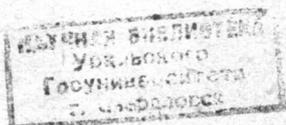


Министерство высшего и среднего специального  
образования РСФСР

Уральский ордена Трудового Красного Знамени  
государственный университет им. А. М. Горького

**ПАСКАЛЬ В СИСТЕМЕ *SCP*  
(ОСНОВНЫЕ СВЕДЕНИЯ)**

Методические указания для студентов  
математико-механического факультета  
и слушателей ФПК



Свердловск  
1989

2-408

29.9.98

2-3315

**Паскаль в системе SCP**  
**Методические указания**  
для студентов математико-механического  
факультета и слушателей ФПК

Редактор **Т. С. Непряхина**  
Технический редактор **Э. А. Максимова**

---

Подписано в печать 01.06.89      Формат 60 x 84 1/16.  
Бумага для множительных аппаратов.      Печать плоская.  
Уч.-изд. л. 2,41.      Усл. печ. л. 2,38.      Заказ 573.  
Тираж 500 экз.      Бесплатно.

Уральский ордена Трудового Красного Знамени государственный университет им. А. М. Горького. Свердловск, пр. Ленина, 51.

---

Типолаборатория УрГУ. Свердловск, пр. Ленина, 51.

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ РСФСР  
УРАЛЬСКИЙ ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ ИМ. А. М. ГОРЬКОГО

ПАСКАЛЬ В СИСТЕМЕ ССР  
(ОСНОВНЫЕ СВЕДЕНИЯ)

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ  
МАТЕМАТИКО-МЕХАНИЧЕСКОГО ФАКУЛЬТЕТА И СЛУШАТЕЛЕЙ ФПК

СВЕРДЛОВСК  
1989

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПОДГОТОВЛЕНЫ  
КАФЕДРОЙ ИНФОРМАТИКИ И ПРОЦЕССОВ УПРАВЛЕНИЯ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ СОДЕРЖАТ СВЕДЕНИЯ, ИСПОЛЬЗУЕМЫЕ ПРИ ОПИСАНИИ АЛГОРИТМОВ, НЕ ТРЕБУЮЩИХ СЛОЖНЫХ ТИПОВ ДАННЫХ, ХРАНЕНИЯ ИНФОРМАЦИИ ВО ВНЕШНЕЙ ПАМЯТИ, АДРЕСОВАНЫ СТУДЕНТАМ МАТЕМАТИКО-МЕХАНИЧЕСКОГО ФАКУЛЬТЕТА, ИЗУЧАЮЩИМ КУРС "ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ И ПРОГРАММИРОВАНИЕ".

СОСТАВИТЕЛЬ: ПЕТЕЛИНА И. С.

РЕЦЕНЗЕНТ: СОЛОВЬЕВА Л. А.

© УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ, 1989



В ЯЗЫКЕ ИМЕЮТСЯ ИДЕНТИФИКАТОРЫ С ЗАРАНЕЕ ОПРЕДЕЛЕННЫМ (ПРЕДОПРЕДЕЛЕННЫМ) СМЫСЛОМ (INTEGER, SIN,...). ОНИ МОГУТ ИСПОЛЬЗОВАТЬСЯ В КАЧЕСТВЕ ИДЕНТИФИКАТОРОВ, ОПРЕДЕЛЯЕМЫХ В ПРОГРАММЕ, НО ТОГДА ПРОГРАММА НЕ МОЖЕТ ПОЛЬЗОВАТЬСЯ ИМИ В ПРЕДОПРЕДЕЛЕННОМ СМЫСЛЕ (ПЕРЕЧЕНЬ ПРЕДОПРЕДЕЛЕННЫХ ИДЕНТИФИКАТОРОВ ПРИВЕДЕН В ПРИЛОЖЕНИИ 2).

МЕТКА ИСПОЛЬЗУЕТСЯ ОПЕРАТОРОМ GOTO ДЛЯ ПЕРЕХОДА, ИМЕЕТ ВИД ЦЕЛОГО ЧИСЛА БЕЗ ЗНАКА ИЛИ ИДЕНТИФИКАТОРА, ПОМЕЩАЕТСЯ ПЕРЕД ОПЕРАТОРОМ И ОТДЕЛЯЕТСЯ ОТ НЕГО СИМВОЛОМ ":". НАПРИМЕР, ?1: A:= X\*X; ER:WRITE ('ОШИБКА'); GOTO 32767; ?:Z:=Y).

КОНСТАНТА - ВЕЛИЧИНА, ЗНАЧЕНИЕ КОТОРОЙ ПРИ ВЫПОЛНЕНИИ ПРОГРАММЫ НЕ ИЗМЕНЯЕТСЯ. ОБРАЩЕНИЕ К КОНСТАНТЕ ЗАДАЕТСЯ УКАЗАНИЕМ САМОЙ КОНСТАНТЫ ПО ПРАВИЛАМ, ЗАВИСЯЩИМ ОТ ЕЕ ТИПА, ЛИБО УКАЗАНИЕМ ИДЕНТИФИКАТОРА КОНСТАНТЫ, ПРИСВОЕННОГО ЕЙ В РАЗДЕЛЕ CONST (СМ. 1.7).

### 1.3. ТИПЫ ДАННЫХ

В ПАСКАЛЕ С КАЖДЫМ ТИПОМ СВЯЗЫВАЕТСЯ МНОЖЕСТВО ЗНАЧЕНИЙ, ПРИНИМАЕМЫХ ДАННЫМИ ЭТОГО ТИПА, И МНОЖЕСТВО ОПЕРАЦИЙ, КОТОРЫЕ ДОПУСКАЕТСЯ ПРИМЕНЯТЬ К ДАННЫМ.

ЯЗЫК, РЕАЛИЗУЕМЫЙ В СИСТЕМЕ ТУРБО ПАСКАЛЬ, ДОПУСКАЕТ ИСПОЛЬЗОВАНИЕ ТИПОВ ДАННЫХ, УКАЗАННЫХ НА РИС.1.

СРЕДИ ТИПОВ ИМЕЕТСЯ НЕСКОЛЬКО ТИПОВ, МНОЖЕСТВА ЗНАЧЕНИЙ, МНОЖЕСТВА ОПЕРАЦИЙ, ИДЕНТИФИКАТОРЫ КОТОРЫХ ОПРЕДЕЛЕННЫ В ЯЗЫКЕ - ПРЕДОПРЕДЕЛЕННЫЕ ТИПЫ. ТИПЫ BYTE, STRING [ ] - НЕ ЯВЛЯЮТСЯ ПРЕДОПРЕДЕЛЕННЫМИ В СТАНДАРТЕ ЯЗЫКА. ПЕРЕЧИСЛИМЫЙ ТИП И ТИП ДИАПАЗОН ОПРЕДЕЛЯЮТСЯ В ПРОГРАММЕ.

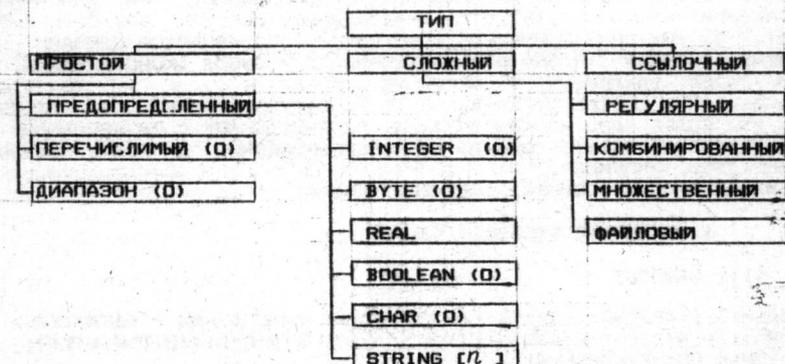


РИС. 1

СРЕДИ ПРОСТЫХ ТИПОВ ВЫДЕЛЯЮТ ТИПЫ, ДЛЯ ЗНАЧЕНИЙ КОТОРЫХ ОПРЕДЕЛЕНА ФУНКЦИЯ ORD - ПОРЯДКОВЫЙ НОМЕР ЗНАЧЕНИЯ В МНОЖЕСТВЕ ДОПУСТИМЫХ ЗНАЧЕНИЙ ТИПА. ТАКИЕ ТИПЫ НАЗЫВАЮТ ОРДИНАЛЬНЫМИ (НА РИС.1 ОНИ ОБОЗНАЧЕНЫ "(0)"). ДЛЯ ЗНАЧЕНИЯ ОРДИНАЛЬНОГО ТИПА ОПРЕДЕЛЕНА ФУНКЦИЯ PRED, SUCC (ЗНАЧЕНИЯ С ПРЕДЫДУЩИМ И ПОСЛЕДУЮЩИМ ПОРЯДКОВЫМИ НОМЕРАМИ) ИЛИ ОДНА ИЗ НИХ.

