

Записи

Задачи с использованием типа
record

Запись — это структурированный тип данных, состоящий из фиксированного числа логически связанных компонентов одного или нескольких типов.

Компоненты записи называются полями, каждое из которых определяется именем. Поле записи содержит имя поля, вслед за которым через двоеточие указывается тип этого поля. Поля записи могут относиться к любому типу, допустимому в языке Паскаль, за исключением файлового типа.

Определение типа записи начнется идентификатором **record** и заканчивается зарезервированным словом **end**. Между ними заключен список компонентов (полей) с указанием идентификаторов полей и типа каждого поля

Пример:

type

Car = record

Number: integer; {Номер}

Marka: string [20]; {Марка автомобиля}

FIO: string [40]; {Фамилия, инициалы
владельца}

Address: string [60] {Адрес владельца}

End;

Var

M, V : Car;

Описание записей возможно и без использования имени типа, например:

```
var rec: record  
    fio: string[20] ;  
    tel: string[7]  
end;
```

Обращение к записи в целом допускается только в операторах присваивания, где слева и справа от знака присваивания используются имена записей одинакового типа.

V := M

Во всех остальных случаях оперируют отдельными полями записей. Чтобы обратиться к отдельной компоненте записи, необходимо задать имя записи и через точку указать имя нужного поля, например в операторах ввода-вывода:

```
Read (M.Number, M.Marka, M.FIO, M.Address);  
Write (M.Number:4, M.Marka:7, M.FIO:12, M.Address:25);
```

Пример массива из записей:

```
type  
    Person = record  
        FIO : string;  
        Age : 1..99;  
        Prof : string [30]  
    end;  
Var  
List : array [1..50] of Person;
```

Для обращения к полям записи можно использовать оператор присоединения **with**

Формат:

```
With <переменная типа запись> do <оператор>
```

Пример:

```
with M do begin  
    Number:= 588;  
    Marka:= 'Renault' ;  
    FIO:= 'Сидоренко А.П.' ;  
    Address:=' ул Лихачева 12-45'  
end
```

Простые задачи

На вход программе подаются сведения об N учащихся в формате: Фамилия, имя, год рождения. Вывести все фамилии учащихся заданного года рождения (год задается с клавиатуры), если таких нет вывести 'no'.

Количество учащихся не более 50.

Вход	Выход
4 2001 Иванов Иван 2002 Петрова Анна 2001 Сергеева Нина 2002 Миронов Антон 2000	Петрова

На вход программе подаются сведения об N учащихся в формате: Фамилия, имя, год рождения. Вывести фамилии все фамилии учащихся наиболее позднего года рождения.

Количество учащихся не более 50.

Вход	Выход
4 Иванов Иван 2002 Петрова Анна 2001 Сергеева Нина 2002 Миронов Антон 2000	Иванов Миронов

На вход программе подаются сведения об N автомобилях и их стоимости в формате: марка, год выпуска, стоимость. Обеспечить ввод данных в программу и найти автомобиль, стоимость которого максимально близка к средней стоимости, вычисленной по всем автомобилям.

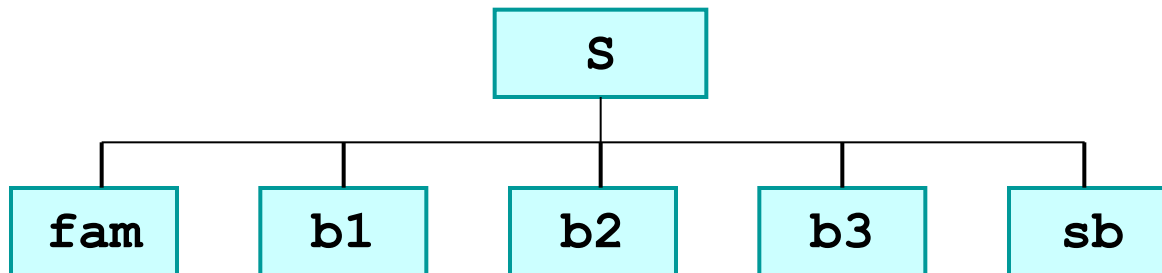
Вход	Выход
3 Калина 2012 250000 Honda SM-X 1999 200000 Opel Astra 2000 350000	Калина 2012

Пример 1

В классе n учеников. Для каждого ученика указаны фамилия и оценки по трем выпускным экзаменам. Требуется вычислить средний балл каждого ученика.

Очевидно, что фамилия ученика — это данные символьного типа, оценка на экзамене — отрезок целочисленного типа 2..5, а средний балл — это вещественное число (тип **real**).

