Символьные строки

Можно ли рассматривать строку как массив?

Чем плох массив символов?

Это массив символов:

```
var B: array[1..N] of char;
```

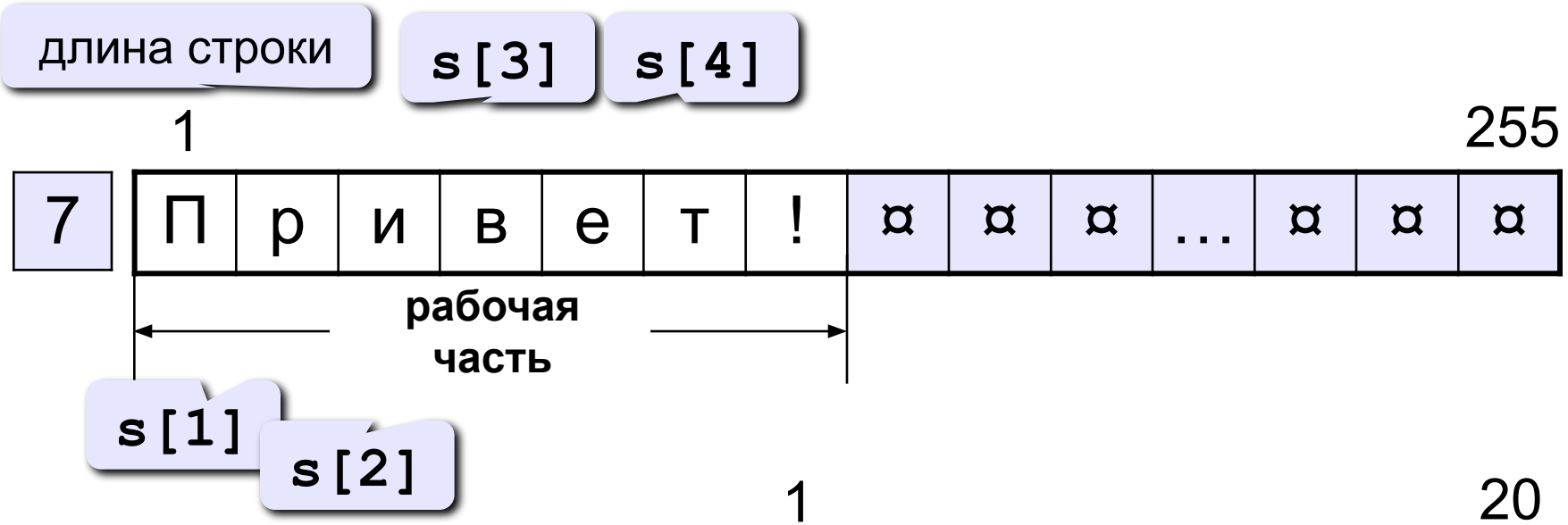
- каждый символ – отдельный объект;
- массив имеет длину N, которая задана при объявлении

Что нужно:

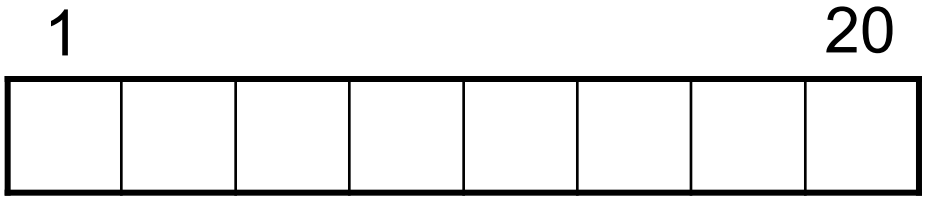
- обрабатывать последовательность символов как единое целое
- строка должна иметь переменную длину

Символьные строки

```
var s: string;
```




```
var s: string[20];
```



Длина строки:

```
var n: integer;
```

```
n := length ( s );
```

Символьные строки

Задача: ввести строку с клавиатуры и заменить все буквы «а» на буквы «б».

```
program qq;
var s: string;
    i: integer;
begin
   writeln('Введите строку');
   readln(s);
   for i:=1 to Length(s) do
      if s[i] = 'а' then s[i] := 'б';
   writeln(s);
end.
```

ввод строки

длина строки

вывод строки

Задания Эти же задания в файле с заданиями.

Выполнить 3 задания, затем "Операции со строками"

Ввести символьную строку и заменить все буквы «а» на буквы «б», как заглавные, так и строчные.

Пример:

```
Введите строку:
ааббссААББСС
Результат:
ббббссББББСС
```

Ввести символьную строку и заменить все буквы «а» на буквы «б» и наоборот, как заглавные, так и строчные.

Пример:

```
Введите строку:
ааббссААББСС
Результат:
ббаассББААСС
```

Задания

Ввести символьную строку и проверить, является ли она палиндромом (палиндром читается одинаково в обоих направлениях).

Пример:

```
Введите строку:
АБВГДЕ
Результат:
Не палиндром.
```

Пример:

```
Введите строку:
КАЗАК
Результат:
Палиндром.
```

Операции со строками

```
var s, s1, s2: string;
```

Запись нового значения:

```
s := 'Вася';
```

Объединение строк: добавить одну строку в конец

```
s1 := 'Привет';
s2 := 'Вася';
s := s1 + ', ' + s2 + '!';
```

`'Привет, Вася!'`

Выделение подстроки: выделить часть строки в другую строку.

```
s := '123456789';

s1 := Copy ( s, 3, 6 );
s2 := Copy ( s1, 2, 3 );
```

с 3-его символа

6 штук

`'345678'`

`'456'`

Удаление и вставка

Удаление части строки:

```
s := '123456789';
Delete ( s, 3, 6 );
```

6 штук

'123456789'

'129'

строка меняется!

с 3-его символа

Вставка в строку:

```
s := '123456789';
Insert ( 'ABC', s, 3 );

Insert ( 'Q', s, 5 );
```

начиная с 3-его символа

'12ABC3456789'

что вставляем

куда вставляем

'12ABQC3456789'

Поиск в строке

Поиск в строке:

s[3]

var n: integer;

```
s := 'Здесь был Вася.';
n := Pos ( 'e', s );
if n > 0 then
     writeln('Буква e - это s[', n, ']')
else writeln('Не нашли');
n := Pos ( 'Вася', s );
s1 := Copy ( s, n, 4 );
```

3

n = 11

Особенности:

- функция возвращает номер символа, с которого начинается образец в строке
- если слова нет, возвращается 0
- поиск с начала (находится **первое** слово)

Примеры

```
s := 'Вася Петя Митя';
n := Pos ( 'Петя', s );
Delete ( s, n, 4 );
Insert ( 'Лена', s, n );
```

```
6
'Вася  Митя'
'Вася Лена Митя'
```

```
s := 'Вася Петя Митя';
n := length ( s );
s1 := Copy ( s, 1, 4 );
s2 := Copy ( s, 11, 4 );
s3 := Copy ( s, 6, 4 );
s := s3 + s1 + s2;
n := length ( s );
```

```
14
'Вася'
'Митя'
'Петя'
'ПетяВасяМитя'
12
```

Пример решения задачи

Задача: Ввести имя, отчество и фамилию. Преобразовать их к формату «фамилия-инициалы».

