля;//обращение к полю имяПоля у переменной имяПеременной типа структуры struct

Примеры:

```
#include <iostream>
#include <Windows.h>
using namespace std;

int main()
{
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);
    struct emp//объявим новый тип данных - структуру emp (от англ. employee - наёмный работник, служащий)
    {
        int empNo;//поле для хранения табельного номера работника (это целое число)
        char name[80];//поле для хранения имени работника (это символьный массив до 80 символов)
        double salary;//поле для хранения заработка работника за месяц (число рублей и копеек - вещественное число)
    };//конец объявления структуры emp
    emp engineer, teacher, professor;//объявление трех переменных типа структуры emp
    emp professor0 = { 121, "Складовская", 710000.95 };//декларация переменной professor0 типа структуры emp и одновременная инициализация этой переменной строкой значений,
    //что равнозначно коду:
    professor0.empNo = 121;//полю empNo переменной professor0 присвоить значение 121
    strcpy_s(professor0.name, 80, "Перова");//правильная инициализация символьного массива функцией strcpy_s(), а не professor0.name = "Складовская";//ошибка
    professor0.salary = 710000.95;//полю salary переменной professor0 присвоить значение 710000.95
    emp eng = { 123, "Иванов", 65000 }, teach = { 124, "Петров", 450000 }, prof = { 127, "Сидоров", 790000 };//объявление трех переменных структурного типа emp с одновременной
    //инициализацией каждой переменной строкой значений. Значения в инициализирующей строке значений идут в порядке, как продекларированы поля при декларации структуры emp
    struct myStruct//совмещение декларации структуры myStruct и одновременной декларации трех переменных типа этой структуры
    {
        double re;
        int im;
    } c1, c2, c3;//декларация переменных c1, c2 и c3 типа структуры myStruct
    struct enyStruct//совмещение объявления структуры enyStruct и декларации и одновременной инициализации строками значений трех переменных типа этой структуры enyStruct
    {//структуры myStruct и enyStruct для компилятора разные, так как их имена различаются. Поля этих структур нам видятся названными одинаково, но компилятор их различает:
        double re;// myStruct.re и enyStruct.re имеют различные полные имена, myStruct.im и enyStruct.im тоже имеют различные полные имена. Полное имя включает имяСтруктуры.имяПоля
        int im;
    } x1 = { 0.5, 0 }, x2 = { 1.99, -2 }, x3 = { -6.09, 1 };//вещественное поле переменной x1 инициализируется значением 0.5, а целочисленное поле переменной x1 - нулем
    //аналогично для переменных x2 и x3
    engineer.empNo = 34;//поместим значение 34 в поле empNo переменной engineer
    strcpy_s(engineer.name, 80, "Попов");//символьный массив (поле name переменной engineer) инициализируется "безопасной" функцией strcpy_s(). Нельзя: engineer.name = "Попов";
    engineer.salary = 67000.92;
    cout << "Введите номер работника1: ";
    cin >> teacher.empNo;
    cout << "Введите имя работника1: ";
    cin >> teacher.name;//если поле-символьный массив не предполагает пробелов, то для его инициализации пользователем с клавиатуры достаточно cin>>
    //cin.getline(teacher.name, 80);//если поле-символьный массив предполагает внутри себя пробелы, то используем cin.getline(массивКудаПоместитьСтроку, максимальныйРазмерБуфера);
    teacher.salary = engineer.salary / 2;//в поле salary переменной teacher поместим половину от значения, находящегося в поле salary переменной engineer
    cout << "Введите номер работника2: ";//запросим пользователя ввести с клавиатуры номер работника
    cin >> professor.empNo;//считаем с клавиатуры значение, введенное пользователем, и поместим в поле empNo переменной professor
    cout << "Введите полные Фамилию Имя и Отчество работника2: ";
    cin.ignore();//если функция cin.getline() не дает результата, поставьте перед ней функцию cin.ignore() чтобы нейтрализовать "зависший" Enter от предыдущего ввода пользователя
    cin.getline(professor.name, 80);//если поле-символьный массив предполагает внутри себя пробелы, используйте cin.getline(массивКудаПоместитьСтроку, максимальныйРазмерБуфера)
    professor.salary = teacher.salary * 1.5;//в поле salary переменной professor поместим величину в 150% от значения, находящегося в поле salary переменной teacher
    cout << professor0.empNo << "\t|\t" << professor0.name << "\t|\t" << professor0.salary << endl << eng.empNo << "\t|\t" << eng.name << "\t|\t" << eng.salary << endl
        << teach.empNo << "\t|\t" << teach.name << "\t|\t" << teach.salary << endl << prof.empNo << "\t|\t" << prof.name << "\t|\t" << prof.salary << endl
        << engineer.empNo << "\t|\t" << engineer.name << "\t|\t" << engineer.salary << endl << teacher.empNo << "\t|\t" << teacher.name << "\t|\t" << teacher.salary << endl
        << professor.empNo << "\t|\t" << professor.name << "\t|\t" << professor.salary << endl << "--------+--------------+--------------------\n";//распечатаем поля таблицей
    c1.re = 12.5;
    c1.im = -24;
    c2.re = c1.re * c1.re + c1.im * c1.im;
    c2.im = pow(c1.im, 3);
    cout << x1.re << "\t|\t" << x1.im << "\t|\t" << endl << x2.re << "\t|\t" << x2.im << "\t|\t" << endl << x3.re << "\t|\t" << x3.im << "\t|\t" << endl
        << c1.re << "\t|\t" << c1.im << "\t|\t" << endl << c2.re << "\t|\t" << c2.im << "\t|\t" << endl;//распечатаем поля таблицей, используя один cout и "склеивая" значения
    system("pause");
    return 0;
}
```