СЛОЖНЫЕ ТИПЫ СТРОЯТСЯ ИЗ ПРОСТЫХ.

## 1.4. ПЕРЕМЕННЫЕ

КАЖДАЯ ПЕРЕМЕННАЯ, ИСПОЛЬЗУЕМАЯ В ПРОГРАММЕ, ДОЛЖНА БЫТЬ ОПИСАНА. ОПИСАНИЕ ПЕРЕМЕННОЙ - УКАЗАНИЕ ЕЕ ИДЕНТИФИКАТОРА И ТИПА ЗНАЧЕНИЙ (НАПРИМЕР, VAR A, B: REAL;).

## 1.5. ПРЕДОПРЕДЕЛЕННЫЕ ТИПЫ ДАННЫХ

### 1.5.1. ВЕЩЕСТВЕННЫЙ ТИП (REAL)

ЗНАЧЕНИЕ ВЕЩЕСТВЕННОГО ТИПА - ЧИСЛО ИЗ ДИАПАЗОНА  $10^{-38} < |X| < 10^{38}$  ПРЕДСТАВЛЕННОЕ В ПОЛЕ ИЗ 6 БАЙТОВ В НОРМАЛИЗОВАННОМ ВИДЕ С 11 ЗНАЧАЩИМИ ЦИФРАМИ.

КОНСТАНТЫ ТИПА:

А) С ФИКСИРОВАННОЙ ТОЧКОЙ (1.0, -0.018, 432.98341),

В) В ЭКСПОНЕНЦИАЛЬНОЙ ФОРМЕ (-1E-6, 8.34E6, 5E+8).

ДЛЯ ДАННЫХ ВЕЩЕСТВЕННОГО ТИПА ОПРЕДЕЛЕНА АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ СЛОЖЕНИЯ ("+"), ВЫЧИТАНИЯ ("-"), УМНОЖЕНИЯ ("\*"), ДЕЛЕНИЯ ("/"). ЯЗЫК ВКЛЮЧАЕТ НЕСКОЛЬКО ПРЕДОПРЕДЕЛЕННЫХ ФУНКЦИЙ С АРГУМЕНТАМИ ВЕЩЕСТВЕННОГО ТИПА (НАПРИМЕР, SQRT(X) - ВЗЯТИЕ КОРНЯ КВАДРАТНОГО ИЗ X; SQR(X) - ВОЗВЕДЕНИЕ X В КВАДРАТ). ПЕРЕЧЕНЬ ЭТИХ ФУНКЦИЙ ПРИВЕДЕН В ПРИЛОЖЕНИИ 3.

ПЕРЕПОЛНЕНИЕ ПРИ ВЫГЛАЖЕНИИ ОПЕРАЦИИ НАД ВЕЩЕСТВЕННЫМИ ДАННЫМИ ВЫЗЫВАЕТ АВАРИЙНОЕ СНЯТИЕ ЗАДАЧИ.

### 1.5.2. ЦЕЛЫЙ ТИП (INTEGER)

ЗНАЧЕНИЕ ЦЕЛОГО ТИПА - ЦЕЛОЕ ЧИСЛО ИЗ ДИАПАЗОНА [-32768, 32767], РАЗМЕЩАЕМОЕ В ПОЛЕ ПАМЯТИ ИЗ ДВУХ БАЙТОВ.

ЦЕЛАЯ КОНСТАНТА МОЖЕТ БЫТЬ ПРЕДСТАВЛЕНА

А) ДЕСЯТИЧНЫМ ЧИСЛОМ (1, -5432, 0),

Б) ШЕСТНАДЦАТЕРИЧНОЙ ЗАПИСЬЮ ВНУТРЕННЕГО ПРЕДСТАВЛЕНИЯ ЧИСЛА (80, 81E7, 8FFFF).

ДЛЯ ЦЕЛЫХ ДАННЫХ ОПРЕДЕЛЕНА ОПЕРАЦИИ СЛОЖЕНИЯ ("+"), ВЫЧИТАНИЯ ("-"), УМНОЖЕНИЯ ("\*"), ПОЛУЧЕНИЯ ЦЕЛОЙ ЧАСТИ ЧАСТНОГО (DIV), ПОЛУЧЕНИЯ ОСТАТКА ОТ ДЕЛЕНИЯ (MOD). СРЕДИ ПРЕДОПРЕДЕЛЕННЫХ ФУНКЦИЙ РЯД ФУНКЦИЙ ИМЕЕТ АРГУМЕНТ ИЛИ РЕЗУЛЬТАТ ЦЕЛОГО ТИПА (НАПРИМЕР, ФУНКЦИЯ SQR(I) ДЛЯ ЦЕЛОГО АРГУМЕНТА I ДАЕТ ЦЕЛЫЙ РЕЗУЛЬТАТ: I<sup>2</sup>; SQRT(I) ДЛЯ ЦЕЛОГО I ДАЕТ ВЕЩЕСТВЕННЫЙ РЕЗУЛЬТАТ:  $\sqrt{I}$ ; ORD(I)=I (ORD(10)=10, ORD(-31)=-31); SUCC(I)=I+1; PRED(I)=I-1 (СМ. ПРИЛОЖЕНИЕ 3).

ПРИ ВЫПОЛНЕНИИ ОПЕРАЦИИ НАД ЦЕЛЫМИ ДАННЫМИ ПЕРЕПОЛНЕНИЕ НЕ ФИКСИРУЕТСЯ.

### 1.5.3. БАЙТОВЫЙ ТИП (BYTE)

ДОПУСТИМЫМИ ЗНАЧЕНИЯМИ ДАННЫХ БАЙТОВОГО ТИПА ЯВЛЯЮТСЯ ЦЕЛЫЕ ЧИСЛА ИЗ ПРОМЕЖУТКА [0, 255]. БАЙТОВЫЙ ТИП - ПОДМНОЖЕСТВО ЦЕЛОГО ТИПА. ВЕЛИЧИНА БАЙТОВОГО ТИПА СОВМЕСТИМА С ЦЕЛЫМИ ВЕЛИЧИНАМИ, Т.Е. МОЖЕТ ИСПОЛЬЗОВАТЬСЯ ВЕЗДЕ, ГДЕ ДОПУСТИМА ВЕЛИЧИНА ТИПА INTEGER. ЗНАЧЕНИЕ БАЙТОВОГО ТИПА ЗАНИМАЕТ 1 БАЙТ ПАМЯТИ.

### 1.5.4. БУЛЕВСКИЙ ТИП (BOOLEAN)

ВЕЛИЧИНА БУЛЕВСКОГО ТИПА ПРИНИМАЕТ ОДНО ИЗ ДВУХ ЗНАЧЕНИЙ: ИСТИНА (TRUE), ЛОЖЬ (FALSE). FALSE, TRUE - ПРЕДОПРЕДЕЛЕННЫЕ КОНСТАНТЫ ТИПА. ПО ОПРЕДЕЛЕНИЮ FALSE < TRUE. ЗНАЧЕНИЕ ДАННОГО ЗАНИМАЕТ 1 БАЙТ ПАМЯТИ.