Пример:

```
Введите имя, фамилию и отчество:
Василий Алибабаевич Хрюндиков
Результат:
Хрюндиков В.А.
```

Алгоритм:

- найти первый пробел и выделить имя
- удалить имя с пробелом из основной строки
- найти первый пробел и выделить отчество
- удалить отчество с пробелом из основной строки
- «сцепить» фамилию, первые буквы имени и фамилии, точки, пробелы…

© К. Поляков, 2006-2011

Программа

```
program qq;
var s, name, otch: string;
    n: integer;
begin
   writeln('Введите имя, отчество и фамилию');
   readln(s);
   n := Pos(' ', s);
   name := Copy(s, 1, n-1); { вырезать имя }
   Delete(s, 1, n);
   n := Pos(' ', s);
   otch := Copy(s, 1, n-1); { вырезать отчество }
   Delete(s, 1, n);          { осталась фамилия }
   s := s + ' ' + name[1] + '.' + otch[1] + '.';
   writeln(s);
end.
```

Задания всем 4,5,6, затем См. файл с заданиями в конце по вариантам

Ввести в одну строку фамилию, имя и отчество, разделив их пробелом. Вывести инициалы и фамилию.

Пример:

```
Введите фамилию, имя и отчество:
Иванов Петр Семёнович
Результат:
П.С. Иванов
```

Ввести имя файла (возможно, без расширения) и изменить его расширение на «.exe».

Пример:

```
Введите имя файла:
qqq
Результат:
qqq.exe
```

```
Введите имя файла:
qqq.com
Результат:
qqq.exe
```

Задания

Ввести путь к файлу и «разобрать» его, выводя каждую вложенную папку с новой строки

Пример:

```
Введите путь к файлу:
C:\Мои документы\10-Б\Вася\qq.exe
Результат:
C:
Мои документы
10-Б
Вася
qq.exe
```

Задачи на обработку строк

Задача: с клавиатуры вводится символьная строка, представляющая собой сумму двух целых чисел, например:

```
12+35
```

Вычислить эту сумму:

```
12+35=47
```

Алгоритм:

1) найти знак «+»
2) выделить числа слева и справа в отдельные строки
3) перевести строки в числа
4) сложить
5) вывести результат

Преобразования «строка»-«число»

Из строки в число:

```
var N, r: integer;
    X: real;
    s: string;
```

```
s := '123';
Val ( s, N, r ); { N = 123 }
 { r = 0, если ошибки не было
   r – номер ошибочного символа}
s := '123.456';
Val ( s, X, r ); { X = 123.456 }
```

Из числа в строку:

```
N := 123;
Str ( N, s );        { '123' }
X := 123.456;
Str ( X, s );        { '1.234560E+002' }
Str ( X:10:3, s ); { '   123.456' }
```

Программа

```
program qq;
var s, s1, s2: string;
    r, n, n1, n2, sum: integer;
begin
  writeln('Введите выражение (сумму чисел):');
  readln(s);
  n:=Pos('+', s);
  s1:=Copy(s, 1, n-1);
  s2:=Copy(s, n+1, Length(s)-n);
  Val(s1, n1, r);
  Val(s2, n2, r);
  sum:=n1+n2;
  writeln(n1, '+', n2, '=', sum);
end.
```

слагаемые-строки

сумма

слагаемые-числа

слагаемые-строки

слагаемые-числа

Задания

«3»: Ввести арифметическое выражение: разность двух чисел. Вычислить эту разность.

Пример:

```
25-12
Ответ: 13
```

«4»: Ввести арифметическое выражение: сумму трёх чисел. Вычислить эту сумму.

Пример:

```
25+12+34
Ответ: 71
```

© К. Поляков, 2006-2011

Задания

«5»: **Ввести арифметическое выражение с тремя числами, в котором можно использовать сложение и вычитание. Вычислить это выражение.**

Пример:

```
25+12+34
```

Ответ: 71

Пример:

```
25+12-34
```

Ответ: 3

Пример:

```
25-12+34
```

Ответ: 47

Пример:

```
25-12-34
```

Ответ: -21

Задания

«6»: **Ввести арифметическое выражение с тремя числами, в котором можно использовать сложение, вычитание и умножение. Вычислить это выражение.**

Пример:

```
25+12*3
```

Ответ: 61

Пример:

```
25*2-34
```

Ответ: 16

Пример:

```
25-12+34
```

Ответ: 47

Пример:

```
25*2*3
```

Ответ: 150

Посимвольный ввод

Задача: с клавиатуры вводится число N, обозначающее количество футболистов команды «Шайба», а затем – N строк, в каждой из которых – информация об одном футболисте таком формате:

<Фамилия> <Имя> <год рождения> <голы>

Все данные разделяются одним пробелом. Нужно подсчитать, сколько футболистов, родившихся в период с 1988 по1990 год, не забили мячей вообще.

Алгоритм:

```
for i:=1 to N do begin
  { пропускаем фамилию и имя }
  { читаем год рождения Year и число голов Gol }
  if (1988 <= Year) and (Year <=1990) and
     (Gol = 0) then  { увеличиваем счетчик }
end;
```

Посимвольный ввод

Пропуск фамилии:

```
var c: char;
```

```
repeat
  read(c);
until c = ' '; { пока не встретим пробел }
```

Пропуск имени:

```
repeat read(c); until c = ' ';
```

Ввод года рождения:

```
var Year: integer;
```

```
read(Year); { из той же введенной строки }
```

Ввод числа голов и переход к следующей строке:

```
readln(Gol); { читать все до конца строки }
```

```
var Gol: integer;
```

© К. Поляков, 2006-2011

Программа

```
program qq;
var c: char;
    i, N, count, Year, Gol: integer;
begin
  writeln('Количество футболистов');
  readln(N);
  count := 0;
  for i:=1 to N do begin
    repeat read(c); until c = ' ';
    repeat read(c); until c = ' ';
    read(Year);
    readln(Gol);
    if (1988 <= Year) and (year <= 1990) and
       (Gol = 0) then count := count + 1;
  end;
  writeln(count);
end.
```

Посимвольный ввод

Если фамилия нужна:

```
var fam: string;
```

```
fam := '';   { пустая строка }
repeat
  read(c);              { прочитать символ }
  fam := fam + c;       { прицепить к фамилии }
until c = ' ';
```

Вместо read(Year):

```
var s: string;
```

```
s := '';   { пустая строка }
repeat
  read(c);            { прочитать символ }
  s := s + c;         { прицепить к году }
until c = ' ';
Val(s, Year, r); { строку – в число }
```

Посимвольный ввод

Если нужно хранить все фамилии:

массив символьных строк

```
const MAX = 100;
var fam: array[1..MAX] of string;
...
fam[i] := '';  { пустая строка }
repeat
  read(c);          { прочитать символ }
  fam[i] := fam[i] + c;
until c = ' ';
```

Задания

Информация о футболистах вводится так же, как и для приведенной задачи (сначала N, потом N строк с данными).

«3»: Вывести фамилии и имена всех футболистов, которые забили больше двух голов.

Пример:

```
Иванов Василий
Семёнов Кузьма
```

«4»: Вывести фамилию и имя футболиста, забившего наибольшее число голов, и количество забитых им голов.

Пример:

```
Иванов Василий 25
```

Задания

«5»: Вывести *в алфавитном порядке* фамилии и имена всех футболистов, которые забили хотя бы один гол. В списке не более 100 футболистов.