Тестируем работу программы:

```
D:\2019\Labs\x64\Debug\Lab4_9.exe
Введите номер работника1: 14
Введите имя работника1: Елисей
Введите номер работника2: 16
Введите полные Фамилию Имя и Отчество работника2: Мусин-Пушкин Ян Ольгердович
121     |       Перова  |       710001
123     |       Иванов  |       65000
124     |       Петров  |       450000
127     |       Сидоров |       790000
34      |       Попов   |       67000.9
14      |       Елисей  |       33500.5
16      |       Мусин-Пушкин Ян Ольгердович     |       50250.7
--------+--------------+--------------------
0.5     |       0       |
1.99    |       -2      |
-6.09   |       1       |
12.5    |       -24     |
732.25  |       -13824  |
Для продолжения нажмите любую клавишу . . .
```

Вложенные структуры

Одна структура может иметь внутри себя поля типа других структур, которые к этому моменту должны быть продекларированы, то есть известны компилятору. Структура не может иметь поля типа самой этой структуры, но структура может иметь поля типа указателя на структуру своего типа (такого же типа, как и она сама).

```
#include <iostream>
using namespace std;
int main()
{
    struct emp//декларируется структура emp
    {
        int empNo;
        char name[80];
        double salary;
    };
    struct date//декларируется структура date
    {
        unsigned short int year;
        unsigned short int month;
        unsigned short int day;
    };
    struct formData//декларируется структура formData, поля которой имеют тип структур date и emp, поэтому эти две структуры должны быть объявлены раньше структуры formData
    {
        date birthday;//поле типа date, само состоящее из трех подполей: year, month и day
        emp employment;//поле типа emp, само состоящее из трех подполей: empNo, name и salary
    };
    formData a = { 1933, 5, 19, 123, "Petrova", 456.78 };//инициализация переменной a типа структуры formData, которая состоит из составных полей типа других структур
    formData b = { { 1933, 5, 19 }, { 123, "Petrova", 456.78 } };//инициализация переменной b типа структуры formData с использованием фигурных скобок {}, помогающих отличать
    //внутренние подструктуры внутри этой структуры. Переменные a и b проинициализированы одинаково по результату
    cout << a.birthday.year << "\t|\t" << a.birthday.month << "\t|\t" << a.birthday.day << "\t|\t" << a.employment.empNo << "\t|\t" << a.employment.name << "\t|\t"
        << a.employment.salary << endl << b.birthday.year << "\t|\t" << b.birthday.month << "\t|\t" << b.birthday.day << "\t|\t" << b.employment.empNo << "\t|\t"
        << b.employment.name << "\t|\t" << b.employment.salary << endl;//распечатываем значения в полях. Доступ осуществляется с указания самого большого объкта и далее
    //посредством оператора доступа "точка" проникаем внуть объекта и идем к самому низовому значению (полю, подполю), которое нас интересует
    a.birthday.year = 1961;//обращение к полю year внутри поля birthday у переменной a
    a.birthday.month = 4;//обращение к полю month внутри поля birthday у переменной a
    a.birthday.day = 26;//обращение к полю day внутри поля birthday у переменной a
    b.birthday = a.birthday;//копирование значений из поля birthday пременной a в поле birthday переменной b. Скопируются значения всех подполей (year, month и day)
    //из поля birthday переменной a в соотвествующие по именам подполя поля birthday переменной b
    cout << a.birthday.year << "\t|\t" << a.birthday.month << "\t|\t" << a.birthday.day << "\t|\t" << a.employment.empNo << "\t|\t" << a.employment.name << "\t|\t"
        << a.employment.salary << endl << b.birthday.year << "\t|\t" << b.birthday.month << "\t|\t" << b.birthday.day << "\t|\t" << b.employment.empNo << "\t|\t"
        << b.employment.name << "\t|\t" << b.employment.salary << endl;//повторно распечаем значения полей и подполей, чтобы проверить, какие теперь в них значения
    system("pause");    return 0;   }
```

```
D:\2019\Labs\x64\Debug\Lab5_0.exe
1933    |       5       |       19      |       123     |       Petrova |       456.78
1933    |       5       |       19      |       123     |       Petrova |       456.78
1961    |       4       |       26      |       123     |       Petrova |       456.78
1961    |       4       |       26      |       123     |       Petrova |       456.78
Для продолжения нажмите любую клавишу . . .
```