ДЛЯ ДАННЫХ БУЛЕВСКОГО ТИПА ОПРЕДЕЛЕНА ОПЕРАЦИИ NOT, AND, OR, XOR. ЕСЛИ A, B ИМЕЮТ БУЛЕВСКИЙ ТИП, ТО

A	!	B	!	NOT A	!	A AND B	!	A OR B	!	A XOR B
FALSE	!	FALSE	!	TRUE	!	FALSE	!	FALSE	!	FALSE
FALSE	!	TRUE	!	TRUE	!	FALSE	!	TRUE	!	TRUE
TRUE	!	FALSE	!	FALSE	!	FALSE	!	TRUE	!	TRUE
TRUE	!	TRUE	!	FALSE	!	TRUE	!	TRUE	!	FALSE

(NOT - ЛОГИЧЕСКОЕ ОТРИЦАНИЕ, AND - ЛОГИЧЕСКОЕ "И", OR - ЛОГИЧЕСКОЕ "ИЛИ", XOR - ОТРИЦАНИЕ РАВНОЗНАЧНОСТИ).

ПРЕДОПРЕДЕЛЕННАЯ ФУНКЦИЯ ODD, ОПРЕДЕЛЕННАЯ ДЛЯ АРГУМЕНТА ЦЕЛОГО ТИПА, ВОЗВРАЩАЕТ ЗНАЧЕНИЕ TRUE, ЕСЛИ АРГУМЕНТ НЕЧЕТЕН, И FALSE, ЕСЛИ ЧЕТЕН.

БУЛЕВСКИЕ ЗНАЧЕНИЯ ИМЕЮТ ЗНАЧЕНИЯ РЕЗУЛЬТАТОВ ОПЕРАЦИИ ОТНОШЕНИЯ, ЗАДАВАЕМЫХ ЗНАКАМИ: "=" (РАВНО), "<" (НЕ РАВНО), "<" (МЕНЬШЕ), "<=" (НЕ БОЛЬШЕ), ">" (БОЛЬШЕ), ">=" (НЕ МЕНЬШЕ). ОПЕРАЦИЯ ОТНОШЕНИЯ ОПРЕДЕЛЕНА ЛИБО ДЛЯ ЧИСЛОВЫХ, ЛИБО ДЛЯ БУЛЕВСКИХ ОПЕРАНДОВ.

ЕСЛИ A=-8.7, B=13.82, L=FALSE, ТО ЗНАЧЕНИЕМ A<B ЯВЛЯЕТСЯ TRUE, ЗНАЧЕНИЕМ A>B ЯВЛЯЕТСЯ FALSE, ЗНАЧЕНИЕМ L>FALSE ЯВЛЯЕТСЯ FALSE.

ОПЕРАЦИИ NOT, AND, OR, XOR ПРИМЕНИМЫ ДЛЯ ОПЕРАНДОВ ЦЕЛОГО ТИПА (ДАЮТ ЦЕЛЫЙ РЕЗУЛЬТАТ). РЕЗУЛЬТАТОМ ОПЕРАЦИИ NOT ДЛЯ ЦЕЛОГО ОПЕРАНДА ЯВЛЯЕТСЯ ИНВЕРСИЯ ЕГО ВНУТРЕННЕГО ПРЕДСТАВЛЕНИЯ: NOT %371=%FC8E, NOT (-15)=NOT %FFEB=14. ОПЕРАЦИИ AND, OR, XOR ВЫПОЛНЯЮТСЯ НАД СООТВЕТСТВУЮЩИМИ ДВОИЧНЫМИ РАЗРЯДАМИ ЦЕЛЫХ ОПЕРАНДОВ, ПРИЧЕМ НУЛЕВОЕ ЗНАЧЕНИЕ РАЗРЯДА СООТВЕТСТВУЕТ FALSE, ЕДИНИЧНОЕ - TRUE: %45 AND %3F=%5; %45 OR %3F=%7F; %45 XOR %3F=%3A.

#### 1.5.5. ВЫРАЖЕНИЕ

ВЫРАЖЕНИЕ СТРОИТСЯ ИЗ КОНСТАНТ, ПЕРЕМЕННЫХ, УКАЗАТЕЛЕЙ ФУНКЦИИ, ВЫРАЖЕНИЙ В КРУГЛЫХ СКОБКАХ С ПОМОЩЬЮ ЗНАКОВ ОПЕРАЦИИ. ЗНАКИ ОПЕРАЦИИ (ПО УБЫВАНИЮ ПРИОРИТЕТА):

- A) - (УНАРНАЯ), NOT;
- B) \*, /, DIV, MOD, AND (ОПЕРАЦИИ ГРУППЫ УМНОЖЕНИЯ);
- C) +, =, SUB, XOR, IN (ОПЕРАЦИИ ГРУППЫ СЛОЖЕНИЯ);
- D) <, <=, >, >=, =, <> (ОПЕРАЦИИ ОТНОШЕНИЯ).

ОПЕРАЦИИ DIV, MOD ОПРЕДЕЛЕНА ТОЛЬКО ДЛЯ ЦЕЛЫХ ОПЕРАНДОВ. ОПЕРАЦИЯ "/" ДЛЯ ОПЕРАНДОВ ЛИБОГО ЧИСЛОВОГО ТИПА (BYTE, INTEGER, REAL) ДАЕТ ВЕЩЕСТВЕННЫЙ РЕЗУЛЬТАТ. БИНАРНЫЕ АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ (КРОМЕ DIV, MOD) ДОПУСКАЮТ ИСПОЛЬЗОВАНИЕ ОПЕРАНДОВ РАЗЛИЧНЫХ ЧИСЛОВЫХ ТИПОВ. РЕЗУЛЬТАТ ОПЕРАЦИИ ИМЕЕТ ТИП REAL, ЕСЛИ ХОТЯ БЫ ОДИН ИЗ ОПЕРАНДОВ ИМЕЕТ ТИП REAL.

СРЕДИ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ И ПРЕДОПРЕДЕЛЕННЫХ ФУНКЦИЙ НЕТ СРЕДСТВ ДЛЯ ВЫЧИСЛЕНИЯ ЗНАЧЕНИЯ  $x^y$ . ЕСЛИ  $x > 0$ , МОЖНО ИСПОЛЬЗОВАТЬ ФУНКЦИЮ EXP, LN:  $x^y \Rightarrow \text{EXP}(\text{LN}(x)*y)$ .

ОПРЕДЕЛЕНА ОПЕРАЦИЯ ПРИ ВЫЧИСЛЕНИИ ЗНАЧЕНИЯ ВЫРАЖЕНИЯ ОПРЕДЕЛЯЕТСЯ КРУГЛЫМИ СКОБКАМИ И ПРИОРИТЕТОМ ОПЕРАЦИИ.

ВЫРАЖЕНИЕ С РЕЗУЛЬТАТОМ ТИПА BOOLEAN НАЗЫВАЕТСЯ БУЛЕВСКИМ ИЛИ ЛОГИЧЕСКИМ, ВЫРАЖЕНИЕ С ЧИСЛОВЫМ РЕЗУЛЬТАТОМ - АРИФМЕТИЧЕСКИМ.

