

## «Разработка алгоритмов и программ с использованием структур данных union и enum»

Функция – это последовательность операторов, оформленная таким образом, что ее можно вызвать по имени из любого места программы. Функция описывается следующим образом:

```
типВозвращаемогоЗначения имяФункции(тип входнойПараметр0, тип входнойПараметр1, тип
входнойПарметр2)
{
    //операторы (тело функции)
    return возвращаемоеЗначение;
}
```

Из функции можно с помощью return вернуть **одно** значение. Если нужно вернуть еще значения, то сначала следует в функции main() создать переменные и передать нашей функции их адреса, чтобы наша пользовательская функция могла изменить значения соответствующих переменных в main() напрямую. Пусть наша пользовательская функция находит максимальное значение элемента двумерного массива и его индексы. Само значение возвратим из функции через return, а i-ый и j-ый индексы вернем через ссылки, поскольку функция примет в качестве входных параметров адреса двух переменных из main(), которые созданы для того, чтобы функция напрямую поместила в них высчитанные ею значения индексов. Передавать значения можно как через ссылки, так и через указатели, ведь они оба хранят адреса переменных, а значит, могут обеспечивать изменение значения переменной напрямую. Получая **простые** переменные в виде входных параметров, наша пользовательская функция на самом деле получит **копии** переданных ей из main() значений, а потому она может с этими копиями работать, но оригиналы значений этих переменных в main() ей изменить нельзя. Пользовательская функция находит значения и передает их функции main(), которая будет их печатать на консоль. **Сама** пользовательская функция на консоль ничего **не** печатает.

```
1 #include <iostream>
2 #include <time.h>
3 #include <stdlib.h>
4 using namespace std;
5 double findMax(double** mas, int m, int n, int& iMax, int& jMax); //прототип функции пользователя findMax()
6
7 int main()
8 {
9     setlocale(LC_CTYPE, "Russian");
10    srand(time(NULL));
11    int m0, n0;
12    do
13    {
14        cout << "Введите количество строк и столбцов: ";
15        cin >> m0 >> n0;
16    } while (m0 <= 0 | n0 <= 0); //цикл выполняется до тех пор, пока пользователь не введет положительные значения для количества строк и столбцов
17    double** arr = new double*[m0]; //создадим массив из вещественных случайных чисел
18    if (arr == NULL)
19    {
20        cout << "Память не выделена на строки.\n";
21        system("pause");
22        return 0;
23    }
24    for (int i = 0; i < m0; i++)
25    {
26        arr[i] = new double[n0];
27        if (arr[i] == NULL)
28        {
29            cout << "Память не выделена на столбец №" << i << ".\n";
30            system("pause");
31            arr = NULL;
32            return 0;
33        }
34        for (int j = 0; j < n0; j++) //если курсор дошел до этой строки, значит память на i-тый столбец выделена и можно заполнить строку значениями
35        {
36            cout << (arr[i][j] = (rand() % 1100) / 11.0) << ' '; //инициализируем элемент значением и СРАЗУ его печатаем на консоль
37        }
38        cout << endl; //переводим курсор в конце строки на следующую строку
39    }
40    int iStrok, jStolb; //создадим переменные для хранения индексов максимального значения элемента. Ниже функция findMax() примет значения адресов этих переменных
41    double m = findMax(arr, m0, n0, iStrok, jStolb); //вызов функции findMax(). Она return'ом возвращает значение максимального элемента, который сохраняем в переменную m
42    cout << "Максимальное значение в массиве: " << m << " у элемента с индексами[" << iStrok << "][" << jStolb << "].\n"; //значения iStrok и jStolb уже изменены функцией
43    system("pause");
44    return 0;
45 }
46 //функция возвращает значение максимального элемента массива (в массиве типа double тип найденного максимального значения будет ТОЖЕ double)
47 double findMax(double** mas, int m, int n, int& iMax, int& jMax) //функция принимает указатель на массив, две переменные и две ссылки
48 {
49     double max = mas[0][0]; //объявим максимальным первый элемент
50     iMax = 0; //индекс по строкам этого "максимального" элемента 0
51     jMax = 0; //индекс по столбцам этого "максимального" элемента 0
52     for (int i = 0; i < m; i++)
53     {
54         for (int j = 0; j < n; j++)
55         {
56             if (mas[i][j] > max) //если очередной элемент массива больше "максимального"
57             {
58                 max = mas[i][j]; //то максимальным элементом сделаем этот элемент
59                 iMax = i; //индекс по строкам очередного максимального элемента присвоим соответствующей ссылке, тем самым изменив значение переменной iStrok в main'e()
60                 jMax = j; //индекс по столбцам очередного максимального элемента присвоим соответствующей ссылке, тем самым изменив значение переменной jStolb в main'e()
61             }
62         }
63     }
64     return max; //возвратим из функции значение найденного максимального элемента
65 }
```

```
D:\VisualStudio2013\SolCollege\Debug\College3.exe
Введите количество строк и столбцов: 5
9
16.5455 55.2727 65.5455 98.0909 51 23.3636 54.4545 44.9091 36.6364
33.6364 16 70.1818 98.5455 61 17.0909 59.7273 59.9091 55.6364
37.9091 45.0909 5.90909 28.6364 70.7273 57 83.7273 67.6364 77.0909
48.3636 85.6364 58.0909 43 90.0909 91.7273 82.0909 20.2727 8.45455
13.9091 73.1818 63.1818 48.2727 95.3636 32.9091 95.1818 7.90909 19
Максимальное значение в массиве: 98.5455 у элемента с индексами[1][3].
Для продолжения нажмите любую клавишу . . .
```