Введем понятные идентификаторы: **fam**, **b1**, **b2**, **b3** и **sb**. Переменная **s** — это запись, содержащая информацию по одному ученику. Организацию такой записи можно продемонстрировать графически.



```
program zapis1;  
const n=10;  
type  
s = record  
    fam: string;  
    b1, b2, b3: 2..5;  
    sb: real;  
end;  
var tbl: array [1..n] of s;  
    i: integer;  
begin  
    for i:= 1 to n do  
        readln  
        (tbl[i].fam,tbl[i].b1,tbl[i].b2,tbl[i].b3) ;  
    for i:= 1 to n do  
begin {подсчитываются средние баллы}  
tbl[i].sb:= (tbl[i].b1 + tbl[i].b2 + tbl[i].b3)/3;  
    writeln(tbl[i].sb) end;  
end.
```

Простая программа с использованием записей:
пусть нам необходимо заполнить сведения о студенте (Ф.
И.О., дата рождения, адрес, курс и группа), а затем
вывести эти сведения на экран.

```
program primer1;  
type anketa=record  
    fio: string[45];  
    dat_r: string[8];  
    adres: string[50];  
    curs: 1..5;  
    grupp: string[3]  
end;  
var student: anketa;
```

```
begin
    writeln ('введите сведения о студенте');
    {обратите внимание, ввод каждого поля осуществляется
отдельно}
    writeln ('введите фамилию, имя и отчество');
    readln (student.fio);
    writeln ('введите дату рождения');
    readln (student.dat_r);
    writeln ('введите адрес');
    readln(student.adres);
    writeln ('введите курс');
    readln(student.curs);
    writeln ('введите группу');
    readln (student.grupp);
    writeln ('ввод закончен');
    writeln ;
    {обратите внимание, что вывод записи осуществляется по
полям}
    writeln ('фамилия студента: ', student . fio );
    writeln(' дата рождения : ', student.dat_r);
    writeln(' адрес : ', student.adres);
    writeln(' курс : ', student.curs);
    writeln(' группа : ', student.grupp);
end.
```

C4

На вход программе сначала подаётся значение N – количество учеников школы, следующие N строк содержат список дней рождения учеников этой школы. Каждый день рождения занимает отдельную строку, содержащую дату, записанную по правилам русского языка строчными буквами, например «двадцать третье февраля». Год не указывается, слова отделяются друг от друга одним пробелом, в начале и в конце строки дополнительных пробелов нет.

Напишите эффективную программу, которая определяет количество дней рождения в каждом календарном месяце и выводит названия месяцев и соответствующие количества в порядке убывания количества дней рождения в месяце. Если у двух и более месяцев количество дней рождения совпадает, эти месяцы следует выводить в том порядке, в каком они следуют в календаре. Если в каком-то месяце дней рождения нет, данные об этом месяце выводить не надо.

Размер памяти, которую использует Ваша программа, не должен зависеть от длины исходного списка.

Перед текстом программы кратко опишите используемый Вами алгоритм решения задачи.

Пример входных данных:

```
5
первое мая
восьмое марта
девятое мая
двадцать третье февраля
девятое мая
```

Пример выходных данных для приведённого выше примера входных данных:

```
май 3
февраль 1
март 1
```

```
type data = record
  m:string;
  c:integer;
end;
var s: string;
    n,i,j:integer;
    a:array[1..12]of data;
    t:data;
```

```
begin
readln(n) ;
for i:=1 to 12 do
    begin
a[ i].c:=0;
    case i of
1: a[ i].m:='январь' ;
2: a[ i].m:='февраль' ;
3: a[ i].m:='март' ;
4: a[ i].m:='апрель' ;
5: a[ i].m:='май' ;
6: a[ i].m:='июнь' ;
7: a[ i].m:='июль' ;
8: a[ i].m:='август' ;
9: a[ i].m:='сентябрь' ;
10: a[ i].m:='октябрь' ;
11: a[ i].m:='ноябрь' ;
12: a[ i].m:='декабрь' ;
    end;
    end;
```



```

for i:=1 to n do begin
readln(s);
while pos(' ',s)<>0 do delete(s,1,pos(' ',s)); {в
строке оставляем только название месяца}
s:=copy(s,1,3); {по трем буквам месяц определяется
однозначно}
if s = 'янв' then a[1].c:=a[1].c+1;
if s = 'фев' then a[2].c:=a[2].c+1;
if s = 'мар' then a[3].c:=a[3].c+1;
if s = 'апр' then a[4].c:=a[4].c+1;
if s = 'мая' then a[5].c:=a[5].c+1;
    {"май" из трех букв, поэтому
учитываем окончание}
if s = 'июн' then a[6].c:=a[6].c+1;
if s = 'июл' then a[7].c:=a[7].c+1;
if s = 'авг' then a[8].c:=a[8].c+1;
if s = 'сен' then a[9].c:=a[9].c+1;
if s = 'окт' then a[10].c:=a[10].c+1;
if s = 'ноя' then a[11].c:=a[11].c+1;
if s = 'дек' then a[12].c:=a[12].c+1;
end;