#### 1.6. СТРУКТУРА ПРОГРАММ

##### ПРИМЕР ПРОГРАММЫ

```
PROGRAM CALPOWER;
CONST EPS=1E-7; (* РАЗДЕЛ ОПРЕДЕЛЕНИЯ КОНСТАНТ *)
```

```

VAR      (* РАЗДЕЛ ОПИСАНИЯ ПЕРЕМЕННЫХ *)
N: INTEGER; (* ПОКАЗАТЕЛЬ СТЕПЕНИ *)
X: REAL;   (* ОСНОВАНИЕ СТЕПЕНИ *)
Z: REAL;   (* РЕЗУЛЬТАТ *)
M, K: INTEGER;
Y: REAL;
B: BOOLEAN;
(* РАЗДЕЛ ОПЕРАТОРОВ *)
BEGIN
WRITELN('ВЫЧИСЛЕНИЕ X**N. ':20);
WRITE('ЧЕРЕЗ ПРОБЕЛ НАБЕРИТЕ ЗНАЧЕНИЯ X,N, НАЖМИТЕ <ET>');
READLN(X,N); B:=TRUE;
IF (X=0) AND (N=0)
THEN Z:=0
ELSE
BEGIN
M:=ABS(N); Z:=1; Y:=X;
WHILE M>0 DO
BEGIN
WHILE NOT ODD(M) DO
BEGIN
Y:=SQR(Y); M:=M DIV 2
END; (* WHILE NOT ODD(M) *)
Z:=Z*Y; M:=M-1
END (* WHILE M>0 *)
END; (* ELSE *)
IF N<0 THEN (* ОТРИЦАТЕЛЬНЫЙ ПОКАЗАТЕЛЬ СТЕПЕНИ *)
IF ABS(Z)>EPS THEN Z:=1E0/Z ELSE B:=FALSE;
(* ВЫДАЧА РЕЗУЛЬТАТА *)
IF B THEN WRITELN(X,'**',N,'=',Z)
ELSE WRITELN('СТЕПЕНЬ НЕ ОПРЕДЕЛЕНА')
END. (* CALPOWER *)

```

ПРОГРАММА НА ЯЗЫКЕ ПАСКАЛЬ - ОДНА ПРОГРАММНАЯ ЕДИНИЦА, СОСТОИТ ИЗ ЗАГЛОВОКА, ПРОГРАММНОГО БЛОКА, ЗАКАНЧИВАЕТСЯ СИМВОЛОМ ".\*".

<ЗАГЛОВОК> ::= PROGRAM <ИДЕНТИФИКАТОР>;  
ОПРЕДЕЛЯЕТ ИМЯ ПРОГРАММЫ (НАПРИМЕР, PROGRAM NORMA);  
ПРОГРАММНЫЙ БЛОК СОСТОИТ ИЗ РАЗДЕЛОВ, ОТНОСЯЩИХСЯ К ОПИСАТЕЛЬНОЙ И ВЫПОЛНЯЕМОЙ ЧАСТЯМ БЛОКА. В ОПИСАТЕЛЬНОЙ ЧАСТИ СООБЩАЮТСЯ СВЕДЕНИЯ ОБ ОБЪЕКТАХ, ИСПОЛЪЗУЕМЫХ В БЛОКЕ; В ВЫПОЛНЯЕМОЙ ЧАСТИ ОПИСЫВАЕТСЯ АЛГОРИТМ, РЕАЛИЗУЕМЫЙ БЛОКОМ.

#### ОПИСАТЕЛЬНАЯ ЧАСТЬ БЛОКА

##### 1.6.1. РАЗДЕЛ ОПИСАНИЯ МЕТКИ

```
[ LABEL <МЕТКА> { , <МЕТКА> } ; ]
```

НАПРИМЕР, LABEL 1, M10, ERR;

КАЖДАЯ МЕТКА, ИСПОЛЪЗУЕМАЯ В ВЫПОЛНЯЕМОЙ ЧАСТИ БЛОКА, ДОЛЖНА БЫТЬ ВКЛЮЧЕНА В СПИСОК РАЗДЕЛА LABEL.

##### 1.6.2 РАЗДЕЛ ОПРЕДЕЛЕНИЯ КОНСТАНТ

```
[ CONST <ОПРЕДЕЛЕНИЕ КОНСТАНТЫ>; { <ОПРЕДЕЛЕНИЕ КОНСТАНТЫ>; } ]
```

<ОПРЕДЕЛЕНИЕ КОНСТАНТЫ> ::= <ИДЕНТИФИКАТОР КОНСТАНТЫ> = <ЗНАЧЕНИЕ КОНСТАНТЫ>

В КАЧЕСТВЕ ЗНАЧЕНИЯ КОНСТАНТЫ МОЖЕТ ИСПОЛЬЗОВАТЬСЯ САМА КОНСТАНТА, ИДЕНТИФИКАТОР РАНЕЕ ОПРЕДЕЛЕННОЙ КОНСТАНТЫ С УНАРНЫМ ЗНАКОМ "-" ИЛИ БЕЗ ЗНАКА. НАПРИМЕР, CONST NMAX=100; EPS=1E-7; NMIN=-NMAX;

ЕСЛИ КОНСТАНТА ЧАСТО ВСТРЕЧАЕТСЯ В ТЕКСТЕ ИЛИ СО ВРЕМЕНЕМ МОЖЕТ БЫТЬ ИЗМЕНЕНА, УДОБНО ПРИСВОИТЬ ЕЙ ИДЕНТИФИКАТОР. ТОГДА В НУЖНОМ МЕСТЕ ПРОГРАММЫ ВМЕСТО КОНСТАНТЫ ПОМЕЩАЕТСЯ ЭТОТ ИДЕНТИФИКАТОР. ИЗМЕНЕНИЕ КОНСТАНТЫ СОПРОВОЖДАЕТСЯ ИЗМЕНЕНИЕМ ТОЛЬКО РАЗДЕЛА CONST.

В ПРОГРАММЕ НЕДОПУСТИМО ПРИСВАИВАНИЕ ЗНАЧЕНИЯ ИМЕНОВАННЫМ КОНСТАНТАМ.

ТИП ЗНАЧЕНИЯ ИМЕНОВАННОЙ КОНСТАНТЫ ОПРЕДЕЛЯЕТСЯ ПО ТИПУ ЗНАЧЕНИЯ, УКАЗАННОГО В ПРАВОЙ ЧАСТИ ЕЕ ОПРЕДЕЛЕНИЯ: КОНСТАНТЫ NMAX, NMIN ЦЕЛОГО ТИПА, КОНСТАНТА EPS ТИПА REAL.

ПАСКАЛЬ ИСПОЛЬЗУЕТ ИМЕНОВАННЫЕ ПРЕОПРЕДЕЛЕННЫЕ КОНСТАНТЫ. СРЕДИ НИХ, КРОМЕ КОНСТАНТ FALSE, TRUE СО СМЫСЛОМ "ЛОЖЬ" И "ИСТИНА", КОНСТАНТЫ PI, MAXINT, ЗНАЧЕНИЯМИ КОТОРЫХ ЯВЛЯЮТСЯ 3.1415926536E+00 ТИПА REAL И 32767 ТИПА INTEGER.

#### 1.6.3. РАЗДЕЛ ОПРЕДЕЛЕНИЯ ТИПОВ:

```
[ TYPE <ОПРЕДЕЛЕНИЕ ТИПА>; { <ОПРЕДЕЛЕНИЕ ТИПА>; } ]
<ОПРЕДЕЛЕНИЕ ТИПА> ::= <ИДЕНТИФИКАТОР ТИПА> = <ТИП>
```

НАПРИМЕР, TYPE IND=1..10;  
РАЗДЕЛ ПРЕДНАЗНАЧЕН ДЛЯ ОПРЕДЕЛЕНИЯ ТИПОВ, ИСПОЛЬЗУЕМЫХ В ПРОГРАММЕ И ОТЛИЧАЮЩИХСЯ ОТ ПРЕОПРЕДЕЛЕННЫХ ТИПОВ.

#### 1.6.4. РАЗДЕЛ ОПИСАНИЯ ПЕРЕМЕННЫХ:

```
[ VAR <СЕКМЕНТ ОПИСАНИЯ ПЕРЕМЕННЫХ>;
  { <СЕКМЕНТ ОПИСАНИЯ ПЕРЕМЕННЫХ>; } ]
<СЕКМЕНТ ОПИСАНИЯ ПЕРЕМЕННЫХ> ::= <ИДЕНТИФИКАТОР ПЕРЕМЕННОЙ> { ,
<ИДЕНТИФИКАТОР ПЕРЕМЕННОЙ> } : <ТИП>
```

НАПРИМЕР,

```
VAR L,L1: BOOLEAN; (*СЕКМЕНТ ОПИСАНИЯ ПЕРЕМЕННЫХ*)
    X,Y,Z: REAL; (*СЕКМЕНТ ОПИСАНИЯ ПЕРЕМЕННЫХ*)
    V: BOOLEAN; (*ОПИСАНИЕ ПЕРЕМЕННОЙ*)
```

КАЖДАЯ ПЕРЕМЕННАЯ, ИСПОЛЬЗУЕМАЯ В ПРОГРАММЕ, ДОЛЖНА БЫТЬ ОПИСАНА В РАЗДЕЛЕ VAR. ОПИСАНИЕ ПЕРЕМЕННОЙ - УКАЗАНИЕ ЕЕ ИДЕНТИФИКАТОРА И ТИПА.