## Рекурсивные пользовательские функции

Функция может вызывать сама себя, то есть в определении функции в ее теле есть код вызова функции с этим же именем (то есть ее самой). Такая функция называется рекурсивной. Рекурсивная функция – функция, которая вызывает сама себя явно или косвенно. Явная рекурсия происходит, когда функция вызывает **сама себя** из своего тела. Косвенная рекурсия происходит, когда функция из своего тела вызывает **другую функцию**, а та, в свою очередь, вызывает эту функцию или вызывает другие функции, цепочка вызовов которых приводит к вызову нашей изначальной функции. Для решения некоторых задач требуется рекурсия, поскольку она позволяет пошагово, «по кусочкам», решать задачу и, в конце концов, прийти к конечному результату. Если рекурсия к конечному результату не приводит, то есть ее вызов бесконечный, зацикленный, – это неверно написанная рекурсивная функция. При рекурсивном вызове функции код ее тела не дублируется, а работает тот же самый, но с другими входными параметрами (принимаемыми значениями). Эти значения добавляются в стек (динамическая структура данных, в которую объекты добавляются с одного конца и из него же поочередно извлекаются). При бесконечной рекурсии стек, располагающийся в динамической области оперативной памяти, переполнится, и работа программы будет аварийно завершена операционной системой. Рекурсия часто потребляет много оперативной памяти.

Например, создадим рекурсивную функцию для нахождения факториала числа (математическая запись  $x!$ ). Факториал числа – это произведение всех целых чисел от 1 до данного числа включительно. Факториал нуля есть единица. Факториал отрицательного числа не вычисляется, но можно поменять знак у отрицательного числа.

```
1 #include <iostream>
2 using namespace std;
3 long long int factorial(long long int x); //прототип рекурсивной функции расчета факториала длинного целого числа
4
5 int main()
6 {
7     setlocale(LC_CTYPE, "Russian");
8     long long int x; //на данном ПК факториал корректно рассчитывается до числа 20 включительно. Расчет больших значений приводит к переполнению ячейки памяти типа long long int
9     cout << "x: "; //поэтому полезно сделать проверку вводимого числа, меньше ли оно 21
10    cin >> x;
11    long long int y = factorial(x); //рекурсивная функция вызывается один раз, а ее самовыводы будут происходить в процессе выполнения кода ее тела.
12    cout << "Факториал числа " << x << " равен " << y << endl; //возвращенный функцией factorial() результат выше сохранили в переменной y, и теперь используем ее
13    //чтобы увидеть переполнение ячеек памяти даже типа long long int, расчитаем возрастающие числа и их факториалы в столбик
14    for (long long int i = 0; i < 35; i++)
15    {
16        cout << i << "\t" << factorial(i) << endl;
17    }
18    system("pause");
19    return 0;
20 }
21
22 long long int factorial(long long int x) //рекурсивная функция возвращает целочисленное значение
23 {
24     if (x < 0)
25     {
26         x = -x; //если число отрицательное - поменяем ему знак
27     }
28
29     if (x == 0 | x == 1) //условие, при котором рекурсивные вызовы функции надо завершить
30     {
31         return 1; //важно, чтобы рекурсивная функция приходила к концу и завершалась возвращением некоторого значения, а не вызовом себя снова
32     }
33     else
34     {
35         return x * factorial(x - 1); //такой возврат из функции обеспечивает умножение текущего числа на факториал числа, на единицу меньшего данного числа
36     } //и так будет продолжаться до тех пор, пока текущее число не станет равным единице
37 }
```

```
D:\VisualStudio2013\SolCollege\Debug\College4.exe
x: -5
Факториал числа -5 равен 120
0      1
1      1
2      2
3      6
4     24
5    120
6   720
7  5040
8 40320
9 362880
10 3628800
11 39916800
12 479001600
13 6227020800
14 87178291200
15 1307674368000
16 20922789888000
17 355687428096000
18 6402373705728000
19 121645100408832000
20 24329020081766400000
21 -4249290049419214848
22 -1250660718674968576
23 8128291617894825984
24 -7835185981329244160
25 7034535277573963776
26 -1569523520172457984
27 -5483646897237262336
28 -5968160532966932480
29 -7055958792655077376
30 -8764578968847253504
31 4999213071378415616
32 -6045878379276664832
33 3400198294675128320
34 4926277576697053184
Для продолжения нажмите любую клавишу . . .
```

Задание: написать функцию для нахождения результата у математической функции Аккермана для двух неотрицательных чисел  $m$  и  $n$ . Протестировать работу функции на значениях от 0 до 4. Функция Аккермана( $m, n$ ) принимает два параметра  $m$  и  $n$ , и представляет собой систему функций:

Если  $m == 0$ , то вернуть значение  $n+1$ ; (здесь конец рекурсивных вызовов)

Если  $m > 0$  и при этом  $n == 0$ , то вызвать функцию Аккермана с параметрами  $m-1$  и 1;