Пример:

```
Васильев Иван
Иванов Василий
Кутузов Михаил
Пупкин Василий
```

Программирование на языке Паскаль
Часть II

Тема 7. Рекурсивный перебор

Рекурсивный перебор

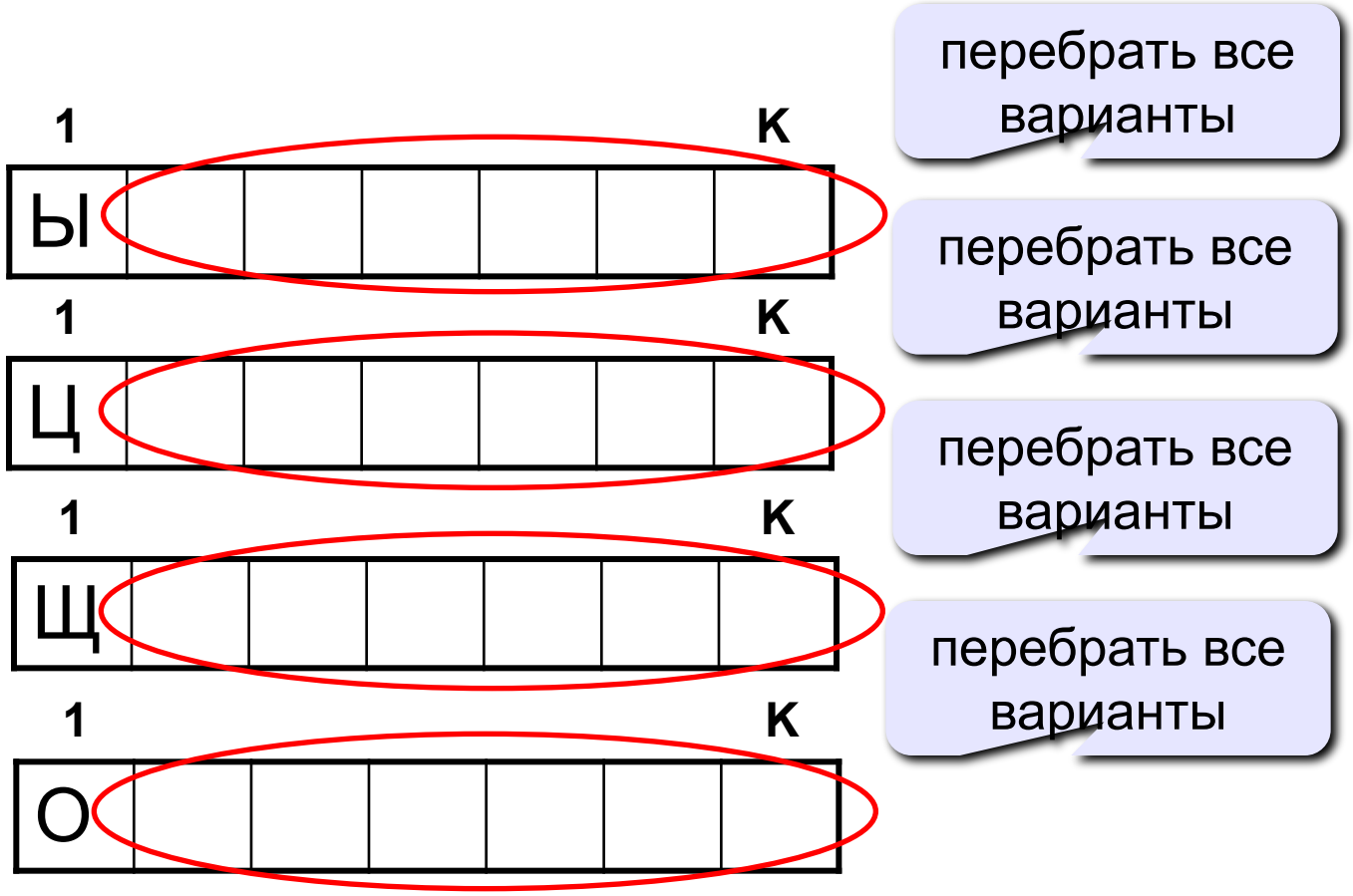
Задача: Алфавит языка племени «тумба-юмба» состоит из букв **Ы**, **Ц**, **Щ** и **О**. Вывести на экран все слова из **K** букв, которые можно составить в этом языке, и подсчитать их количество. Число **K** вводится с клавиатуры.

Количество вариантов:

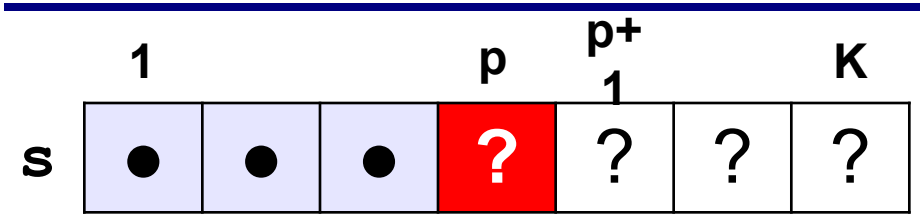
$$N = 4 \cdot 4 \cdot 4 \cdot \boxtimes \cdot 4 = 4^K$$

Рекурсивный перебор

Рекурсия: Решения задачи для слов из **K** букв сводится к 4-м задачам для слов из **K-1** букв.



Процедура



Глобальные переменные:

```
var s: string;
    count, K: integer;
```

```
procedure Rec(p: integer);
begin
  if p > K then begin
    writeln(s);
    count := count+1;
  end
  else begin
    s[p]:='Ы'; Rec ( p+1 );
    s[p]:='Ц'; Rec ( p+1 );
    s[p]:='Щ'; Rec ( p+1 );
    s[p]:='О'; Rec ( p+1 );
  end;
end;
```

окончание рекурсии

рекурсивные вызовы

? А если букв много?

Процедура

```
procedure Rec(p: integer);
const letters = 'ЫЩЩО';
var i: integer;
begin
  if p > k then begin
     writeln(s);
     count := count+1;
  end
  else begin
     for i:=1 to length(letters) do begin
       s[p] := letters[i];
       Rec(p+1);
     end;
  end;
end;
```

все буквы

локальная переменная

цикл по всем буквам

Программа

```
program qq;
var s: string;
    K, i, count: integer;
    procedure Rec(p: integer);
    ...
    end;
begin
  writeln('Введите длину слов:');
  read ( K );
  s := '';
  for i:=1 to K do s := s + ' ';
  Rec ( 1 );
  writeln('Всего ', count, ' слов');
end.
```

глобальные переменные

процедура

строка из K пробелов

Задания

Алфавит языка племени «тумба-юмба» состоит из букв **Ы**, **Ц**, **Щ** и **О**. Число **К** вводится с клавиатуры.

«3»: Вывести на экран все слова из **К** букв, в которых первая буква – **Ы**, и подсчитать их количество.

«4»: Вывести на экран все слова из **К** букв, в которых буква **Ы** встречается более 1 раза, и подсчитать их количество.

«5»: Вывести на экран все слова из **К** букв, в которых есть одинаковые буквы, стоящие рядом (например, **ЫЩЩО**), и подсчитать их количество.

© К. Поляков, 2006-2011

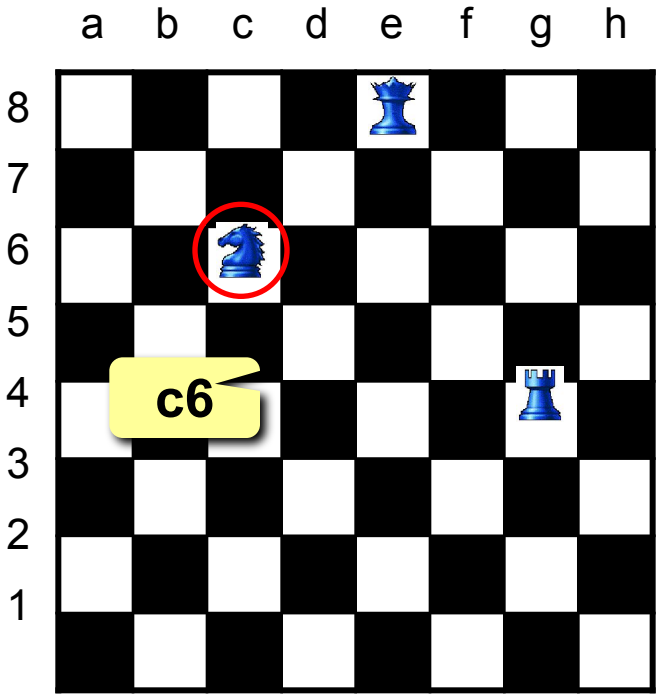
Программирование на языке Паскаль
Часть II

Тема 8. Матрицы

© К. Поляков, 2006-2011

Матрицы

Задача: запомнить положение фигур на шахматной доске.