#### 1.6.5. РАЗДЕЛ ОПИСАНИЯ ПОДПРОГРАММ

ПАСКАЛЬ ДОПУСКАЕТ ИСПОЛЬЗОВАНИЕ ТОЛЬКО ВНУТРЕННИХ ПОДПРОГРАММ. КАЖДАЯ ПОДПРОГРАММА, К КОТОРОЙ ЕСТЬ ОБРАЩЕНИЕ В ВЫПОЛНЯЕМОЙ ЧАСТИ БЛОКА, КРОМЕ ПРЕОПРЕДЕЛЕННЫХ ПОДПРОГРАММ, ДОЛЖНА БЫТЬ ОПИСАНА В РАЗДЕЛЕ ОПИСАНИЯ ПОДПРОГРАММ БЛОКА.

ПОДПРОГРАММА МОЖЕТ БЫТЬ ОРГАНИЗОВАНА В ВИДЕ ПРОЦЕДУРЫ ИЛИ ФУНКЦИИ. ОБРАЩЕНИЕ К ВЫПОЛНЕНИЮ ПРОЦЕДУРЫ ЗАДАЕТ ОПЕРАТОР ВЫЗОВА ПРОЦЕДУРЫ:

```
<ИДЕНТИФИКАТОР ПРОЦЕДУРЫ> [ (<СПИСОК ФАКТИЧЕСКИХ АРГУМЕНТОВ> ) ],
ОБРАЩЕНИЕ К ВЫПОЛНЕНИЮ ФУНКЦИИ ЗАДАЕТ УКАЗАТЕЛЬ ФУНКЦИИ, ИСПОЛЬЗУЕМЫЙ КАК ОПЕРАНД ВЫРАЖЕНИЯ:
<ИДЕНТИФИКАТОР ФУНКЦИИ> (<СПИСОК ФАКТИЧЕСКИХ АРГУМЕНТОВ>).
```

ВЫПОЛНЯЕМАЯ ЧАСТЬ БЛОКА СОСТОИТ ИЗ ОДНОГО РАЗДЕЛА - РАЗДЕЛА ОПЕРАТОРОВ.

#### 1.6.6. РАЗДЕЛ ОПЕРАТОРОВ:

```
BEGIN
  <ОПЕРАТОР> { ; <ОПЕРАТОР> }
```

END

УСЛОВНО ОПЕРАТОРЫ ПОДРАЗДЕЛЯЮТСЯ НА ПРОСТЫЕ, СОСТАВНЫЕ И СЛОЖНЫЕ.

ПРОСТОЙ ОПЕРАТОР НЕ СОДЕРЖИТ ВНУТРИ СЕБЯ ДРУГИХ ОПЕРАТОРОВ.  
<ПРОСТОЙ ОПЕРАТОР> ::= <ОПЕРАТОР ПРИСВАИВАНИЯ>! <ОПЕРАТОР ПЕРЕХОДА>!<ПУСТОЙ ОПЕРАТОР>! <ОПЕРАТОР ВЫЗОВА ПРОЦЕДУРЫ>

СОСТАВНОЙ ОПЕРАТОР:

BEGIN <ОПЕРАТОР> { ; <ОПЕРАТОР> } END

ИСПОЛЬЗУЕТСЯ ТОГДА, КОГДА НА ГРУППУ ПОСЛЕДОВАТЕЛЬНЫХ ОПЕРАТОРОВ НАДО СМОТРЕТЬ КАК НА ОДНО ЦЕЛОЕ.

РАЗДЕЛ ОПЕРАТОРОВ БЛОКА ЯВЛЯЕТСЯ СОСТАВНЫМ ОПЕРАТОРОМ. В ТОМ МЕСТЕ, ГДЕ СИНТАКСИС ДОПУСКАЕТ ПОМЕЩЕНИЕ ТОЛЬКО ОДНОГО ОПЕРАТОРА, А АЛГОРИТМ ТРЕБУЕТ ВЫПОЛНЕНИЯ ДЕЙСТВИЯ, ОПИСЫВАЕМЫХ ГРУППОЙ ОПЕРАТОРОВ, ИСПОЛЬЗУЕТСЯ СОСТАВНОЙ ОПЕРАТОР (НАПРИМЕР, В ОПЕРАТОРАХ IF, FOR).

СЛОЖНЫЙ ОПЕРАТОР ВКЛЮЧАЕТ В СЕБЯ ОДИН ИЛИ НЕСКОЛЬКО ДРУГИХ ОПЕРАТОРОВ (IF, FOR, WHILE, REPEAT, CASE, WITH).

ПРИ НАПИСАНИИ ПРОГРАММЫ СЛЕДУЕТ ПОМНИТЬ, ЧТО ПРОБЕЛЫ МОЖНО ПОМЕЩАТЬ ТОЛЬКО МЕЖДУ ЛЕКСИЧЕСКИМИ ЕДИНИЦАМИ. СЛУЖЕБНЫЕ СЛОВА В ТЕКСТЕ ДОЛЖНЫ БЫТЬ ВЫДЕЛЕНЫ ПРОБЕЛАМИ ИЛИ ДРУГИМИ РАЗДЕЛИТЕЛЯМИ. НАПРИМЕР, ЗАПИСЬ К MOD 2 В ФОРМЕ KMOD2 НЕДОПУСТИМА.

КОММЕНТАРИЙ ИМЕЕТ ВИД "(\* <ПОСЛЕДОВАТЕЛЬНОСТЬ ЛЮБЫХ ДОПУСТИМЫХ НА ПЭВМ СИМВОЛОВ>\*)", ПОМЕЩАЕТСЯ В ЛЮБОЕ МЕСТО ПРОГРАММЫ, ГДЕ МОЖЕТ БЫТЬ РАЗМЕЩЕН ПРОБЕЛ.

## 1.7. ОСНОВНЫЕ ОПЕРАТОРЫ

### 1.7.1. ОПЕРАТОР ПРИСВАИВАНИЯ :

<ПЕРЕМЕННАЯ> := <ВЫРАЖЕНИЕ>

ОБЕ ЧАСТИ ОПЕРАТОРА ПРИСВАИВАНИЯ ДОЛЖНЫ БЫТЬ ОДНОГО ТИПА. ИЗ ЭТОГО ПРАВИЛА ЕСТЬ ИСКЛЮЧЕНИЕ: ПЕРЕМЕННОЙ ВЕЩЕСТВЕННОГО ТИПА МОЖНО ПРИСВОИТЬ ЗНАЧЕНИЕ ВЫРАЖЕНИЯ ЦЕЛОГО ТИПА.

НАПРИМЕР,

VAR I: INTEGER; X, Y, Z: REAL;

BEGIN

I:=0; X:=Y\*SIN(X/2); Z:=100-I;

### 1.7.2. ОПЕРАТОР ПЕРЕХОДА

GOTO <МЕТКА>

МЕТКА ДОЛЖНА БЫТЬ ВКЛЮЧЕНА В СПИСОК МЕТОК РАЗДЕЛА LABEL ПРОГРАММНОГО БЛОКА. ОБЛАСТЬ ДЕЙСТВИЯ МЕТКИ - БЛОК, В КОТОРОМ МЕТКА ОПИСАНА. НАПРИМЕР,

LABEL 15; ... VAR X, Y: REAL; ... GOTO 15; ... 15: Y:=X-2;

ЯЗЫК ПАСКАЛЬ ИМЕЕТ ДОСТАТОЧНО СРЕДСТВ ДЛЯ ОПИСАНИЯ АЛГОРИТМОВ БЕЗ ИСПОЛЬЗОВАНИЯ ОПЕРАТОРА ПЕРЕХОДА. НО ПРИ ВКЛЮЧЕНИИ В ПРОГРАММУ ОПЕРАТОРА GOTO СЛЕДУЕТ ПОМНИТЬ, ЧТО В СИСТЕМЕ TURBO ПАСКАЛЬ КОНТРОЛЬ ЗА ДОПУСТИМОСТЬЮ ПЕРЕХОДА ПО МЕТКЕ ОЧЕНЬ СЛАБЫЙ (НАПРИМЕР, ПЕРЕХОД НА ТЕЛО ЦИКЛА, МИНУЯ ЕГО ЗАГОЛОВОК FOR, НЕ ВЫЗЫВАЕТ ДИАГНОСТИКИ НИ НА ЭТАПЕ ТРАНСЛЯЦИИ, НИ НА ЭТАПЕ ВЫПОЛНЕНИЯ ПРОГРАММЫ).