Если  $m > 0$  и при этом  $n > 0$ , то вызвать функцию Аккермана с параметрами  $m-1$  и функцию Аккермана с параметрами  $m$  и  $n-1$ .

$$A(m, n) = \begin{cases} n + 1 & m = 0 \\ A(m - 1, 1) & m > 0, n = 0 \\ A(m - 1, A(m, n - 1)) & m > 0, n > 0 \end{cases}$$

```
1 #include <iostream>
2 using namespace std;
3 int Akkerman(int m, int n);
4
5 int main()
6 {
7     setlocale(0, "Russian");
8     int m0, n0;
9     do
10     {
11         cout << "Введите m и n: ";
12         cin >> m0 >> n0; //протестируйте программу вводом пар чисел от 0 и 0 до 4 и 4. Сделайте вывод. Математически доказано, что функция Аккермана имеет конец
13     }
14     while (m0 < 0 || n0 < 0); //по условию задачи m0 и n0 числа неотрицательные, поэтому требуем ввести именно неотрицательные два числа
15     int x = Akkerman(m0, n0); //передаем два числа функции Аккермана и сохраняем возвращаемый этой функцией результат в переменной x
16     cout << "Результат: " << x << endl; //печатаем результат
17     system("pause");
18     return 0;
19 }
20 int Akkerman(int m, int n) //функция Аккермана принимает два целых числа и возвращает целое число
21 {
22     if (m == 0) //ветка выполнения №1
23     {
24         cout << " 1:\t" << m << "\t" << n << endl; //по условию задачи печатать номер выполняющейся ветки не требовалось, но это нужно для наблюдения за внутренней работой рекурсивной функции
25         return n + 1; //кроме номера исполняющейся ветки распечатаем выше и значения m и n в столбик
26     }
27     else
28     {
29         if (m > 0 & n == 0) //ветка выполнения №2
30         {
31             cout << " 2:\t" << m << "\t" << n << endl;
32             return Akkerman(m - 1, 1);
33         }
34         if (m > 0 & n > 0) //ветка выполнения №3
35         {
36             cout << " 3:\t" << m << "\t" << n << endl;
37             return Akkerman(m - 1, Akkerman(m, n - 1)); //функция рекурсивно вызывает себя, причем один из параметров, передаваемый новому вызову функции сам являлся вызовом такой же функции
38         }
39     } //не все случаи учтены и заканчиваются возвратом значений. Если жк0... Но фильтр отрицательных значений помещен в main() до вызова данной функции
40 }
```

```
D:\VisualStudio2013\SolCollege\Debug\College5.exe
Введите m и n: 2
3
3:      2      3
3:      2      2
3:      2      1
2:      2      0
3:      1      1
2:      1      0
1:      0      1
1:      0      2
3:      1      3
3:      1      2
3:      1      1
2:      1      0
1:      0      1
1:      0      2
1:      0      3
1:      0      4
1:      0      5
1:      0      6
3:      1      7
3:      1      6
3:      1      5
3:      1      4
3:      1      3
3:      1      2
3:      1      1
2:      1      0
1:      0      1
1:      0      2
1:      0      3
1:      0      4
1:      0      5
1:      0      6
1:      0      7
1:      0      8
Результат: 9
Для продолжения нажмите любую клавишу . . .
```

### Определение перечисления enum

При написании программ часто возникает потребность определить несколько именованных констант, для которых требуется, чтобы все они имели различные целые значения. Для этого удобно воспользоваться перечисляемым типом данных.

Формат:

```
enum[ имяТипа ] { имяКонстанты0, имяКонстанты1, имяКонстанты2, ... };
```

Имя типа задается в том случае, если в программе требуется определять переменные этого типа. Компилятор обеспечивает, чтобы эти переменные принимали значения только из списка констант.

Константы **должны** быть **целочисленными** и могут инициализироваться обычным образом. При отсутствии инициализатора первая константа обнуляется, а каждой следующей присваивается на 1 большее значение, чем предыдущей:

```
enum Err { ERR_READ, ERR_WRITE, ERR_CONVERT};
```

Константам ERR\_READ, ERR\_WRITE, ERR\_CONVERT присваиваются значения 0, 1 и 2 соответственно по умолчанию. Либо мы присваиваем им целые значения посредством оператора присвоения « = ».

Создадим в нашей функции перечисление для нумерации ветвей выполнения рекурсии 1, 2, 3.

```
20 int Akkerman(int m, int n)//функция Аккермана принимает два целых числа и возвращает целое число
21 {
22     enum Branch { br1 = 1, br2 = 2, br3 = 3 };//создадим перечисление из трех констант для нумерации ветвей выполнения и присвоим им целые значения, иначе по умолчанию присвоится 0, 1, 2
23     if (m == 0)//ветка выполнения №1
24     {
25         //ниже для печати номера ветки выполнения используем не символ "1", а константу br1 с таким же значением из перечисления Branch
26         cout << br1 << ":t" << m << "t" << n << endl;//по условию задачи печатать номер выполняющейся ветки не требовалось, но это нужно для наблюдения за внутренней работой рекурсивной функции
27         return n + 1;//кроме номера исполняющейся ветки распечатаем выше и значения m и n в столбик
28     }
29     else
30     {
31         if (m > 0 & n == 0)//ветка выполнения №2
32         {
33             cout << br2 << ":t" << m << "t" << n << endl;//для печати номера ветки2 используем не символ "2", а константу br2 с таким же значением из перечисления Branch
34             return Akkerman(m - 1, 1);
35         }
36         if (m > 0 & n > 0)//ветка выполнения №3
37         {
38             cout << br3 << ":t" << m << "t" << n << endl;//для печати номера ветки3 используем не символ "3", а константу br3 с таким же значением из перечисления Branch
39             return Akkerman(m - 1, Akkerman(m, n - 1));//функция рекурсивно вызывает себя, причем один из параметров, передаваемый новому вызову функции сам является вызовом такой же функции
40         }
41     }
42 }
```