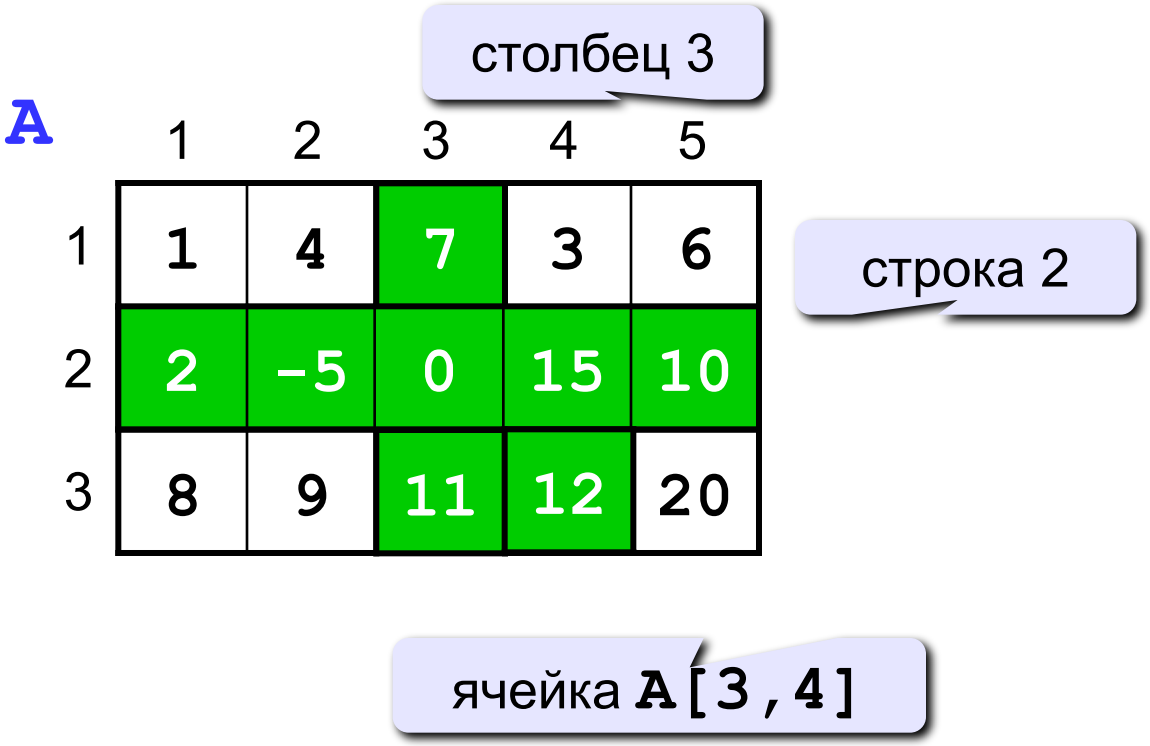
	1	2	3	4	5	6	7	8
8	0	0	0	0	2	0	0	0
7	0	0	0	0	0	0	0	0
6	0	0	3	0	0	0	0	0
5	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	4	0
3	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0

A[6,3]

Матрицы

Матрица – это прямоугольная таблица чисел (или других элементов одного типа).

Матрица – это массив, в котором каждый элемент имеет два индекса (номер строки и номер столбца).



Матрицы

Объявление:

```
const N = 3;
      M = 4;
var A: array[1..N,1..M] of integer;
    B: array[-3..0,-8..M] of integer;
    Q: array['a'..'d',False..True] of real;
```

Ввод с клавиатуры:

Если переставить циклы?

```
for j:=1 to M do
  for i:=1 to N do begin
    write('A[',i,',',j,']=');
    read ( A[i,j] );
  end;
```

i j

A[1,1] 2
A[1,2] 5
A[1,3] 1
=
A[3,4] 5
=

Матрицы

Заполнение случайными числами

```
for i:=1 to N do
  for j:=1 to M do
    A[i,j] := random(25) - 10;
```

цикл по строкам

цикл по столбцам

тервал?

Вывод на экран

```
for i:=1 to N do begin
  for j:=1 to M do
    write ( A[i,j]:5 );
  writeln;
end;
```

вывод строки

```
   12   25    1   13
  156    1   12  447
    1  456  222   23
```

в той же строке

перейти на новую строку

Если переставить циклы?

Обработка всех элементов матрицы

Задача: заполнить матрицу из 3 строк и 4 столбцов случайными числами и вывести ее на экран. Найти сумму элементов матрицы.

```
program qq;
const N = 3; M = 4;
var A: array[1..N,1..M] of integer;
    i, j, S: integer;
begin
  { заполнение матрицы и вывод на экран}
  S := 0;
  for i:=1 to N do
    for j:=1 to M do
      S := S + A[i,j];
  writeln('Сумма элементов матрицы ', S);
end.
```

Задания

Заполнить матрицу из 8 строк и 5 столбцов случайными числами в интервале [-10,10] и вывести ее на экран.

«3»: Удвоить все элементы матрицы и вывести её на экран.

«4»: Найти минимальный и максимальный элементы в матрице их номера. Формат вывода:

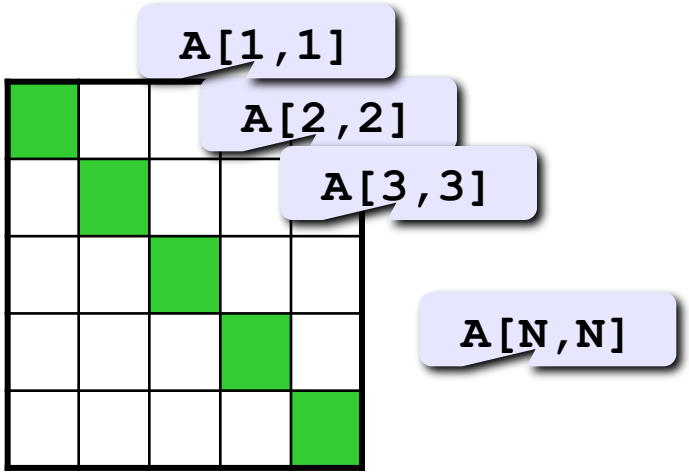
```
Минимальный элемент  A[3,4]=-6
Максимальный элемент A[2,2]=10
```

«5»: Вывести на экран строку, сумма элементов которой максимальна. Формат вывода:

```
Строка 2:  3  5  8  9  8
```

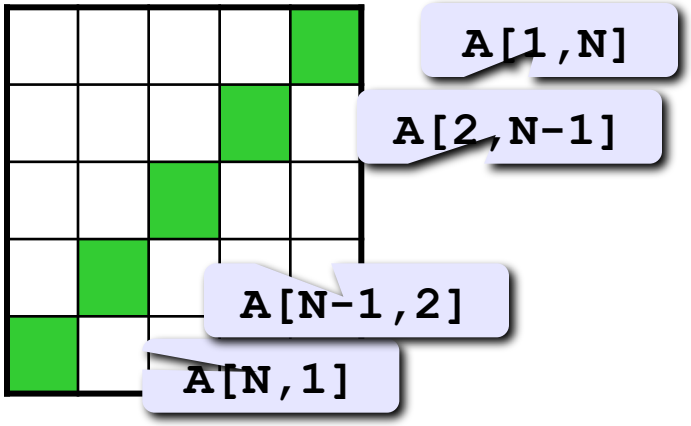
Операции с матрицами

Задача 1. Вывести на экран главную диагональ квадратной матрицы из N строк и N столбцов.




```
for i:=1 to N do
  write ( A[i,i]:5 );
```

Задача 2. Вывести на экран вторую диагональ.