### 1.7.3. УСЛОВНЫЙ ОПЕРАТОР:

IF <БУЛЕВСКОЕ ВЫРАЖЕНИЕ> THEN <ОПЕРАТОР 1> [ ELSE <ОПЕРАТОР 2> ]

ДЕЙСТВИЯ, ОПРЕДЕЛЯЕМЫЕ ОПЕРАТОРОМ:

А) ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ВЫРАЖЕНИЯ,

Б) ЕСЛИ РЕЗУЛЬТАТ - TRUE, ВЫПОЛНЕНИЕ ОПЕРАТОРА 1,

В) ЕСЛИ РЕЗУЛЬТАТ - FALSE И ЧАСТЬ С ELSE В ОПЕРАТОРЕ ЗАДАНА, ВЫПОЛНЕНИЕ ОПЕРАТОРА 2.

НАПРИМЕР, ОПЕРАТОР IF X<0 THEN Y:=X\*SIN(X) ELSE Y:=-X-1  
ОПИСЫВАЕТ ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ

$$Y = \begin{cases} X \sin(X), & \text{ЕСЛИ } X < 0, \\ X - 1, & \text{ЕСЛИ } X > 0. \end{cases}$$

#### 1.7.4. ОПЕРАТОРЫ ЦИКЛА

##### А. ОПЕРАТОР WHILE:

WHILE <БУЛЕВСКОЕ ВЫРАЖЕНИЕ> DO <ОПЕРАТОР>

ДЕЙСТВИЯ, ОПРЕДЕЛЯЕМЫЕ ОПЕРАТОРОМ:

А) ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ БУЛЕВСКОГО ВЫРАЖЕНИЯ,

Б) ЕСЛИ РЕЗУЛЬТАТ - TRUE, ВЫПОЛНЕНИЕ ОПЕРАТОРА, ВКЛЮЧЕННОГО

В ОПЕРАТОР WHILE, ПЕРЕХОД НА А,

В) ЕСЛИ РЕЗУЛЬТАТ - FALSE, ПЕРЕХОД К ВЫПОЛНЕНИЮ СЛЕДУЮЩЕГО  
ОПЕРАТОРА (ВЫХОД ИЗ ЦИКЛА).

НАПРИМЕР, ЕСЛИ VAR X: REAL; TO ГРУППА ОПЕРАТОРОВ

X:=1E35; XN:=0;

WHILE ABS(XN-XS)>1E-7 DO

BEGIN XS:=XN; XN:=XS-COS(XS) END;

ЗАДАЕТ ВЫЧИСЛЕНИЕ КОРНЯ УРАВНЕНИЯ  $X = \cos X$  С ТОЧНОСТЬЮ  $\epsilon = 10^{-7}$   
МЕТОДОМ ИТЕРАЦИИ.

##### Б. ОПЕРАТОР REPEAT

REPEAT <ОПЕРАТОР> ; <ОПЕРАТОР> UNTIL <БУЛЕВСКОЕ  
ВЫРАЖЕНИЕ>

ДЕЙСТВИЯ, ОПРЕДЕЛЯЕМЫЕ ОПЕРАТОРОМ:

А) ВЫПОЛНЕНИЕ ГРУППЫ ОПЕРАТОРОВ, ЗАКЛЮЧЕННЫХ МЕЖДУ REPEAT И  
UNTIL,

Б) ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ БУЛЕВСКОГО ВЫРАЖЕНИЯ,

В) ЕСЛИ РЕЗУЛЬТАТ - FALSE, ПЕРЕХОД НА А, ИНАЧЕ - ПЕРЕХОД К  
ВЫПОЛНЕНИЮ СЛЕДУЮЩЕГО ОПЕРАТОРА (ВЫХОД ИЗ ЦИКЛА).

ТО ЕСТЬ БУЛЕВСКОЕ ВЫРАЖЕНИЕ В ОПЕРАТОРЕ WHILE ЗАДАЕТ УСЛОВИЕ  
ПОВТОРЕНИЯ ЦИКЛА, В ОПЕРАТОРЕ REPEAT- UNTIL - УСЛОВИЕ ВЫХОДА ИЗ  
ЦИКЛА.

НАПРИМЕР, ЕСЛИ VAR S: REAL; I: INTEGER; TO ГРУППА ОПЕРАТОРОВ

S:=0; I:=0;

REPEAT I:=I+1; S:=S+SQR(I) UNTIL I>100

ЗАДАЕТ ВЫЧИСЛЕНИЕ  $S = \sum_{i=1}^{100} i^2$ .

##### В. ОПЕРАТОР FOR (ОПЕРАТОР ЦИКЛА С ПАРАМЕТРОМ)

FOR  $\alpha := \alpha$  TO/DOWNTO  $\beta$  DO <ОПЕРАТОР>

ЗДЕСЬ  $\alpha$  - ИДЕНТИФИКАТОР ПЕРЕМЕННОЙ - ПАРАМЕТР ЦИКЛА,

$\alpha, \beta$  - ВЫРАЖЕНИЯ.  $\alpha, \alpha, \beta$  ДОЛЖНЫ БЫТЬ ОДНОГО  
ОРДИНАЛЬНОГО ТИПА.

ДЕЙСТВИЯ, ОПРЕДЕЛЯЕМЫЕ ОПЕРАТОРОМ:

А) ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ  $\alpha, \beta$  ;

Б) ПРИСВАИВАНИЕ  $\alpha$  ЗНАЧЕНИЯ  $\alpha$  ;

В) ЕСЛИ  $\alpha$  ДОПУСТИМО, ВЫПОЛНЕНИЕ ОПЕРАТОРА, ВКЛЮЧЕННОГО В  
ОПЕРАТОР FOR, ПРИСВАИВАНИЕ  $\alpha$  НОВОГО ЗНАЧЕНИЯ, ВОЗВРАТ НА В);

Г) ЕСЛИ  $\alpha$  НЕДОПУСТИМО, ПЕРЕХОД К ВЫПОЛНЕНИЮ ОПЕРАТОРА,  
СЛЕДУЮЩЕГО ЗА ОПЕРАТОРОМ FOR. УСЛОВИЕ ПОВТОРЕНИЯ И НОВОЕ ЗНАЧЕНИЕ  
 $\alpha$  ЗАВИСЯТ ОТ СЛУЖЕБНОГО СЛОВА TO ИЛИ DOWNTD.

СЛУЖЕБНОЕ СЛОВО	УСЛОВИЕ ПОВТОРЕНИЯ	НОВОЕ ЗНАЧЕНИЕ
TO	$x \in [\alpha, \beta] \quad (\alpha \leq \beta)$	$x := \text{SUCCESS}(x)$
DOWNTO	$x \in [\beta, \alpha] \quad (\beta \leq \alpha)$	$x := \text{PRED}(x)$

НАПРИМЕР, ЕСЛИ  
 VAR S, S : REAL; I, I1: INTEGER; B: BOOLEAN;  
 TO ГРУППА ОПЕРАТОРОВ  
 S:=0; FOR I:=1 TO 100 DO S:=S+SQR(I);  
 S1:=0; FOR I1:=100 DOWNTO 1 DO S1:=S1+SQR(I1);  
 FOR B:FALSE TO TRUE DO WRITELN(B)

ЗАДАЕТ ВЫЧИСЛЕНИЕ  $S = \sum_{i=1}^{100} i^2$ ,  $S1 = \sum_{i=1}^{100} i^2$ ,

ВЫДАЧУ ЗНАЧЕНИЙ B (FALSE, TRUE).