Массив структур

Типа структур struct можно создавать не только переменные, указатели, ссылки, но и массивы, причем как статические (см. пример ниже), так и динамические. Обращение к полям структуры в массиве осуществляется по правилу «от большого к маленькому (внутреннему)», то есть сначала нужно указать имя массива, индекс элемента в нем, потом имя поля и, если надо, подполя и т.д. Например, если нам надо обратиться к полю Номер у студента номер 3 в массиве Группа, то нужно написать код в таком порядке: Группа[3].Номер. Уровней вложенности может быть и больше (например, Группы будут находится в свою очередь в массиве Колледж), но главное применять правило написания полного последовательного пути доступа от самой крупной сущности и последовательно обращаясь к ее внутренним «слоям», идя к требуемому нам данному. Это напоминает написание адреса: сначала указывается Страна, потом Область, потом Город, потом Улица, потом Номер дома, потом Номер квартиры. Если сразу обратиться к Улице и Номеру дома и Номеру квартиры, то система (робот, алгоритм) не поймет, какая именно улица нас интересует, поскольку в нескольких городах могут быть улицы с таким названием одновременно, да и названия городов могут совпадать (Брест в Беларуси и Брест во Франции), поэтому и надо написать полный путь в порядке слева направо, чтобы компилятор однозначно понимал, к чему ему обращаться, ведь в одной программе можно создать и одновременно несколько массивов из студентов, различаться они будут по имени массива.

```cpp
#include <iostream>
#include <Windows.h>
using namespace std;

int main()
{
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);
    const int m = 15, n = 20, p = 4;//массивы будут статические, поэтому создаем константы для указания размерности статических массивов
    struct student//декларируем новый тип данных структурного типа - student
    {
        int number;//номер студента по списку группы
        char name[m];//имя студента размером до m символов (пробелы НЕ предполагаются)
        char surname[n];//фамилия студента размером до n символов (пробелы НЕ предполагаются)
        double srBall;//поле для хранения среднего балла студента (предполагаются вещественные числа)
    };
    student group[p];//создаем статический массив из p студентов (в массив можно поместить 4 студента или меньше)
    for (int i = 0; i < p; i++)//в цикле проходим по каждому элементу массива, то есть проходим по индексам элементов массива group
    {
        cout << "Введите номер студента: ";//запрашиваем у пользователя нужные данные
        cin >> group[i].number;//считываем с клавиатуры вводимые пользователем данные и помещаем их в нужное поле очередного (i-го по индексу) элемента массива group
        cout << "Введите имя студента: ";
        cin >> group[i].name;//для символьного массива НЕ предполагающего внутри себя пробелы достаточно использовать cin>>
        cout << "Введите фамилию студента: ";
        cin >> group[i].surname;
        cout << "Введите средний балл студента: ";
        cin >> group[i].srBall;
    }
    cout << "Номер\t|\tФамилия\t|\tИмя\t|\tСредний балл\n--------+---------------+---------------+----------------------\n";//печать "шапки" таблицы на консоль
    for (int i = 0; i < p; i++)//в цикле проходим по всем элементам массива group и печатаем содержимое каждого поля каждого элемента массива (то есть каждого студента)
    {
        cout << group[i].number << "\t|\t" << group[i].surname << "\t|\t" << group[i].name << "\t|\t" << group[i].srBall << endl;
    }
    system("pause");
    return 0;
}
```

```
D:\2019\Labs\x64\Debug\Lab5_1.exe
Введите номер студента: 1
Введите имя студента: Ян
Введите фамилию студента: Иванов
Введите средний балл студента: 4.56
Введите номер студента: 2
Введите имя студента: Олег
Введите фамилию студента: Петров
Введите средний балл студента: 5.43
Введите номер студента: 3
Введите имя студента: Стас
Введите фамилию студента: Сидоров
Введите средний балл студента: 6
Введите номер студента: 4
Введите имя студента: Майя
Введите фамилию студента: Зайцева
Введите средний балл студента: 4.58
Номер    |        Фамилия |        Имя      |        Средний балл
--------+---------------+---------------+----------------------
1        |        Иванов  |        Ян       |        4.56
2        |        Петров  |        Олег     |        5.43
3        |        Сидоров |        Стас     |        6
4        |        Зайцева |        Майя     |        4.58
Для продолжения нажмите любую клавишу . . .
```

В памяти элементы массива group будут располагаться последовательно по индексам от нулевого до третьего индекса включительно, а внутри каждого элемента массива (внутри очередного студента) будут последовательно располагаться составляющие его поля в точно таком же порядке, как указано в декларации структуры Студент.

group массив															
group[0] элемент массива				group[1] элемент массива				group[2] элемент массива				group[3] элемент массива			
number	name	surname	srBall	number	name	surname	srBall	number	name	surname	srBall	number	name	surname	srBall
1	**Ян**	**Иванов**	**4.56**	**2**	**Олег**	**Петров**	**5.43**	**3**	**Стас**	**Сидоров**	**6.00**	**4**	**Майя**	**Зайцева**	**4.58**

Таблица – Схематическое расположение экземпляров структуры student в массиве group на 4 студента (их номера (индексы) от № 0 до № 3 включительно

Структура – это набор из нескольких переменных, возможно различных типов, сгруппированных под одним именем для удобства обработки.

```
struct date // объявление нового структурного типа данных, который программист назвал date
{
        int day;
        int month;
        int year;
        int week_day;
        char mon_name[4];
```

```
};//конец объявления нового структурного типа данных, который программист назвал date
date d1, d2; // объявление переменных типа date
d1.day = 20; // доступ к отдельным элементам структуры: поле day у переменной d1 инициализируется
//значением 20
d1.month = 5;
```

Структуры можно присваивать:

```
d2 = d2;//значения из полей переменной типа структуры d2 копируются в соответствующие одноименные
//поля переменной типа структуры d1
```

Структуры нельзя сравнивать целиком, но можно сравнивать значения полей структур:

```
if( d1 == d2 )
{
    cout << "error"; // так писать нельзя
}
```

Структуры могут быть вложенными одна в другую – учетная карточка служащего может фактически выглядеть так:

```
struct person
{
    char name[20];
    char address[50];
    long zipcode; // почтовый индекс
    long ss_number; // код карточки социального обеспечения человека
    double salary; // зарплата
    date birthdate; // дата рождения. Это поле само типа структуры date
    date hiredate; // дата поступления на работу. Это поле само типа структуры date
};
```

Структура person содержит две структуры типа date.