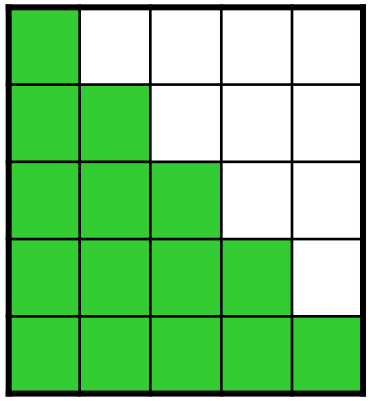


сумма номеров строки и столбца N+1

```
for i:=1 to N do
  write ( A[i, N+1-i ]:5 );
```

Операции с матрицами

Задача 3. Найти сумму элементов, стоящих на главной диагонали и ниже ее.



Одиночный цикл или вложенный?

строка 1: `A[1,1]`

строка 2: `A[2,1]+A[2,2]`

`...`

строка N: `A[N,1]+A[N,2]+...+A[N,N]`

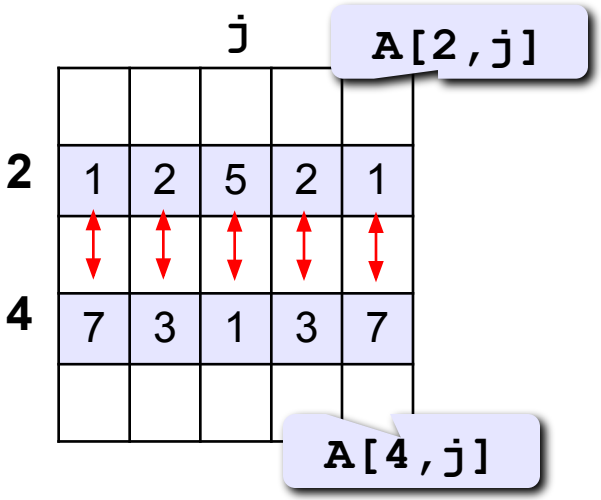
```
S := 0;
for i:=1 to N do
  for j:=1 to i do
    S := S + A[i,j];
```

цикл по всем строкам

складываем нужные элементы строки **i**

Операции с матрицами

Задача 4. Перестановка строк или столбцов. В матрице из N строк и M столбцов переставить 2-ую и 4-ую строки.




```
for j:=1 to M do begin
  c := A[2,j];
  A[2,j] := A[4,j];
  A[4,j] := c;
end;
```

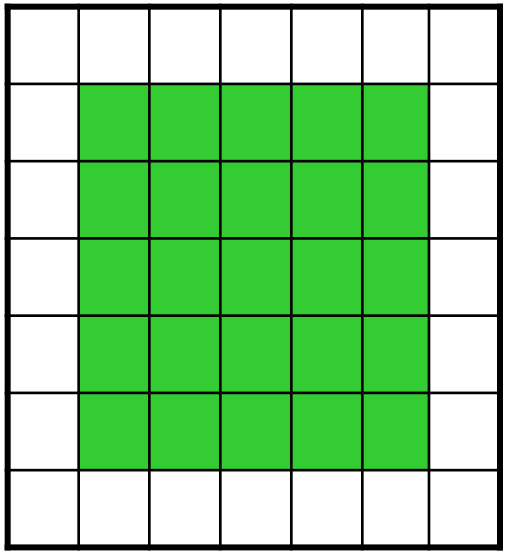
Задача 5. К третьему столбцу добавить шестой.

```
for i:=1 to N do
  A[i,3] := A[i,3] + A[i,6];
```

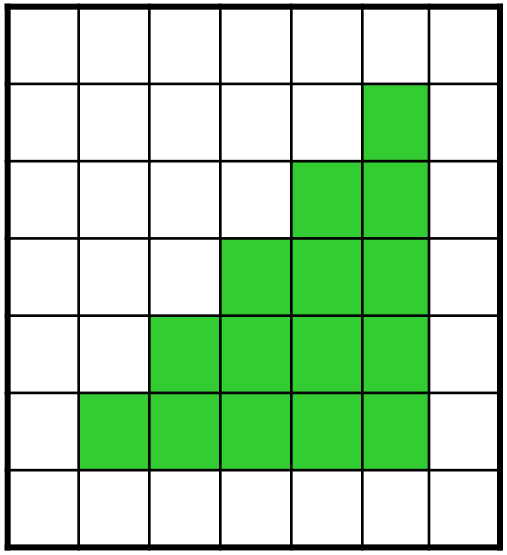
Задания

Заполнить матрицу из 7 строк и 7 столбцов случайными числами в интервале [10,90] и вывести ее на экран. Заполнить элементы, отмеченные зеленым фоном, числами 99, и вывести полученную матрицу на экран.

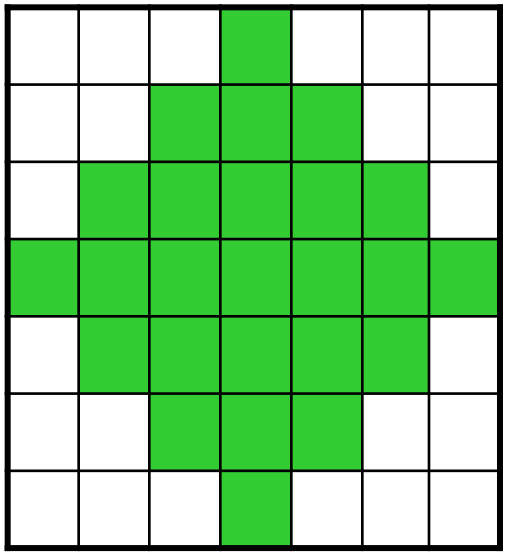
«3»:



«4»:



«5»:



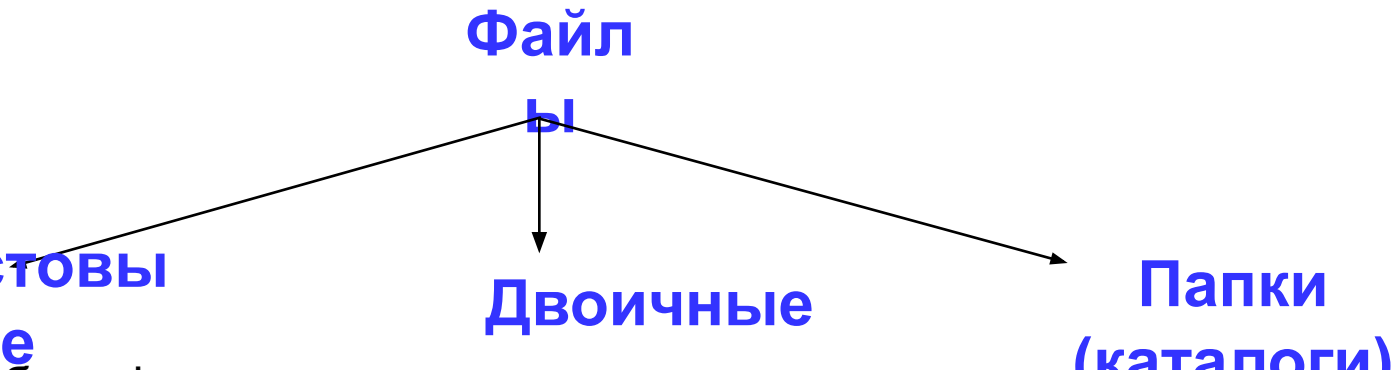
Программирование на языке Паскаль
Часть II

Тема 9. Файлы

© К. Поляков, 2006-2011

Файлы

Файл – это область на диске, имеющая имя.