### 1.8. ОРГАНИЗАЦИЯ ДИАЛОГА "МАШИНА-ПОЛЬЗОВАТЕЛЬ"

ОСНОВНОЕ СРЕДСТВО ОРГАНИЗАЦИИ ДИАЛОГА - ВЫДАЧА ИНФОРМАЦИИ В СТРОКИ ЭКРАНА ДИСПЛЕЯ В ВИДЕ ПОСЛЕДОВАТЕЛЬНОСТЕЙ СИМВОЛОВ И ЧТЕНИЕ ИНФОРМАЦИИ, ПРЕДСТАВЛЕННОЙ В ВИДЕ ПОСЛЕДОВАТЕЛЬНОСТЕЙ СИМВОЛОВ, ИЗ СТРОК ЭКРАНА ДИСПЛЕЯ.

#### 1.8.1. ПРЕДОПРЕДЕЛЕННЫЕ ПРОЦЕДУРЫ ВЫВОДА

ВЫВОД ЗНАЧЕНИЙ ДАННЫХ В СТРОКИ ЭКРАНА ДИСПЛЕЯ ЗАДАЕТ ПРЕДОПРЕДЕЛЕННЫЕ ПРОЦЕДУРЫ

WRITE (<СПИСОК ВЫВОДА>)

WRITELN [ (<СПИСОК ВЫВОДА>)]

<СПИСОК ВЫВОДА> ::= <ЭЛЕМЕНТ СПИСКА> , <ЭЛЕМЕНТ СПИСКА>

<ЭЛЕМЕНТ СПИСКА> ::=  $\alpha$  [ $\beta$  [ $\gamma$ ]]

ЗДЕСЬ  $\alpha$  - ВЫРАЖЕНИЕ ЛЮБОГО ПРЕДОПРЕДЕЛЕННОГО ТИПА ИЛИ ТИПА ДИАПАЗОН, БАЗОВЫМ ТИПОМ КОТОРОГО ЯВЛЯЕТСЯ ОДИН ИЗ ПРЕДОПРЕДЕЛЕННЫХ ТИПОВ (СМ. 2.4,  $\beta, \gamma$  - ВЫРАЖЕНИЯ ЦЕЛОГО ТИПА ( $0 \leq \gamma \leq 24$ , ИСПОЛЬЗУЕТСЯ ТОЛЬКО ДЛЯ  $\alpha$  ТИПА REAL).  $\beta$  ОПРЕДЕЛЯЕТ ШИРИНУ ПОЛЯ, В КОТОРОЕ ВЫДАЕТСЯ ЗНАЧЕНИЕ  $\alpha$ , ПРИЧЕМ ЗНАЧЕНИЕ  $\alpha$  В ПОЛЕ ПРИЖИМАЕТСЯ К ПРАВОМУ КРАЮ ПОЛЯ:  $\dots \dots < \text{ЗНАЧЕНИЕ } \alpha >$ . ОГРАНИЧЕНИЯ НА ВЕЛИЧИНУ  $\beta$  ПРАКТИЧЕСКИ НЕТ.

ЕСЛИ  $\alpha$  - ЧИСЛОВОЕ ВЫРАЖЕНИЕ, ЕГО ЗНАЧЕНИЕ ИЗ ВНУТРИМАШИННОГО ПРЕДСТАВЛЕНИЯ ПРЕОБРАЗУЕТСЯ В ДЕСЯТИЧНУЮ СИСТЕМУ СЧИСЛЕНИЯ И ВЫВОДИТСЯ В ВИДЕ СООТВЕТСТВУЮЩЕЙ ПОСЛЕДОВАТЕЛЬНОСТИ СИМВОЛОВ.

ТИП	ВНЕШНЕЕ ПРЕДСТАВЛЕНИЕ
INTEGER	$\{ 0$ $\{ 3c_n c_{n-1} \dots c_2 c_0$ , ЕСЛИ $\alpha = 0$ , $\{ 3c_n c_{n-1} \dots c_2 c_0$ , ЕСЛИ $\alpha \neq 0$ . ЗДЕСЬ } ::= ПУСТО! - - ЗНАК ЧИСЛА, $c_i$ - ДЕСЯТИЧНАЯ ЦИФРА,
REAL	$M S_1 P_1 P_2$ - ПРЕДСТАВЛЕНИЕ ЧИСЛА $M \cdot 10^{S_1 P_1 P_2}$ В ЭКСПОНЕНЦИАЛЬНОЙ ФОРМЕ. ЗДЕСЬ $M = c_1 c_0 . c_{-1} \dots c_{-10}$ . $S_1 ::= +   -$ - ЗНАК ЧИСЛА, $S_1 ::= +   -$ - ЗНАК ПОРЯДКА, $P_1 P_2$ - ДВУХРАЗЯДНОЕ ПРЕДСТАВЛЕНИЕ ПОРЯДКА:

! WRITE (PI) => 3.1415926536E+00 (ЧИСЛО ЗАНИМАЕТ ПОЛЕ ИЗ 18  
! ПОЗИЦИЙ).  
BOOLEAN ! FALSE!TRUE

РЕГУЛИРУЯ ШИРИНУ ПОЛЯ, ДАННОГО ПАРАМЕТРОМ  $\beta$  ДЛЯ  $\alpha$  ТИПА REAL, СЛЕДУЕТ ПОМНИТЬ, ЧТО  $u_0, u_{-1}, E, P, R_2$  ВЫДАЮТСЯ ВСЕГДА, ПОЭТОМУ  $\beta > \beta_{min}$ ,  $\beta_{min} := 7$  (ЕСЛИ  $\alpha \geq 0$ ) ! В (ЕСЛИ  $\alpha < 0$ ). ЕСЛИ  $\beta < \beta_{min}$ , ПРИНИМАЕТСЯ  $\beta = \beta_{min}$ . ИСПОЛЬЗОВАНИЕ ПАРАМЕТРА  $\gamma$  ДЛЯ  $\alpha$  ТИПА REAL ОПРЕДЕЛЯЕТ ВЫДАЧУ ЗНАЧЕНИЯ  $\alpha$  В ФОРМЕ С ФИКСИРОВАННОЙ ТОЧКОЙ С  $\gamma$  ЦИФРАМИ ПОСЛЕ ДЕСЯТИЧНОЙ ТОЧКИ (С ОКРУГЛЕНИЕМ):  $d, c_k, c_{k-1}, \dots, c_2, c_0, c_{-1}, \dots, c_{-\gamma}$ , ЕСЛИ  $\alpha \neq 0$  И 0, ЕСЛИ  $\alpha = 0$ . ЕСЛИ  $\beta$  НЕДОСТАТОЧЕН ДЛЯ ВЫДАЧИ ЦИФР ДРОБНОЙ ЧАСТИ, ПАРАМЕТР  $\beta$  ИГНОРИРУЕТСЯ (WRITE (PI/10:10:10) => 0.3141592654).

ВЫПОЛНЕНИЕ ПРОЦЕДУРЫ WRITE:

НА ЭКРАН, НАЧИНАЯ С ПОЗИЦИИ, ПОДСВЕЩЕННОЙ КУРСОРОМ, ВЫДАЮТСЯ ЗНАЧЕНИЯ ЭЛЕМЕНТОВ СПИСКА ВЫВОДА, ЗАПОЛНЯЯ ТЕКУЩУЮ СТРОКУ. ЕСЛИ СТРОКА ЗАПОЛНЕНА, ВЫДАЧА ПРОДОЛЖАЕТСЯ В СЛЕДУЮЩУЮ СТРОКУ ЭКРАНА С ЕЕ НАЧАЛА. ПОСЛЕ ВЫДАЧИ ПОСЛЕДНЕГО ЭЛЕМЕНТА СПИСКА ВЫПОЛНЕНИЕ ПРОЦЕДУРЫ ЗАВЕРШЕНО, КУРСОР ПОКАЗЫВАЕТ НА ОЧЕРЕДНУЮ СВОБОДНУЮ ПОЗИЦИЮ ЭКРАНА.