Если мы определим emp как

person emp;

то

emp.birthdate.month //будет ссылаться на месяц рождения служащего emp. Операция указания члена структуры точка "." ассоциируется (читается) слева направо.

Массивы структур

Структуры часто образуют массивы, то есть экземпляры структур помещают в массивы. Чтобы объявить массив структур, вначале необходимо определить саму структуру (то есть определить агрегатный тип данных), а затем объявить переменную массива этого же типа (то есть создать массив типа нашей структуры). Например, чтобы объявить 100-элементный массив структур типа person, который был определен ранее, напишите следующее:

person list[100];//продекларирован массив с именем list, в который можно поместить 100 экземпляров структур типа person, то есть в него можно поместить информацию о ста служащих. Это выражение создаст 100 наборов переменных, каждый из которых организован так, как определено в структуре person, но в конкретное поле каждого из этих элементов массива можно поместить конкретное данное о служащем (например, календарный день рождения служащего № 25 в нашем списке из ста служащих; пусть это будет 16-е число месяца):

```
list[25].birthdate.date = 16;
```

Оператор typedef

Зарезервированное слово typedef образовано от англ. выражения type definition – определение типа, то есть определение нового пользовательского типа, ситуация, когда программист задает «свое» имя для некоторого типа данных. Зарезервированное слово typedef позволяет назначить синоним некоторому типу данных.

Синтаксис оператора typedef:

typedef декларацияТипа названиеСиноним;

Например:

```cpp
#include <iostream>
using namespace std;

typedef double veschestvennoe;//дадим типу double синоним veschestvennoe
typedef int tseloe;//дадим типу int синоним tseloe

const int m = 15, n = 20, p = 4;//создаем константы для указания размерности статических массивов
typedef struct beta     //начало выражения typedef
{
    unsigned int key;
    char name[m];
} mas[n];               //конец выражения typedef

int main()
{
    veschestvennoe a = 2.98;//компилятор знает, что тип "veschestvennoe" - это тоже самое, что и тип double, то есть написан код: double a = 2.98;
    tseloe b = -7;//компилятор знает, что тип "tseloe" - это тоже самое, что и тип int, то есть написан код: int b = -7;
    cout << pow(b, 2) << endl << sqrt(a) << endl;//функции математической библиотеки знают, что b типа int и a типа double, поэтому корректно работают
    mas g;//ввиду переопределения типа ранее, на самом деле мы создали массив из n элементов типа структуры beta
    g[0].key = 100;//поскольку это массив, то обращаемся к его элементам
    strcpy_s(g[0].name, m, "Japan");
    cout << g[0].key << ' ' << g[0].name << endl;
    for (int i = 1; i < n; i++)
    {
        g[i].key = i;
        _itoa_s(pow(i, 3), g[i].name, m, 10);
        cout << g[i].key << ' ' << g[i].name << endl;
    }
    system("pause");
    return 0;
}
```

```
D:\2019\Labs\x64\Debug\Lab5_3.exe
49
1.72627
100 Japan
1 1
2 8
3 27
4 64
5 125
6 216
7 343
8 512
9 729
10 1000
11 1331
12 1728
13 2197
14 2744
15 3375
16 4096
17 4913
18 5832
19 6859
Для продолжения нажмите любую клавишу . . .
```

Объединения union

Объединение – частный случай структуры, все поля которой располагаются по одному и тому же адресу.

Формат описания такой же, как у структуры, только вместо ключевого слова struct используется ключевое слово union.

Длина объединения (размер памяти, выделяемый объединению) равна длине его наибольшего поля. Тип поля в объединении может быть любым, в том числе и структурой struct.

В каждый момент времени в переменной типа «объединение» хранится только одно значение. Объединения применяют для экономии памяти, когда известно, что больше одного поля одновременно не требуется.

Инициализироваться объединение может только первым описанным полем, которым может стать любое из полей объединения, но **только одно** из всех полей объединения.

Основное достоинство объединений – возможность разных трактовок одного и того же содержимого (кода) памяти, то есть когда значение одного поля можно интерпретировать и как значение другого поля объединения, но данные поля должны быть соответствующих «похожих» типов данных, например int, short int, long int, char, bool.

Пример объединения:

```
union myUnion
{
        float f;
        unsigned long k;
};//в экземплярах типа этого объединения myUnion можно заполнить только одно поле: либо f, либо – k
```

Битовые поля

Битовые поля – особый вид полей структуры. Битовые поля часто используются для компактного хранения некоторых типов данных, то есть для экономии места в оперативной памяти. Например, для флажков «да/нет» достаточно 1 бита, а переменная типа bool будет занимать 1 байт = 8 бит. Общий синтаксис описания битового поля:

тип [имяБитовогоПоля]: ширина;//надпись в квадратных скобках [] на метаязыке означает «допустимый, но не обязательный код», то есть имя битовому полю можно дать или не дать на усмотрение разработчика (рекомендуем все же давать имя). Ширина записывается числом в десятичной системе счисления, но обозначает максимальное количество бит, достаточное для хранения максимального значения, записанного в двоичной системе счисления, помещаемого в данное битовое поле. То есть берем максимальное значение, которое может понадобиться поместить в битовое поле, переводим его в двоичную систему счисления, и считаем количество бит, занимаемое этим бинарным значением, - полученное число в десятичном виде записываем как ширину битового поля.

Члены битовых полей могут иметь значения только типов: signed int, unsigned int, signed char или unsigned char.

Битовые поля могут находиться только внутри структуры struct, а не сами по себе.