только текст без оформления, не содержат управляющих символов (с кодами < 32)

ACSII (1 байт на символ)

UNICODE (2 байта на символ)

```
*.txt, *.log,
*.htm, *.html
```

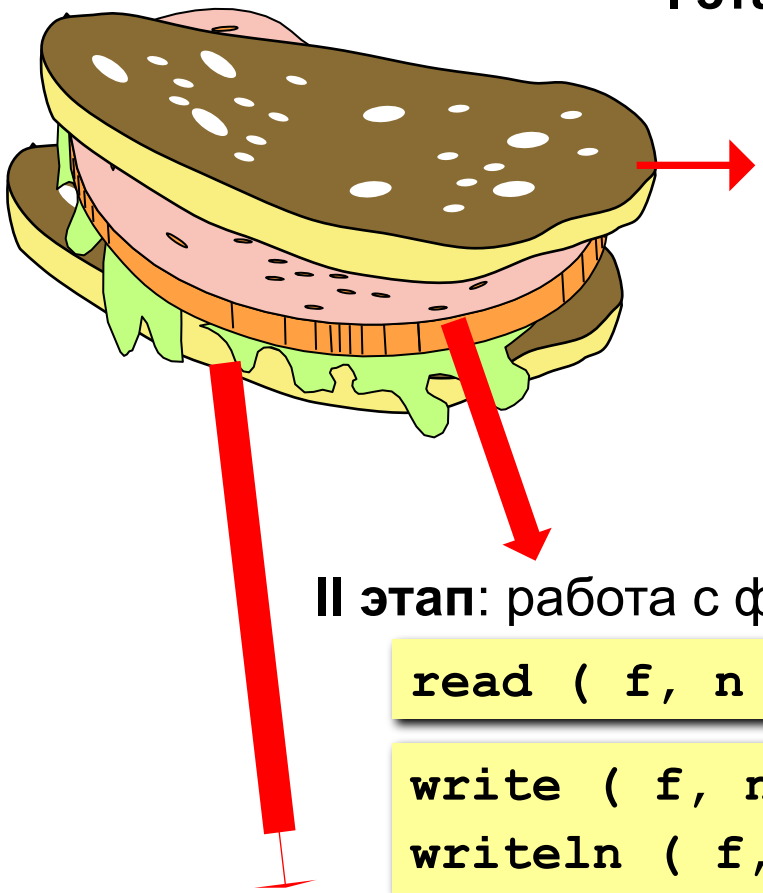
могут содержать любые символы кодовой таблицы

```
*.doc, *.exe,
*.bmp, *.jpg,
*.wav, *.mp3,
*.avi, *.mpg
```

Принцип сэндвича

Переменная типа «текстовый файл»:

```
var f: text;
```



I этап. открыть файл :

- связать переменную **f** с файлом

```
assign(f, 'qq.txt');
```

- открыть файл (сделать его активным, приготовить к работе)

```
reset(f); {для чтения}
```

```
rewrite(f); {для записи}
```

II этап: работа с файлом

```
read ( f, n );  { ввести значение n }
```

```
write ( f, n );  { записать значение n }
writeln ( f, n );{с переходом на нов.строку }
```

III этап: закрыть файл

```
close(f);
```

Работа с файлами

Особенности:

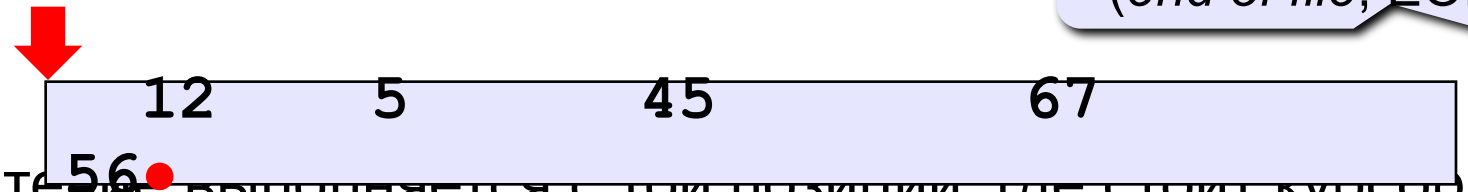
- имя файла упоминается только в команде **`assign`**, обращение к файлу идет через файловую переменную
- файл, который открывается на чтение, должен **существовать**
- если файл, который открывается на запись, существует, старое содержимое **уничтожается**
- данные записываются в файл в текстовом виде
- при завершении программы все файлы закрываются автоматически
- после закрытия файла переменную **`f`** можно использовать еще раз для работы с другим файлом

Последовательный доступ

- при открытии файла курсор устанавливается в начало

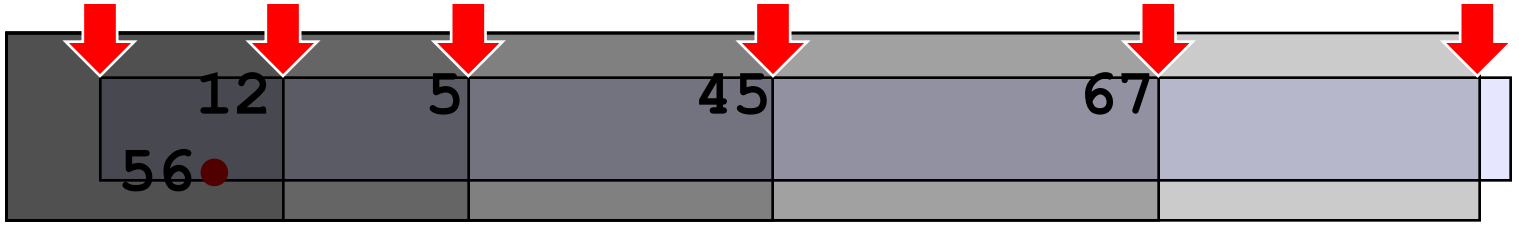
```
assign ( f, 'qq.txt' );
reset ( f );
```

конец файла (*end of file*, EOF)



- чтение выполняется с той позиции, где стоит курсор
- после чтения курсор сдвигается на первый непрочитанный символ

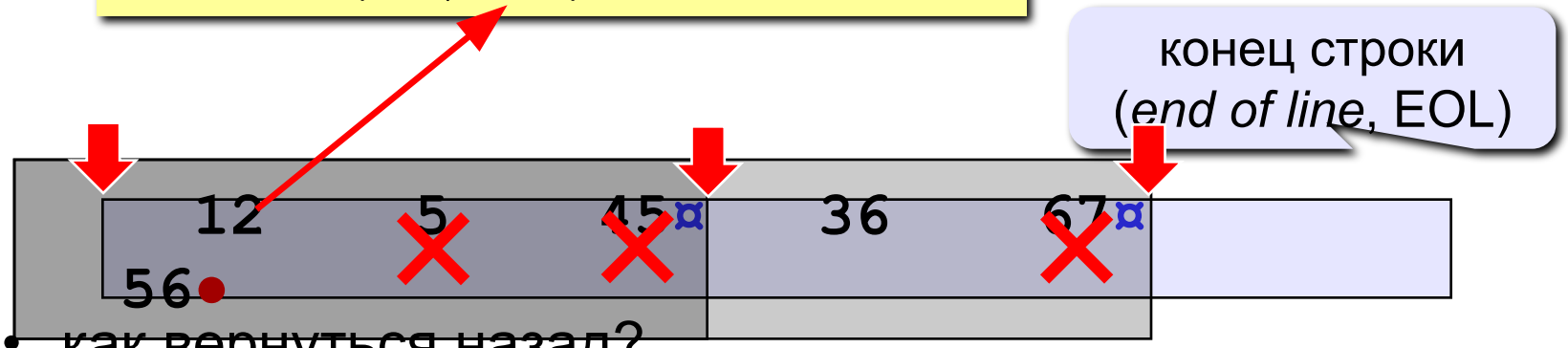
```
read ( f, x );
```



Последовательный доступ

- чтение до конца строки

```
readln ( f, x );
```



- как вернуться назад?

```
close ( f );
reset ( f ); { начать с начала }
```

Пример

Задача: **в файле `input.txt` записаны числа (в столбик), сколько их – неизвестно. Записать в файл `output.txt` их сумму.**

Можно ли обойтись без массива?