```

Вместо серии if-ов:

```

if s = 'мая' then
a[5].c:=a[5].c+1;
for j:= 1 to 12 do
if pos(s,a[j].m)<>0
then
a[j].c:=a[j].c+1;
end;

```

```
for i := 1 to 11 do
  for j := i+1 to 12 do
    if a[j-1].c < a[j].c then
      begin
        t:=a[j-1];a[j-1]:=a[j];a[j]:=t
      end;
  for i:= 1 to 12 do
    if a[i].c <> 0
      then writeln(a[ i].m+' ',a[ i].c)
  end.
```

На вход программы подаются фамилии и имена учеников. Известно, что общее количество учеников не превосходит 100. В первой строке вводится количество учеников, принимавших участие в соревнованиях, N. Далее следуют N строк, имеющих следующий формат:

<Фамилия> <Имя>

Здесь *<Фамилия>* – строка, состоящая не более чем из 20 символов; *<Имя>* – строка, состоящая не более чем из 15 символов. При этом *<Фамилия>* и *<Имя>* разделены одним пробелом. Примеры входных строк:

Иванова Мария

Петров Сергей

Требуется написать программу, которая формирует и печатает уникальный логин для каждого ученика по следующему правилу: если фамилия встречается первый раз, то логин – это данная фамилия, если фамилия встречается второй раз, то логин – это фамилия, в конец которой приписывается число 2 и т.д. Например, для входной последовательности

Иванова Мария

Петров Сергей

Бойцова Екатерина

Петров Иван

Иванова Наташа

будут сформированы следующие логины:

Иванова

Петров

Бойцова

Петров2

Иванова2

Пример С4 из ЕГЭ

В этой задаче мы должны хранить в памяти все **разные** фамилии и считать, сколько раз они встретились. При этом имена нас не интересуют, поэтому можно выделить такой массив записей

```
var Info: array[1..100] of record  
    name: string;      { фамилия }  
    count: integer;    { счетчик }  
end;
```

В памяти нужно хранить не все фамилии подряд, а только разные, поэтому нужен еще счетчик разных фамилий (**nFam**), в который в начале (важно!) надо записать ноль.

Здесь легко читать информацию целыми строками, а затем «вырезать» фамилию с помощью стандартных функций (фамилия окажется в строке **s**):

```
readln(s) ;  
p := Pos(' ', s) ;  
s := Copy(s, 1, p-1) ;
```

Теперь проверяем, если ли уже такая фамилия в списке. Нужно в цикле просмотреть **nFam** первых элементов массива **Info**, в случае, если фамилия найдена, флаг **exist** устанавливается в **True**, а номер найденной фамилии в массиве сохраняется в переменной **k**:

```
k := 1; exist := False;  
    while (not exist) and (k <= nFam) do  
begin  
    if s = Info[k].name then  
        exist := True  
    else k := k + 1;  
end;
```

Если фамилия найдена, мы увеличиваем счетчик в **k**-м элементе массива и выводим фамилию и (новое) значение счетчика; иначе увеличиваем счетчик разных фамилий **nFam** и добавляем в очередную запись прочитанную фамилию, в счетчик записываем

Полный код программы:

```
var Info: array[1..100] of record
    name: string;
    count: integer;
end;
i, k, p, N, nFam: integer;
s: string;
exist: boolean;
begin
    readln(N);
    nFam := 0;
    for i:=1 to N do begin
        readln(s);
        p := Pos(' ', s);
        s := Copy(s, 1, p-1);
        k := 1; exist := False;
```

```
while (not exist) and (k <= nFam) do begin
    if s = Info[k].name then
        exist := True
    else k := k + 1;
end;
if exist then begin
    Info[k].count := Info[k].count + 1;
    writeln(s, Info[k].count);
end
else begin
    nFam := nFam + 1;
    Info[nFam].name := s;
    Info[nFam].count := 1;
    writeln(s);
end
end;
end.
```