**Задания 1, 2 и 3** реализовать с помощью массивов, структур struct, перечислений enum, объединений union и битовых полей в зависимости от заданной предметной области и требований задания.

**В Задании 1 реализовать поиск в массиве из элементов типа пользовательских структур struct по двум полям.**

1	1	Отдел кадров. Ф.И.О. работника, образование, специальность, подразделение, должность, оклад, дата трудоустройства на предприятие. Выбор по должности, стажу работы. Даты реализовать с помощью битового поля, должность – с помощью перечисления.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о животных (живых существах): <ul style="list-style-type: none"> <li>· птицы (вид, вес, водоплавающие (да/нет), дальность полета);</li> <li>· рыбы (вид, цвет (4 цвета), ареал обитания (4 ареала));</li> <li>· рептилии (вид, страна происхождения, длина).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места - вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму): <ul style="list-style-type: none"> <li>◆ масть карты;</li> <li>◆ достоинство карты;</li> <li>◆ цвет карты.</li> </ul>
2	1	Горожанин. Ф.И.О., дата рождения, адрес, пол (м, ж), социальный статус (новорожденный, ребенок, подросток, юноша, девушка, взрослый, пожилой). Выборка по полу и году рождения. Дату рождения реализовать с помощью битового поля, социальный статус – с помощью перечисления.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о платежах: <ul style="list-style-type: none"> <li>· наличными (вид валюты, сумма);</li> <li>· по карточке (тип карточки (4 типа), № карточки, фамилия, сумма);</li> <li>· SMS-banking (оператор, банк, № карточки, № телефона, сумма).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет - вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму): <ul style="list-style-type: none"> <li>◆ год (с 1900 по 2100);</li> <li>◆ месяц;</li> <li>◆ день месяца.</li> </ul>
3	1	Ученики. Ф.И.О., класс (цифра + буква), предметы, оценки, средний балл. Выбор по фамилии, выбор по среднему баллу. Класс реализовать с помощью битовых полей, предметы – через перечисление.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о клиентах: <ul style="list-style-type: none"> <li>· студенты (№ зачетки, фамилия, пол(м/ж), средний балл);</li> <li>· преподаватели (табельный номер, фамилия, должность (4 должности));</li> <li>· слушатели (№ паспорта, фамилия, специальность (4 специальности)).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет - вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму): <ul style="list-style-type: none"> <li>◆ номер поезда (4 десятичных цифры);</li> <li>◆ время отправления;</li> <li>◆ тип поезда (4 типа).</li> </ul>

4	1	Клиенты банка. Ф.И.О., тип счета (срочный, льготный, до востребования, кредитный и т. д.), номер счета, сумма на счете, дата последнего изменения. Выбор по номеру счета, по диапазону суммы (<100, >100). Дату реализовать с помощью битового поля, тип счета – с помощью перечисления.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о товарах: <ul style="list-style-type: none"> <li>· в наличии (номенклатурный №, наименование, единицы измерения (штуки, кг, упаковки), цена);</li> <li>· на складе (№ склада, № стеллажа, тип контейнера (белый, синий, красный));</li> <li>· под заказ (№ заказа, фамилия, дата).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места - вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму): <ul style="list-style-type: none"> <li>◆ цвет (16 цветов);</li> <li>◆ длина (до 17 цифр);</li> <li>◆ тип (3 типа).</li> </ul>
5	1	Личная библиотека. Автор книги, название, издательство, количество страниц, раздел библиотеки (специальная, справочная, историческая, научная литература, хобби, домашнее хозяйство, беллетристика и т. д.), происхождение (покупка, находка, подарок, обмен), номер комнаты, стеллажа и полки стеллажа, где хранится книга, наличие книги в данный момент (есть или нет). Выбор книг по автору, году; инвентаризация библиотеки (вывод всего списка книг по категориям). Происхождение книги реализовать с помощью перечисления. Номер комнаты, стеллажа и полки стеллажа, где хранится книга, реализовать посредством битовых полей.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о фигурах: <ul style="list-style-type: none"> <li>· треугольник (три стороны);</li> <li>· ромб (две диагонали);</li> <li>· окружность (радиус).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места - вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму): <ul style="list-style-type: none"> <li>◆ пол (м/ж);</li> <li>◆ возраст;</li> <li>◆ рост.</li> </ul>
6	1	Справочник автомобилей. Марка автомобиля, цвет, заводской номер, дата выпуска, тип кузова (седан, универсал и т. д.), дата последнего техосмотра, владелец. Выбор транспортных средств по владельцу или номеру. Формирование приглашений на техосмотр в соответствии со сроком. Дату выпуска и последнего техосмотра реализовать с помощью битового поля, марку – с помощью перечисления.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о предметах искусства: <ul style="list-style-type: none"> <li>· картина (размеры (длина, ширина), материал (холст, дерево, бумага));</li> <li>· скульптура (размеры, материал (мрамор, бронза, гипс));</li> <li>· ювелирное изделие (вес, материал (придумайте сами), стоимость).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места - вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):