Пример:

```
struct enyBitStruct
{
        unsigned b1: 1;//поле для хранения одного бита
        unsigned b2: 2;//поле для хранения двух бит
        unsigned b4: 4;//поле для хранения четырех бит
};
```

Массивы битовых полей недопустимы. К битовым полям не может быть применен оператор взятия адреса “&”, так как битовое поле может находиться внутри байта.

ПРИМЕР 1. Структура «комплексное число»

Создать тип для представления комплексного числа. Реализовать ввод-вывод комплексного числа, определить сумму двух комплексных чисел.

```
#include <iostream>
using namespace std;
struct complex//объявление структуры, имитирующей комплексное число
{//структура объявляется до main'а и прототипов функций, то есть структура объявляется как глобальная –
//видная всему коду программы за ней
      float re, im;//продекларированы два вещественных поля
};
complex read();//прототип функции для ввода комплексного числа
void print(complex c);//прототип функции для вывода комплексного числа
complex add(complex, complex);//прототип функции для нахождения суммы двух комплексных чисел

int main()
{
      complex c1, c2, c3;// определение трех комплексных чисел
      c1 = read();// ввод первого комплексного числа
      c2 = read();// ввод второго комплексного числа
      c3 = add(c1, c2);//сложение двух комплексных чисел
      print(c3);// вывод результата на консоль
      system("pause");
      return 0;
}
complex read()// ввод комплексного числа (полное определение функции)
{
       complex c;
       cout << "re = ";
       cin >> c.re;
       cout << "im = ";
       cin >> c.im;
       return c;
}
void print (complex c)// вывод на экран комплексного числа (полное определение функции)
{
       cout << c.re << " + i " << c.im << endl;
}
complex add(complex c1, complex c2)// сложение двух комплексных чисел (полное определение функции)
{
      complex c3;
      c3.re = c1.re + c2.re;
      c3.im = c1.im + c2.im;
      return c3;
}
```

ПРИМЕР 2. Массив структур

Заполнить массив записей, содержащих поля: ФИО, ключ. Вывести значения записей массива на экран.

```
#include <iostream>
using namespace std;

int main()
{
    const int m = 40, n = 3;
    int i;
    struct human
    {
        unsigned int key;
        char name[m];
    } group[n];// объявление массива из трех экземпляров структуры human (массив из трех переменных типа
```

```
//стркутуры human)
    for( i = 0; i < n; i++ )// заполнение полей массива записями (экземплярами структуры human, которые состоят
    {//из своих полей
        cout << "FIO = ";
        cin.getline( group[i].name, m);// ввод ФИО для i-го элемента массива
         cout << "key = ";
         cin >> group[i].key;// ввод значения key для i-го элемента массива
         cin.get();//ожидать нажатия клавиши клавиатуры пользователем
    }
    for( i = 0; i < n; i++ )// вывод значений элементов массива на экран
    {
          cout << "FIO = " << group[i].name;
          cout << "key = " << group[i].key << endl;
    }
    system("pause");
    return 0;
}
```

ПРИМЕР 3. Передача структуры в качестве аргумента. Структура как возвращаемое значение.
Заполнить массив записей, содержащих поля: ФИО, ключ. Вывести значения записей массива на экран.
Пример аналогичен предыдущему, но оформлен с использованием функций.

```
#include <iostream>
using namespace std;

const int m = 40;
struct student// объявление структуры student
{
     unsigned int key;
     char name[m];
};
student read();//прототип функции ввода значений полей структуры student
void print(student);// прототип функции вывода значений полей структуры student

int main()
{
     const int n = 3;
    student group[n];// объявление массива из экземпляров структуры student
    for( int i = 0; i < n; i++ )// заполнение полей массива экземплярами структуры student
    group[i] = read();
    for( int i = 0; i < n; i++ )// вывод значений элементов массива на экран
    {
              print(group[i]);
     }
     system("pause");
     return 0;
}
student read()// ввод значений полей структуры student (полное определение функции)
{
    student stud;
     cout << "FIO = ";
     cin.getline(stud.name, m);// ввод ФИО
     cout << "key = ";
     cin >> stud.key;// ввод значения key
     cin.get();
     return stud;
}
void print(student stud)// вывод значений полей структуры student (полное определение функции)
{
      cout << "FIO = " << stud.name;
      cout << "key = " << stud.key << endl;
```

```
}
```

ПРИМЕР 4. Использование оператора typedef
Заполнить массив из структур, содержащих поля: ФИО, ключ. Вывести значения записей массива на экран.
Пример аналогичен предыдущему, но добавлен оператор typedef.

```
#include <iostream>
using namespace std;

const int m = 40;
struct student// объявление структуры student
{
      unsigned int key;
      char name[m];
};
student read();// ввод значений полей структуры student
void print(student);//вывод значений полей структуры student

int main()
{
      const int n = 3;
      typedef student arr_student[n];// определение типа для массива из структур student
      arr_student group;// объявление массива из структур student
     for( int i = 0; i < n; i++ )// заполнение полей массива из структур student
     {
      group[i] = read();
     }
     for( i = 0; i < n; i++ )// вывод значений элементов массива на экран
     {
       print(group[i]);
     }
      system(“pause”);
      return 0;
}
student read()//функция ввода значений полей структуры student
{
        student stud;
        cout << "FIO = ";
        cin.getline(stud.name, m);// ввод ФИО
        cout << "key = ";
        cin >> stud.key;// ввод значения key
        cin.get();
        return stud;
}
void print(student stud)// вывод значений полей структуры student
{
      cout << "FIO = " << stud.name;
      cout << "key = " << stud.key << endl;
}
```