Алгоритм:

1. Открыть файл **`input.txt`** для чтения.
2. **`S := 0;`**
3. Если чисел не осталось, перейти к шагу 7.
4. Прочитать очередное число в переменную **`x`**.
5. **`S := S + x;`**
6. Перейти к шагу 3.
7. Закрыть файл **`input.txt`**.
8. Открыть файл **`output.txt`** для записи.
9. Записать в файл значение **`S`**.
10. Закрыть файл **`output.txt`**.

цикл с условием **«пока есть данные»**

Программа

```
program qq;
var s, x: integer;
    f: text;
begin
   assign(f, 'input.txt');
   reset(f);
   s := 0;
   while not eof(f) do begin
     readln(f, x);
     s := s + x;
   end;
   close(f);
   assign(f, 'output.txt');
   rewrite(f);
   writeln(f, 'Сумма чисел ', s);
   close(f);
end.
```

логическая функция, возвращает **True**, если достигнут конец файла

запись результата в файл **output.txt**

Задания

В файле `data.txt` записаны числа, сколько их – неизвестно.

«3»: Найти сумму чётных чисел и записать её в файл **`output.txt`**.

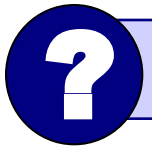
«4»: Найти минимальное и максимальное из четных чисел и записать их в файл **`output.txt`**.

«5»: Найти длину самой длинной цепочки одинаковых чисел, идущих подряд, и записать её в файл **`output.txt`**.

© К. Поляков, 2006-2011

Обработка массивов

Задача: в файле `input.txt` записаны числа (в столбик), сколько их – неизвестно, но не более 100. Переставить их в порядке возрастания и записать в файл `output.txt`.



Можно ли обойтись без массива?

Проблемы:

1. для сортировки надо удерживать в памяти все числа сразу (массив);
2. сколько чисел – неизвестно.

Решение:

3. выделяем в памяти массив из 100 элементов;
4. записываем прочитанные числа в массив и считаем их в переменной `N`;
5. сортируем первые `N` элементов массива;
6. записываем их в файл.

© К. Поляков, 2006-2011

Чтение данных в массив

Глобальные переменные:

```
var A: array[1..100] of integer;
    f: text;
```

Функция: ввод массива, возвращает число элементов

```
function ReadArray: integer;
var i: integer;
begin
  assign(f, 'input.txt');
  reset(f);
  i := 0;
  while (not eof(f)) and (i < 100) do begin
    i := i + 1;
    readln(f, A[i]);
  end;
  close(f);
  ReadArray :=
        i;
end;
```

цикл заканчивается, если достигнут конец файла или прочитали 100 чисел

Программа

```
program qq;
var A: array[1..100] of integer;
    f: text;
    N, i: integer;
function ReadArray: integer;
...
end;
Begi

   N := ReadArray;
   { сортировка первых N элементов }

   assign(f, 'output.txt');
   rewrite(f);
   for i:=1 to N do
     writeln(f, A[i]);
   close(f);
enc
```

вывод отсортированного массива в файл

Задания

В файле `input.txt` записаны числа (в столбик), известно, что их не более 100.

«3»: Отсортировать массив по убыванию и записать его в файл **`output.txt`**.

«4»: Отсортировать массив по убыванию последней цифры и записать его в файл **`output.txt`**.

«5»: Отсортировать массив по возрастанию суммы цифр и записать его в файл **`output.txt`**.

Обработка текстовых данных

Задача: в файле `input.txt` записаны строки, в которых есть слово-паразит «*короче*». Очистить текст от мусора и записать в файл `output.txt`.

Файл `input.txt` :

```
Мама, короче, мыла, короче, раму.
Декан, короче, пропил, короче, бутан.
А роза, короче, упала на лапу, короче, Азора.
Каждый, короче, охотник желает, короче, знать, где ...
```

Результат - файл `output.txt` :

```
Мама мыла раму.
Декан пропил бутан.
А роза упала на лапу Азора.
Каждый охотник желает знать, где сидит фазан.
```

Обработка текстовых данных

Алгоритм:

пока не кончились данные

1. Прочитать строку из файла (`readln`).
2. Удалить все сочетания ", *короче,*" (`Pos, Delete`).
3. Записать строку в другой файл.
4. Перейти к шагу 1.

Обработка строки `s`:

искать «, короче,»

```
repeat
  i := Pos(', короче,', s);
  if i <> 0 then Delete(s, i, 9);
until i = 0;
```

удалить 9 символов

Особенности:

надо одновременно держать открытыми два файла (один в режиме чтения, второй – в режиме записи).

Работа с двумя файлами одновременно

```
program qq;
var s: string;
    i: integer;
    fIn, fOut:
       text;
begin
  assign(fIn, 'input.txt');
  reset(fIn);
  assign(fOut, 'output.txt');
  rewrite(fOut);
  { обработать файл }
  close(fIn);
  close(fOut);
end.
```

файловые переменные

открыть файл для чтения

открыть файл для записи

Полный цикл обработки файла

пока не достигнут конец файла

```
while not eof(fIn) do begin
  readln(fIn, s);
  repeat
    i := Pos(', короче,', s);
    if i <> 0 then
      Delete(s, i, 9);
  until i = 0;
end;
```

обработка строки

запись «очищенной» строки

Задания

В файле `input.txt` записаны строки, сколько их – неизвестно.

«3»: Заменить все слова «короче» на «в общем» и записать результат в файл **`output.txt`**.

«4»: Вывести в файл **`output.txt`** только те строки, в которых есть слово «пароход». В этих строках заменить все слова «короче» на «в общем».

«5»: Вывести в файл **`output.txt`** только те строки, в которых больше 5 слов (слова могут быть разделены несколькими пробелами).

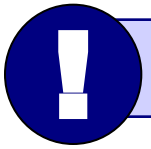
Сортировка списков

Задача: в файле `list.txt` записаны фамилии и имена пользователей сайта (не более 100). Вывести их в алфавитном порядке в файл `sort.txt`.

Файл `list.txt` :

```
Федоров Иван
Иванов Федор
Анисимов Никита
Никитин Николай
```

Нужен ли массив!



Для сортировки нужен массив!

Результат – файл `sort.txt` :

```
Анисимов Никита
Иванов Федор
Никитин Николай
Федоров Иван
```

Сортировка списков

Алгоритм:

1)прочитать строки из файла в массив строк, подсчитать их в переменной **N**

2)отсортировать первые **N** строк массива по алфавиту

3)вывести первые **N** строк массива в файл

Объявление массива (с запасом):

```
const MAX = 100;
var s: array[1..MAX] of string;
```

© К. Поляков, 2006-2011

Сортировка списков

Ввод массива строк из файла:

```
var f:Text;
    N: integer;
```

```
assign(f, 'list.txt');
reset(f);
N:=0;
while not eof(f) do begin
  N:=N+1;
  readln(f, s[N]);
end;
close(f);
```

Сортировка списков

Сортировка первых N элементов массива:

```
for i:=1 to N-1 do begin
  nMin:=i;
  for j:=i+1 to N do
    if s[j]<s[nMin] then nMin:=j;
  if i<>nMin then begin
    c:=s[i];
    s[i]:=s[nMin];
    s[nMin]:=c;
  end;
end;
```

```
var i, j, nMin:
    integer;
  c: string;
```

Какой метод?