		<ul style="list-style-type: none"> <li>◆ марка автомобиля (10 вариантов);</li> <li>◆ объем двигателя;</li> <li>◆ пробег.</li> </ul>
7	1	Склад. Наименование товара, цена, количество, процент торговой надбавки (5, 10, 15, 20, 25, 30%), номер цеха хранения, номер ряда, номер стеллажа и номер полки на стеллаже, где хранится товар. Выбор по наименованию, цене. Вывод всего списка товаров на складе с расчетом сумм. Процент торговой надбавки реализовать с помощью перечисления. Номер цеха хранения, номер ряда, номер стеллажа и номер полки на стеллаже, где хранится товар, реализовать посредством битовых полей.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о комплектующих: <ul style="list-style-type: none"> <li>· монитор (диагональ, ЖК/ЭЛТ, разрешение (4 варианта));</li> <li>· клавиатура (количество клавиш, русификация (да/нет));</li> <li>· мышь (производитель, количество кнопок, наличие провода (да/нет), оптическая/механическая).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму): <ul style="list-style-type: none"> <li>◆ марка мотоцикла (10 вариантов);</li> <li>◆ год выпуска (с 1990-2009);</li> <li>◆ объем двигателя.</li> </ul>
8	1	Авиарейсы. Номер рейса, пункт назначения, время вылета, дата вылета, стоимость билета, количество мест. Выбор по пункту назначения, дате вылета. Дату вылета реализовать с помощью битового поля, пункт назначения – с помощью перечисления.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о недвижимости: <ul style="list-style-type: none"> <li>· комната (площадь, к-во соседей, свободна(да/нет));</li> <li>· квартира (район (4 варианта), количество комнат, площадь, этаж);</li> <li>· дом (район (4 варианта), площадь застройки, жилая площадь, процент готовности).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму): <ul style="list-style-type: none"> <li>◆ сигнал светофора (красный, желтый, зеленый);</li> <li>◆ количество полос (до 6);</li> <li>◆ допустимая скорость автомобиля.</li> </ul>
9	1	Вокзал. Номер поезда, пункт назначения, дни следования, время прибытия, время отправления. Выбор по пункту назначения, дате. Вывод расписания по времени. Дни следования реализовать с помощью перечисления (названия дней недели). Время выезда и прибытия реализовать с помощью битового поля (часы, минуты).
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о транспорте: <ul style="list-style-type: none"> <li>· автомобиль (марка (4 варианта), грузоподъемность, количество пассажиров);</li> <li>· поезд (тип поезда (курьерский, скорый, пассажирский), количество вагонов, скорость);</li> <li>· самолет (модель (4 варианта), к-во пассажиров, скорость).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:



		<p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ номер группы (6 цифр);</li> <li>◆ номер студента в группе (2 цифры);</li> <li>◆ пол (м/ж).</li> </ul>
10	1	Государство. Наименование, столица, численность населения, площадь, фамилия президента, количество крупнейших городов, количество областей, количество регионов. Выбор государства по названию, занимаемой площади (> заданного значения). Форму правления реализовать с помощью перечисления. Количество крупнейших городов, количество областей, количество регионов реализовать посредством битовых полей.
	2	<p>Объединения и перечисления.</p> <p>Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о животных (живых существах):</p> <ul style="list-style-type: none"> <li>· птицы (вид, вес, водоплавающие (да/нет), дальность полета);</li> <li>· рыбы (вид, цвет (4 цвета), ареал обитания (4 ареала));</li> <li>· рептилии (вид, страна происхождения, длина).</li> </ul> <p>Создать массив из 5 таких структур.</p> <p>Написать программу, выполняющую следующие действия:</p> <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	<p>Протестируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:</p> <p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ масть карты;</li> <li>◆ достоинство карты;</li> <li>◆ цвет карты.</li> </ul>
11	1	Ломбард. Фамилия клиента, наименование товара, оценочная стоимость, сумма, выданная под залог, дата сдачи, срок хранения. Выбор товаров по истечении срока хранения, по наименованию товара. Дату сдачи реализовать с помощью битового поля; наименование (категорию) товара реализовать посредством перечисления.
	2	<p>Объединения и перечисления.</p> <p>Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о платежах:</p> <ul style="list-style-type: none"> <li>· наличными (вид валюты, сумма);</li> <li>· по карточке (тип карточки (4 типа), № карточки, фамилия, сумма);</li> <li>· SMS-banking (оператор, банк, № карточки, № телефона, сумма).</li> </ul> <p>Создать массив из 5 таких структур.</p> <p>Написать программу, выполняющую следующие действия:</p> <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	<p>Протестируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:</p> <p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ год (с 1900 по 2100);</li> <li>◆ месяц;</li> <li>◆ день месяца.</li> </ul>
12	1	Записная книжка. Ф.И.О, дата рождения, адрес, телефон, место работы или учебы, должность. Автоматическое формирование поздравления с днем рождения (по текущей дате). Поиск по фамилии. Дату рождения реализовать с помощью битового поля; должность – с помощью перечисления.
	2	<p>Объединения и перечисления.</p> <p>Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о клиентах:</p> <ul style="list-style-type: none"> <li>· студенты (№ зачетки, фамилия, пол(м/ж), средний балл);</li> <li>· преподаватели (табельный номер, фамилия, должность (4 должности));</li> <li>· слушатели (№ паспорта, фамилия, специальность (4 специальности)).</li> </ul> <p>Создать массив из 5 таких структур.</p> <p>Написать программу, выполняющую следующие действия:</p> <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Протестируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:



		<p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ номер поезда (4 десятичных цифры);</li> <li>◆ время отправления;</li> <li>◆ тип поезда (4 типа).</li> </ul>
13	1	Студенты. Ф.И.О., дата поступления, специальность, группа, факультет, средний балл. Выбор по году поступления, фамилии, среднему баллу, группе. Дату поступления реализовать с помощью битового поля, факультет – с помощью перечисления.
	2	<p>Объединения и перечисления.</p> <p>Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о товарах:</p> <ul style="list-style-type: none"> <li>· в наличии (номенклатурный №, наименование, единицы измерения (штуки, кг, литры, упаковки), цена);</li> <li>· на складе (№ склада, № стеллажа, тип контейнера (белый, синий, красный));</li> <li>· под заказ (№ заказа, фамилия, дата).</li> </ul> <p>Создать массив из 5 таких структур.</p> <p>Написать программу, выполняющую следующие действия:</p> <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	<p>Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:</p> <p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ цвет (16 цветов);</li> <li>◆ длина (до 17 цифр);</li> <li>◆ тип (3 типа).</li> </ul>
14	1	Список клиентов гостиницы. Паспортные данные, даты приезда и отъезда, номер, тип размещения (люкс, одноместный, двухместный, трехместный, апартаменты). Поиск постояльца по дате приезда или по фамилии. Даты приезда и отъезда реализовать с помощью битового поля, тип размещения – с помощью перечисления.
	2	<p>Объединения и перечисления.</p> <p>Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о фигурах:</p> <ul style="list-style-type: none"> <li>· треугольник (три стороны);</li> <li>· ромб (две диагонали);</li> <li>· окружность (радиус).</li> </ul> <p>Создать массив из 5 таких структур.</p> <p>Написать программу, выполняющую следующие действия:</p> <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	<p>Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:</p> <p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ пол (м/ж);</li> <li>◆ возраст;</li> <li>◆ рост.</li> </ul>
15	1	Справочник абитуриента. Наименование вуза, адрес, перечень специальностей, конкурс в последние 4 года по каждой специальности, размер оплаты при платном обучении. Выбор по разным критериям: все о данном вузе; все о данной специальности, поиск минимального конкурса по данной специальности. Конкурсы за последние 4 года по каждой специальности реализовать через битовые поля, перечень специальностей – через перечисления.
	2	<p>Объединения и перечисления.</p> <p>Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о предметах искусства:</p> <ul style="list-style-type: none"> <li>· картина (размеры (длина, ширина), материал (холст, дерево, бумага));</li> <li>· скульптура (размеры, материал (мрамор, бронза, гипс));</li> <li>· ювелирное изделие (вес, материал (придумайте сами), стоимость).</li> </ul> <p>Создать массив из 5 таких структур.</p> <p>Написать программу, выполняющую следующие действия:</p> <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:

		<p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ марка автомобиля (10 вариантов);</li> <li>◆ объем двигателя;</li> <li>◆ пробег.</li> </ul>
16	1	<p>Преподаватели. Фамилия преподавателя, название экзамена, преподаваемые предметы, дата экзамена. Выбор по фамилии, дате экзамена. Дату экзамена реализовать с помощью битового поля; преподаваемые предметы – с помощью перечисления.</p>
	2	<p>Объединения и перечисления.</p> <p>Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о комплектующих:</p> <ul style="list-style-type: none"> <li>· монитор (диагональ, ЖК/ЭЛТ, разрешение (4 варианта));</li> <li>· клавиатура (количество клавиш, русификация (да/нет));</li> <li>· мышь (производитель, количество кнопок, наличие провода (да/нет), оптическая/механическая).</li> </ul> <p>Создать массив из 5 таких структур.</p> <p>Написать программу, выполняющую следующие действия:</p> <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	<p>Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:</p> <p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ марка автомобиля (10 вариантов);</li> <li>◆ год выпуска (с 1990-2009);</li> <li>◆ объем двигателя.</li> </ul>
17	1	<p>Авиарейсы. Для каждого рейса указаны номер рейса, тип самолета, пункт назначения, время вылета. Время вылета реализовать с помощью битового поля, тип самолета – с помощью перечисления.</p>
	2	<p>Объединения и перечисления.</p> <p>Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о недвижимости:</p> <ul style="list-style-type: none"> <li>· комната (площадь, количество соседей, свободна (да/нет));</li> <li>· квартира (район (4 варианта), количество комнат, площадь, этаж);</li> <li>· дом (район (4 варианта), площадь застройки, жилая площадь, процент готовности).</li> </ul> <p>Создать массив из 5 таких структур.</p> <p>Написать программу, выполняющую следующие действия:</p> <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	<p>Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:</p> <p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ сигнал светофора (красный, желтый, зеленый);</li> <li>◆ количество полос (до 6);</li> <li>◆ скорость автомобиля.</li> </ul>
18	1	<p>Мастерская по ремонту электроники. Наименование группы изделий (телевизоры, радиоприемники и т. п.), марка изделия, дата приемки в ремонт, состояние готовности заказа (выполнен, не выполнен). Дату приемки в ремонт реализовать с помощью битового поля, группу изделий – через перечисления.</p>
	2	<p>Объединения и перечисления.</p> <p>Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о транспорте:</p> <ul style="list-style-type: none"> <li>· автомобиль (марка (4 варианта), грузоподъемность, количество пассажиров);</li> <li>· поезд (тип поезда (курьерский, скорый, пассажирский), количество вагонов, скорость);</li> <li>· самолет (модель (4 варианта), к-во пассажиров, скорость).</li> </ul> <p>Создать массив из 5 таких структур.</p> <p>Написать программу, выполняющую следующие действия:</p> <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	<p>Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:</p> <p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ номер группы (6 цифр);</li> <li>◆ номер студента в группе (2 цифры);</li> </ul>