ПРИМЕР 5. Байтовое представление вещественного числа через объединение union
Написать функцию для вывода на экран байтового представления в ЭВМ вещественного числа в формате float.

```
#include <iostream>
using namespace std;

union Uflc// объявление объединения типа union
{
    float f;//вещественное число размером 4 байта
    unsigned long l;// длинное целое без знака размером 4 байта
    unsigned char c[4];// символьный массив из 4 байт, поскольку один символ занимает 1 байт
};
```

```
void print(Uflc);//вывод (печать) полей объединения

int main()
{
	Uflc FLC = { 1.0 };// инициализация поля в объединении (только одного поля)
	print(FLC);// вывод (печать) полей объединения
	cout<<"Input float number = ";
	cin >> FLC.f;// ввод вещественного числа
	print(FLC);// вывод по байтам представления вещественного числа
	FLC.l = 1;// ввод длинного целого значения
	print(FLC);// вывод полей объединения
	system(“pause”);
	return 0;

}
void print(Uflc FLC)//вывод полей объединения
{
    cout << "float = " << FLC.f << endl;
    cout << "long = " << hex << FLC.l << endl;
    cout << "char = ";
    cout.unsetf(ios::dec);
    cout.setf(ios::hex);
   for( int i = 0; i < 4; i++)
   {
      cout << (unsigned(FLC.c[i])&0xff) << " ";
    }
    cout.unsetf(ios::hex);
    cout.setf(ios::dec);
    cout << endl;
}
```

ПРИМЕР 6. Код символа через объединение union
Формирование кода символа с помощью битовых полей объединения:

```
#include <iostream>
using namespace std;

int main()
{
	union codeUnion
	{
	     char simb;
	     struct z
	    {
	        int x: 5;
	        int y: 3;
	     } hh;
	} cod;
	cod.hh.x = 4;// x = 00100₂
	cod.hh.y = 2;// y = 010₂
	cout << cod.simb;// ‘D’ (код = 01000100₂ == 68₁₀ == 44₁₆)
	system(“pause”);
	return 0;
}
```

ПРИМЕР 7. Битовое представление числа
Написать функцию для вывода на экран битового представления в ЭВМ числа в формате unsigned char.

```
#include <iostream>
using namespace std;
void binary(unsigned char ch);//прототип функции вывода на экран двоичного представления байта-параметра
```

```
int main()
{
	unsigned c;
	cin >> c;// вводим число 0…255
	if( c < 256 )
	{
		binary(c);// битовое представление числа с
	}
	system(“pause”);
	return 0;
}
void binary(unsigned char ch)//вывод на экран двоичного представления байта-параметра
{
	union myByteChar
	{
		unsigned char ss;
		struct myStruct
		{
			unsigned a0: 1;
			unsigned a1: 1;
			unsigned a2: 1;
			unsigned a3: 1;
			unsigned a4: 1;
			unsigned a5: 1;
			unsigned a6: 1;
			unsigned a7: 1;
		} byte;
	} cod;
cod.ss = ch;//занесение в объединение значения параметра
cout << "byte = ";
cout << " " << cod.byte.a7 << " " << cod.byte.a6 << " " << cod.byte.a5 << " " << cod.byte.a4// вывод бит
	<< " " << cod.byte.a3 << " " << cod.byte.a2 << " " << cod.byte.a1 << " " << cod.byte.a0 << endl;
}
```

Перечисления enum

Перечисление — это набор именованных целочисленных констант. Например, перечисление, в котором приведены названия монет, используемые в США (1 доллар = 100 центов): penny (пенни, монета в один цент), nickel (никель, монета в пять центов), dime (монета в 10 центов), quarter (25 центов, четверть доллара), half-dollar (полдоллара), dollar (монета-доллар).

Перечисления определяются во многом так же, как и структуры: началом объявления перечислимого типа служит ключевое слово enum. Перечисление в общем виде выглядит так:

enum имяПеречисления {список целыхКонстант} список переменныхТипаПеречисления;

Здесь имяПеречисления и список переменныхТипаПеречисления не являются обязательными, но хотя бы что-то одно из них должно присутствовать. Следующий фрагмент кода определяет перечисление с именем coin (монета):

```
enum coin { penny, nickel, dime, quarter, half_dollar, dollar };
```

Имя перечисления можно использовать для объявления переменных данного перечислимого типа. Вот код, в котором money (деньги) объявляется в качестве переменной типа coin:

```
enum coin money;
```

С учетом этих объявлений верными являются следующие выражения:

```
money = dime;
if(money == quarter)
{
	cout << "Денег всего четверть доллара.\n";
}
else
{
	cout << "Всех денег НЕ четверть доллара.\n";
}
if(money == dime)
{
```

```
        cout << “Есть 10 центов всего.\n”;
}
```

Главное, что нужно знать для понимания перечислений — каждый их элемент представляет собой целое константное число. В таком виде элементы перечислений можно применять везде, где используются целые числа. По умолчанию каждому элементу перечисления дается значение, на единицу большее, чем у его предшественника. Первый элемент перечисления имеет по умолчанию значение 0, если мы не присвоили ему явно другое значение. Поэтому, при выполнении кода:

```
cout << penny << “ “ << dime << endl;//на экран будет выведено 0 2.
```