Сортировка списков

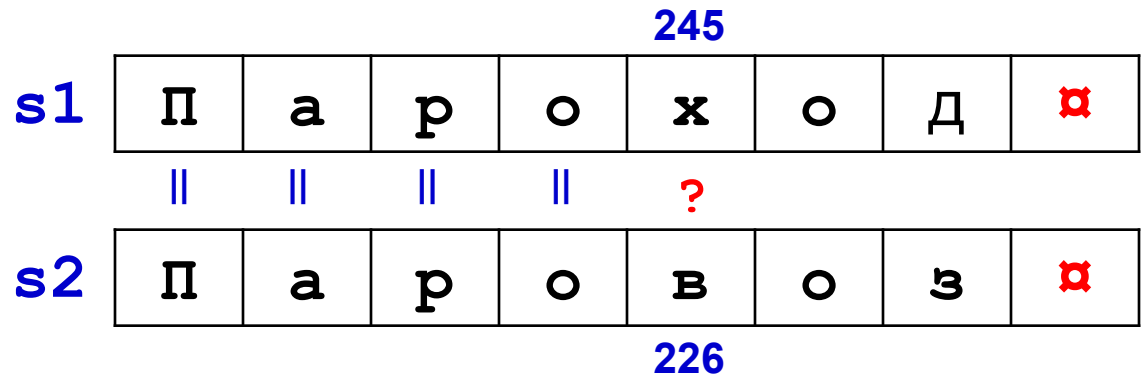
Вывод первых N строк массива в файл:

```
assign(f, 'sort.txt');
rewrite(f);
for i:=1 to N do
  writeln(f, s[i]);
close(f);
```

```
var f:Text;
    i, N: integer;
```

Сортировка списков

Как сравниваются строки:



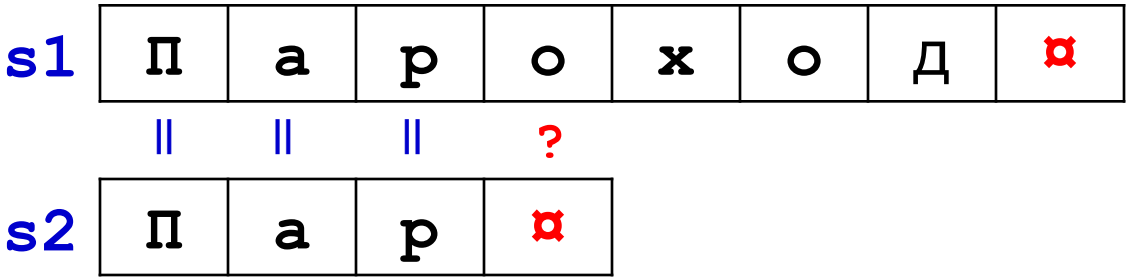
Кодовая таблица:

	А	Б	В	…	Я	а	б	в	…	х	…	я
Win	192	193	194	…	223	224	225	226	…	245	…	255
UNICODE	1040	1041	1042	…	1071	1072	1073	1074	…	1093	…	1103

```
код('х') > код('в')
'х' > 'в'
'Пароход' > 'Паровоз'
```

Сортировка списков

Как сравниваются строки:




```
'х' > ¤

'Пароход' > 'Пар'
```

Сортировка списков

Работа с отдельной строкой массива:

```
var s: array[1..MAX] of string;
    c: string; {вспомогательная строка}
...
for i:=1 to N do begin
  c:= s[i];
    { работаем со строкой c, меняем ее }
  s[i]:= c;
end;
```

Задания

«3»: Добавить к списку нумерацию:

```
1) Анисимов Никита
2) Иванов Федор
```

«4»: Выполнить задачу на «3» и сократить имя до первой буквы:

```
1) Анисимов Н.
2) Иванов Ф.
```

«5»: Выполнить задачу на «4», но при выводе начинать с имени:

```
1) Н. Анисимов
2) Ф. Иванов
```

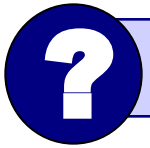
© К. Поляков, 2006-2011

Списки с числовыми данными

Задача: в файле `marks.txt` записаны фамилии и имена школьников и баллы, полученные ими на экзамене (0-100). В файле не более 100 строк. Вывести в файл `best.txt` список тех, кто получил более 75 баллов.

Файл `marks.txt`:

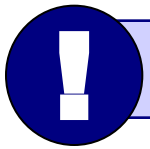
```
Федоров Иван 78
Иванов Федор 63
Анисимов Никита 90
Никитин Николай 55
```



Нужен ли массив!

Результат – файл `best.txt`:

```
Федоров Иван 78
Анисимов Никита 90
```



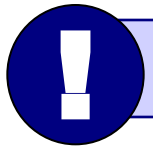
Используем два файла одновременно!

Работа с двумя файлами одновременно

```
var fIn, fOut: Text;
...
assign(fIn, 'marks.txt');
reset(fIn);
assign(fOut, 'best.txt');
rewrite(fOut);
while not eof(fIn) do begin
    { обработка строк из файла }
end;
close(fIn);
close(fOut);
```

Цикл обработки файла

```
var ball: integer;
...
while not eof(fIn) do begin
  readln(fIn, s);
    { обработка строки s }
    { ball:=результат на экзамене }
  if ball > 75 then
    writeln(fOut, s);
end;
```



Оба файла открыты одновременно!

Преобразования «строка»-«число»

Из строки в число:

```
var N, r: integer;
    X: real;
    s: string;
```

```
s := '123';
Val ( s, N, r ); { N = 123 }
 { r = 0, если ошибки не было
   r – номер ошибочного символа}
s := '123.456';
Val ( s, X, r ); { X = 123.456 }
```

Из числа в строку:

```
N := 123;
Str ( N, s );       { '123' }
X := 123.456;
Str ( X, s );       { '1.234560E+002' }
Str ( X:10:3, s ); { '   123.456' }
```

Обработка строки

```
var n, r: integer;
    s, fam, name: string;
```

s:

8	2

```
n:=Pos(' ', s);          { n:=7; }
fam:=Copy(s,1,n-1);      { fam:='Пупкин'; }
Delete(s, 1, n);         { s:='Вася 82'; }
n:=Pos(' ', s);          { n:=5; }
name:=Copy(s,1,n-1);     { name:='Вася'; }
Delete(s, 1, n);           { s:='82'; }
Val(s, ball, r);         { ball:=82; }
```

Задания

«3»: Добавить к списку нумерацию:

```
1) Федоров Иван 78
2) Анисимов Никита 90
```

«4»: Выполнить задачу на «3» и сократить имя до первой буквы:

```
1) Федоров И. 78
2) Анисимов Н. 90
```

«5»: Выполнить задачу на «4», но отсортировать список по алфавиту.

```
1) Анисимов Н. 90
2) Федоров И. 78
```

«6»: Выполнить задачу на «4», но отсортировать список по убыванию отметки (балла).

© К. Поляков, 2006-2011

Конец фильма

ПОЛЯКОВ Константин Юрьевич
д.т.н., учитель информатики высшей категории,
ГОУ СОШ № 163, г. Санкт-Петербург
kpolyakov@mail.ru