		◆ пол (м/ж).
19	1	Цех. Наименование, количество работников, количество изготавливаемых деталей за смену, номер цеха, количество конвейерных линий и количество аварийных выходов. Для заданного цеха необходимо вывести количество выпущенных изделий по каждому наименованию. Номер цеха, количество конвейерных линий и количество аварийных выходов реализовать с помощью битового поля, наименование изделий – через перечисления.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о комплектующих: <ul style="list-style-type: none"> <li>· монитор (диагональ, ЖК/ЭЛТ, разрешение (4 варианта));</li> <li>· клавиатура (количество клавиш, русификация (да/нет));</li> <li>· мышь (производитель, количество кнопок, наличие провода (да/нет), оптическая/механическая).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму): <ul style="list-style-type: none"> <li>◆ марка мотоцикла (10 вариантов);</li> <li>◆ год выпуска (с 1990 г. по 2009 г.);</li> <li>◆ объем двигателя.</li> </ul>
20	1	Спортивные соревнования. Наименование страны, вид спорта, название команды, ФИО игрока, игровой номер, возраст. Вывести информацию о самой молодой команде. Игровой номер и возраст реализовать с помощью битового поля, вид спорта – через перечисления.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о транспорте: <ul style="list-style-type: none"> <li>· автомобиль (марка (4 варианта), грузоподъемность, количество пассажиров);</li> <li>· поезд (тип поезда (курьерский, скорый, пассажирский), количество вагонов, скорость);</li> <li>· самолет (модель (4 варианта), к-во пассажиров, скорость).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму): <ul style="list-style-type: none"> <li>◆ номер группы (6 цифр);</li> <li>◆ номер студента в группе (2 цифры);</li> <li>◆ пол (м/ж).</li> </ul>
21	1	Знак зодиака. Фамилия, имя, знак зодиака, дата рождения. Дату рождения реализовать с помощью битового поля, знак зодиака – через перечисление.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о недвижимости: <ul style="list-style-type: none"> <li>· комната (площадь, к-во соседей, свободна(да/нет));</li> <li>· квартира (район (4 варианта), количество комнат, площадь, этаж);</li> <li>· дом (район (4 варианта), площадь застройки, жилая площадь, процент готовности).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму): <ul style="list-style-type: none"> <li>◆ сигнал светофора (красный, желтый, зеленый);</li> <li>◆ количество полос (до 6);</li> <li>◆ допустимая скорость автомобиля.</li> </ul>

22	1	Дисплеи. Наименование модели, фирма-производитель, цена, размер по диагонали (в дюймах), размер по горизонтали и вертикали (в пикселях), оптическое разрешение ( $ppi = (\sqrt{\text{горизонталь}^2 + \text{вертикаль}^2}) / \text{диагональ}$ ) (точек на дюйм). Диагональ, ширину и высоту дисплея реализовать с помощью битовых полей; фирму-производителя – через перечисления.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о комплектующих: <ul style="list-style-type: none"> <li>· монитор (диагональ, ЖК/ЭЛТ, разрешение (4 варианта));</li> <li>· клавиатура (количество клавиш, русификация (да/нет));</li> <li>· мышь (производитель, количество кнопок, наличие провода (да/нет), оптическая/механическая).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму): <ul style="list-style-type: none"> <li>◆ марка мотоцикла (10 вариантов);</li> <li>◆ год выпуска (с 1990 г. по 2009 г.);</li> <li>◆ объем двигателя.</li> </ul>
23	1	Донор крови. Фамилия, имя, группа крови, дата рождения, адрес. Дату рождения реализовать с помощью битового поля, группу крови – через перечисления.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о предметах искусства: <ul style="list-style-type: none"> <li>· картина (размеры (длина, ширина), материал (холст, дерево, бумага));</li> <li>· скульптура (размеры, материал (мрамор, бронза, гипс));</li> <li>· ювелирное изделие (вес, материал (придумайте сами), стоимость).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места - вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму): <ul style="list-style-type: none"> <li>◆ марка автомобиля (10 вариантов);</li> <li>◆ объем двигателя;</li> <li>◆ пробег.</li> </ul>
24	1	Военнослужащий. Фамилия, имя, адрес, дата рождения, должность, звание. Дату рождения реализовать с помощью битового поля, звание – через перечисления.
	2	Объединения и перечисления. Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о фигурах: <ul style="list-style-type: none"> <li>· треугольник (три стороны);</li> <li>· ромб (две диагонали);</li> <li>· окружность (радиус).</li> </ul> Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия: <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места - вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей: 1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму): <ul style="list-style-type: none"> <li>◆ пол (м/ж);</li> <li>◆ возраст;</li> <li>◆ рост.</li> </ul>
25	1	Кинолента. Название, режиссер (фамилия, имя), год выхода, страна, стоимость производства, прибыль от кинопроката. Год выхода реализовать с помощью битового поля, страну – через перечисления.
	2	Объединения и перечисления.