Для одного или более элементов перечисления можно указать значение, используемое как инициализатор. Для этого после перечислителя (члена перечисления) надо поставить знак присвоения, а затем — целое значение (инициализатор). Перечислителям, которые идут после инициализатора, присваиваются значения, большие предшествующего на единицу и так далее. Например, следующий код присваивает перечислителю quarter значение 100:

```
enum coin { penny, nickel, dime, quarter = 100, half_dollar, dollar };
```

В результате следующие значения появятся у элементов перечисления:

```
penny       0
nickel      1
dime        2
quarter     100
half_dollar 101
dollar      102
```

Например, чтобы выводить название монеты, вид которой находится в money, потребуется следующий код:

```
switch(money)
{
        case penny:
        {
                cout <<"пенни\n";
                break;
        }
        case nickel:
        {
                cout <<"никель\n";
                break;
        }
        case dime:
        {
                cout <<"десятицентовик\n";
                break;
        }
        case quarter:
        {
                cout <<"четверть доллара\n";
                break;
        }
        case half_dollar:
        {
                cout <<"полдоллара\n";
                break;
        }
        case dollar:
        {
                cout <<"доллар\n";
                break;
        }
        default:
        {
                cout << “Это не монета из США.\n”;
                break;
        }
}
```

Запомнить определения и примеры

В массиве можно хранить данные только одного типа, например, целочисленный массив может хранить только целые числа и ничего другого. Но в жизни есть много объектов (например, человек), у которых разные характеристики: фамилия – массив символов, возраст (лет) – целое число, вес (в кг) – вещественное число, наличие высшего образования – да (true) или нет (false) и иные характеристики, которые можно описать базовыми типами данных. Но это разнотипные данные, и хранить их как элементы одного массива нельзя, для этого используют структуры данных struct. Структура данных struct – это создаваемый программистом составной объект из логически связанных данных разного типа, объединенных в группу с одним именем. Входящее в структуру данное (переменная, массив) называется полем, т.е. структура состоит из полей. В памяти все поля структуры располагаются вместе, рядом, в том порядке, в каком мы их продекларировали. Структура занимает размер памяти, равный сумме размеров всех ее полей.

Синтаксис объявления структуры данных такой:

```
struct имяСтруктуры
{
        тип поле1;
        тип поле2;
        тип поле3;
        …
};
```

Создадим структуру «Человек»:

```
struct Chelovek
{
        char familia[20];//фамилия размером до 19 букв включительно или меньше
        int age;//возраст (лет)
        double weight;//вес (кг)
        bool highEducation;//0 или false – нет, 1 или true – есть высшее образование
        //аналогично создаем другие требуемые нам поля
};
```

Теперь можно создавать переменные типа структуры Человек, и у каждой такой переменной будут поля для хранения фамилии, возраста, веса и статуса наличия высшего образования.

Chelovek a;//переменная типа структуры Человек декларирована (создана в оперативной памяти, под нее выделено место), но не проинициализирована (ее полям не присвоены значения)

a = {“Smirnov”, 19, 61.5, false};//полям переменной a присвоены значения строкой инициализации явно.

К полям переменной типа структуры обращаться через синтаксис « имяСтруктуры**.**имяПоля »:

a.age = 25;//полю Возраст переменной a присвоить значение 25 (лет) явно в коде

cin >> a.age;//считать с клавиатуры значение, вводимое пользователем, и присвоить его полю Возраст переменой a

cout << a.age;//распечатать на консоль значение, хранимое в ***настоящий момент*** в поле возраст переменной a

Структуры можно помещать в массивы. Каждый элемент такого массива будет структурой, которая, в свою очередь, состоит из полей. Такой массив должен иметь тип структуры:

Chelovek komanda[15];//декларируем статический массив из 15 элементов типа структуры Человек

Желая обратиться к полю элемента массива структур, следует сначала указать имяМассива с индексом требуемого элемента, потом имяПоля, разделяя их оператором доступа «точка»:

komanda[0].age = 37;//присвоить полю возраст самого первого элемента массива (его номер 0) значение 37 (лет)

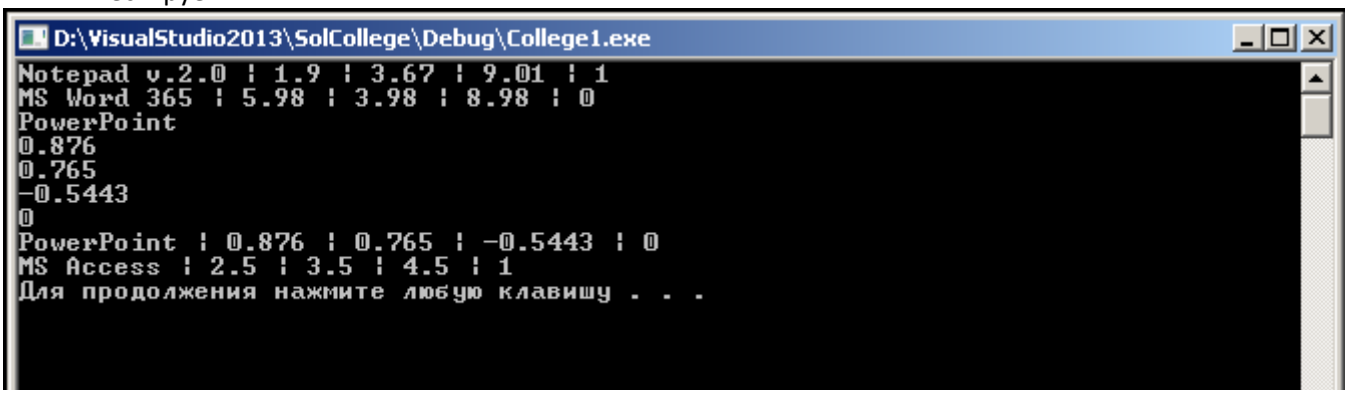
Работать с массивом структур следует в цикле. Если поле структуры само является массивом, то обращение к его конкретному элементу делается строго иерархично: имяМассива структур с индексом «точка» полеМассив с индексом:

cout << komanda[14].familia[0];//распечатать первую букву фамилии пятнадцатого члена команды (его номер 14 если считать от нуля включительно).

Далее показано создание и работа со структурой Компьютерная Программа, состоящей из полей: названиеПрограммы, поле-массив из 3-х вещественных чисел и логическое поле-статус Включено/выключено.

```cpp
#include <iostream>
using namespace std;

int main()
{
    struct programma
    {
        char name[15];//название из 9 символов
        double mas[3];
        bool vklucheno;
    };
    programma a = { "Notepad v.2.0", { 1.9, 3.67, 9.01 }, true };
    cout << a.name << " | " << a.mas[0] <<" | " << a.mas[1] <<" | "<< a.mas[2]<<" | " << a.vklucheno << endl;

    programma c;
    strcpy_s( c.name, "MS Word 365");//присвоить c.name строку символов явно нельзя. Воспользуемся функцией копирования строки 2 в втроку 1
    c.mas[0] = 5.98;
    c.mas[1] = 3.98;
    c.mas[2] = 8.98;
    c.vklucheno = false;
    cout << c.name << " | " << c.mas[0] << " | " << c.mas[1] << " | " << c.mas[2] << " | " << c.vklucheno << endl;

    programma b;
    cin >> b.name;//вводить только одно слово без пробелов (размером до 14 символов) на английской раскладке
    cin >> b.mas[0];
    cin >> b.mas[1];
    cin >> b.mas[2];
    cin >> b.vklucheno;//0-выключено; 1-включено
    cout << b.name << " | " << b.mas[0] << " | " << b.mas[1] << " | " << b.mas[2] << " | " << b.vklucheno << endl;

    programma *w = &a;//указателю на структуру присвоим адрес структуры a (то есть будем изменять значения полей структуры a через указатель w)
    strcpy_s((*w).name, "MS Access");
    (*w).mas[0] = 2.5;//аналогичный результат у записи  w->mas[0] = 2.5;
    (*w).mas[1] = 3.5;
    (*w).mas[2] = 4.5;
    (*w).vklucheno = true;
    cout << (*w).name << " | " << (*w).mas[0] << " | " << (*w).mas[1] << " | " << (*w).mas[2] << " | " << (*w).vklucheno << endl;
    system("pause");
    return 0;
}
```

Тестируем:



Задания по варианту:

1	1	Создать структуру Автомобиль, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
2	1	Создать структуру Спортсмен, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые

		введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
3	1	Создать структуру Мотоцикл, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
4	1	Создать структуру Компьютер, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
5	1	Создать структуру Дом, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
6	1	Создать структуру Завод, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».

	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
7	1	Создать структуру Государство, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
8	1	Создать структуру Город, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
9	1	Создать структуру Кинофильм (или киносериал), имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
10	1	Создать структуру Самолет, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой.

		Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
11	1	Создать структуру Фирма, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
12	1	Создать структуру Наручные часы, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
13	1	Создать структуру Растение, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
14	1	Создать структуру Автомастерская, имеющую поля типа int, double, char, bool, статический строковый массив

		(из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
15	1	Создать структуру Аэропорт, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
16	1	Создать структуру Работник, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
17	1	Создать структуру Корабль, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через

		ссылку на переменную.
18	1	Создать структуру Обувь, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
19	1	Создать структуру Детская игрушка, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
20	1	Создать структуру Одежда, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
21	1	Создать структуру Бизнесмен, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).

	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
22	1	Создать структуру Смартфон, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
23	1	Создать структуру Животное (зверь), имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
24	1	Создать структуру Материк (континент), имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
25	1	Создать структуру Ноутбук, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной,

		обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
26	1	Создать структуру Улица, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
27	1	Создать структуру Магазин, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
28	1	Создать структуру Планета, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
29	1	Создать структуру Музыкальная композиция (песня), имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы).

		Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.
30	1	Создать структуру Спортивная команда, имеющую поля типа int, double, char, bool, статический строковый массив (из символов) и статический массив числового типа. Придумать подходящие по смыслу имена полей и их типы.
	2	Создайте переменную типа вашей структуры и проинициализируйте ее явно инициализирующей строкой. Создайте переменную типа вашей структуры и в коде присвойте значения ее полям, обращаясь к каждому из них отдельно. Распечатайте значения полей переменных через « | ».
	3	Создайте переменную типа вашей структуры и обеспечьте инициализацию ее полей значениями, которые введет пользователь с клавиатуры (в ответ на соответствующие фразы-приглашения вашей программы). Распечатайте значения полей переменной через « | ».
	4	Создайте переменную типа вашей структуры и указатель на нее. Проинициализируйте поля переменной, обращаясь к ним только через указатель. Распечатайте значения полей переменой, обращаясь к ним через указатель на переменную. Помните, что при работе с указателем на структуру, обращение к ее полям будет не через «точку» (переменная**.**поле), а через «тире и знак «больше»» (указатель**–>**поле).
	5	Создайте переменную типа вашей структуры и ссылку на нее. Проинициализируйте поля переменной, обращаясь к ним только через ссылку. Распечатайте значения полей переменой, обращаясь к ним через ссылку на переменную.