		<p>Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о товарах:</p> <ul style="list-style-type: none"> <li>· в наличии (номенклатурный №, наименование, единицы измерения (штуки, кг, упаковки), цена);</li> <li>· на складе (№ склада, № стеллажа, тип контейнера (белый, синий, красный));</li> <li>· под заказ (№ заказа, фамилия, дата).</li> </ul> <p>Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия:</p> <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места - вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	<p>Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:</p> <p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ цвет (16 цветов);</li> <li>◆ длина (до 17 цифр);</li> <li>◆ тип (3 типа).</li> </ul>
26	1	<p>Велосипед. Название производителя, наименование модели, год выпуска, ростовка рамы (размер, число в дюймах), страна-производитель, число передач (от 1 до 3-х), число скоростей (от 1 до 36-и), наличие или отсутствие багажника, вес, материал рамы (обычная сталь, хром-молибденовая сталь, марганцевая сталь, алюминий, титан, карбон, дерево). Материал рамы реализовать с помощью перечисления. Число передач и скоростей реализовать с помощью битовых полей.</p>
	2	<p>Объединения и перечисления.</p> <p>Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о клиентах:</p> <ul style="list-style-type: none"> <li>· студенты (№ зачетки, фамилия, пол(м/ж), средний балл);</li> <li>· преподаватели (табельный номер, фамилия, должность (4 должности));</li> <li>· слушатели (№ паспорта, фамилия, специальность (4 специальности)).</li> </ul> <p>Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия:</p> <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет - вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	<p>Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:</p> <p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ номер поезда (4 десятичных цифры);</li> <li>◆ время отправления;</li> <li>◆ тип поезда (4 типа).</li> </ul>
27	1	<p>Компьютерная игра. Название, версия (номер), жанр (квест, головоломка, шутер («стрелялка»), стратегия и т.д.), дата выхода на рынок, дата изъятия из продажи, однопользовательская или многопользовательская, локальная или сетевая, средний рейтинг у критиков. Жанр игры реализовать с помощью перечисления. Дату реализовать как битовое поле.</p>
	2	<p>Объединения и перечисления.</p> <p>Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о платежах:</p> <ul style="list-style-type: none"> <li>· наличными (вид валюты, сумма);</li> <li>· по карточке (тип карточки (4 типа), № карточки, фамилия, сумма);</li> <li>· SMS-banking (оператор, банк, № карточки, № телефона, сумма).</li> </ul> <p>Создать массив из 5 таких структур. Написать программу, выполняющую следующие действия:</p> <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места – вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет - вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
	3	<p>Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:</p> <p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ год (с 1900 по 2100);</li> <li>◆ месяц;</li> <li>◆ день месяца.</li> </ul>
28	1	<p>Интернет-сайт. Адрес (например, www.name.org), владелец, фирма-разработчик, языки и технологии кодировки (html, css, Python, C++, Java, JavaScript, C#, Blazor, JSON), дата опубликования, домен (org, gov, com, biz, us, su, ru, by, uk, ge, fr, me, tv). Домен и языки разработки реализовать с помощью перечислений; дату</p>

	опубликования реализовать посредством битового поля.
2	<p>Объединения и перечисления.</p> <p>Описать структуру для хранения следующей информации (при необходимости используйте перечисления): данные о животных (живых существах):</p> <ul style="list-style-type: none"> <li>· птицы (вид, вес, водоплавающие (да/нет), дальность полета);</li> <li>· рыбы (вид, цвет (4 цвета), ареал обитания (4 ареала));</li> <li>· рептилии (вид, страна происхождения, длина).</li> </ul> <p>Создать массив из 5 таких структур.</p> <p>Написать программу, выполняющую следующие действия:</p> <ul style="list-style-type: none"> <li>· добавление записи в массив; если в массиве нет места - вывести соответствующее сообщение;</li> <li>· удаление записи из массива; если такой записи нет – вывести соответствующее сообщение;</li> <li>· вывод всех записей массива.</li> </ul>
3	<p>Продекларируйте битовые поля и напишите код, работающий с экземплярами ваших битовых полей:</p> <p>1. Опишите битовое поле для хранения следующей информации (размер задавать по минимуму):</p> <ul style="list-style-type: none"> <li>◆ масть карты;</li> <li>◆ достоинство карты;</li> <li>◆ цвет карты.</li> </ul>