ВЫПОЛНЕНИЕ ПРОЦЕДУРЫ WRITELN ОТЛИЧАЕТСЯ ОТ ВЫПОЛНЕНИЯ ПРОЦЕДУРЫ WRITE ЛИШЬ ТЕМ, ЧТО ВЫДАЧА ЗАВЕРШАЕТСЯ ПЕРЕМЕЩЕНИЕМ КУРСОРА К НАЧАЛУ ОЧЕРЕДНОЙ СТРОКИ ЭКРАНА ДИСПЛЕЯ.

ЕСЛИ  $V := \text{FALSE}$ , ТО ПРОЦЕДУРЫ

WRITELN; WRITE (V:7, NOT V, PI); WRITELN (PI:10);  
WRITE (PI:10:4,5:3,5,-5,-5:3)

ЗАДАЕТ ВЫДАЧУ ПОСЛЕДОВАТЕЛЬНОСТЬЮ СИМВОЛОВ

...FALSETRUE...3.1415926536E+003.1416E+00

...3.1415...55-5,-5,

КУРСОР ОСТАЕТСЯ ЗА ПОСЛЕДНИМ ВЫДАННЫМ СИМВОЛОМ.

1.8-2. ПРЕДОПРЕДЕЛЕННЫЕ ПРОЦЕДУРЫ ВВОДА

ДЛЯ ВВОДА ЗНАЧЕНИЙ ДАННЫХ, ПРЕДСТАВЛЯЕМЫХ В ВИДЕ ПОСЛЕДОВАТЕЛЬНОСТЕЙ СИМВОЛОВ В СТРОКЕ ЭКРАНА ДИСПЛЕЯ, ИСПОЛЬЗУЮТСЯ ПРЕДОПРЕДЕЛЕННЫЕ ПРОЦЕДУРЫ

READ (<СПИСОК ВВОДА>),  
READLN [ (<СПИСОК ВВОДА>) ] .

ЗДЕСЬ <СПИСОК ВВОДА> := <ПЕРЕМЕННАЯ> { , <ПЕРЕМЕННАЯ> } . ПЕРЕМЕННАЯ ДОЛЖНА БЫТЬ ОДНОГО ИЗ ПРЕДОПРЕДЕЛЕННЫХ ТИПОВ (КРОМЕ ТИПА BOOLEAN) ИЛИ ТИПА ДИАПАЗОН, БАЗОВЫМ ТИПОМ КОТОРОГО ЯВЛЯЕТСЯ ПРЕДОПРЕДЕЛЕННЫЙ ТИП, ОТЛИЧНЫЙ ОТ ТИПА BOOLEAN.

ВЫПОЛНЕНИЕ ПРОЦЕДУР

А) ВЫПОЛНЕНИЕ ПРОГРАММЫ ПРИОСТАНАВЛИВАЕТСЯ ДО ТЕХ ПОР, ПОКА ПОЛЬЗОВАТЕЛЕМ БУДЕТ НАЖАТА КЛАВИША <ET>;

Б) ПОЛЬЗОВАТЕЛЬ ДОЛЖЕН НАБРАТЬ ЗНАЧЕНИЕ ЭЛЕМЕНТОВ СПИСКА ВВОДА В СООТВЕТСТВУЮЩЕЙ ПОСЛЕДОВАТЕЛЬНОСТИ, НАЧИНАЯ С МЕСТА ЭКРАНА, ПОДСВЕЩЕННОГО КУРСОРОМ, И НАЖАТЬ КЛАВИШУ <ET> (КОЛИЧЕСТВО НАБИРАЕМЫХ СИМВОЛОВ НЕ МОЖЕТ ПРЕВЫШАТЬ 127);

В) МАШИНА ЧИТАЕТ НАБРАННУЮ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ, ВЫБИРАЕТ В НЕЙ ПОДПОСЛЕДОВАТЕЛЬНОСТИ, СООТВЕТСТВУЮЩИЕ ЭЛЕМЕНТАМ СПИСКА, ДЛЯ ЧИСЛОВЫХ ЭЛЕМЕНТОВ СПИСКА ПРЕОБРАЗУЕТ СИМВОЛЬНЫЕ ЗНАЧЕНИЯ ДАННЫХ ВО ВНУТРИМАШИННОЕ ПРЕДСТАВЛЕНИЕ, ПРИСВАИВАЕТ ПРОЧИТАННЫЕ ЗНАЧЕНИЯ ПЕРЕМЕННЫМ - ЭЛЕМЕНТАМ СПИСКА. ЕСЛИ НАБРАНО ЗНАЧЕНИЕ БОЛЬШЕ, ЧЕМ ТРЕБУЕТ СПИСОК ВВОДА, ЛИШНИЕ ЗНАЧЕНИЯ

ИГНОРИРУЮТСЯ. ЕСЛИ КОЛИЧЕСТВО НАБРАННЫХ ЗНАЧЕНИЙ НЕДОСТАТОЧНО, ПОСЛЕДНИЕ ЭЛЕМЕНТЫ СПИСКА НЕ ПОЛУЧАЮТ ЗНАЧЕНИЙ (ИХ ЗНАЧЕНИЯ ПРИ ВЫПОЛНЕНИИ ПРОЦЕДУРЫ НЕ ИЗМЕНЯЮТСЯ).

ПРИ ВЫДЕЛЕНИИ ПОДПОСЛЕДОВАТЕЛЬНОСТИ СИМВОЛОВ, ХРАНЯЩЕЙ СИМВОЛЬНОЕ ПРЕДСТАВЛЕНИЕ ЧИСЛОВОГО ДАННОГО, ОЧЕРЕДНЫЕ ПРОБЕЛЫ ИГНОРИРУЮТСЯ, ПОДПОСЛЕДОВАТЕЛЬНОСТЬ НАЧИНАЕТСЯ С ПЕРВОГО СИМВОЛА, ОТЛИЧНОГО ОТ ПРОБЕЛА, НЕ МОЖЕТ БЫТЬ ДЛИННЕЕ 30 СИМВОЛОВ, ДОЛЖНА ИМЕТЬ ВИД ЧИСЛОВОЙ КОНСТАНТЫ, ЗАКАНЧИВАЕТСЯ ПРИ ОБНАРУЖЕНИИ ПРОБЕЛА ИЛИ КОНЦА ПОСЛЕДОВАТЕЛЬНОСТИ СИМВОЛОВ.

ПОСЛЕ ВЫПОЛНЕНИЯ ПРОЦЕДУРЫ READ КУРСОР ОСТАЕТСЯ ЗА ПОСЛЕДНИМ НАБРАННЫМ СИМВОЛОМ. ВЫПОЛНЕНИЕ ПРОЦЕДУРЫ READLN ЗАВЕРШАЕТСЯ ПЕРЕМЕЩЕНИЕМ КУРСОРА К НАЧАЛУ ОЧЕРЕДНОЙ СТРОКИ ДИСПЛЕЯ.

НАПРИМЕР, ЕСЛИ VAR X,Y,Z: REAL; K,L: INTEGER; READ(X,Y,L); READLN(Z,K) И ПОЛЬЗОВАТЕЛЬ НАБРАЛ С ТЕКУЩЕЙ ПОЗИЦИИ ЭКРАНА

1.5 -39E-5 4.07 <ET>, ТО ЭЛЕМЕНТЫ СПИСКА ПОЛУЧАЮТ ЗНАЧЕНИЯ X=1.5; Y=0.00039; L=407, КУРСОР ОСТАЕТСЯ ЗА ПОСЛЕДНИМ НАБРАННЫМ СИМВОЛОМ "7". ЕСЛИ ДЛЯ ПРОЦЕДУРЫ READLN(Z,K) НАБРАНА ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ -536 -18 <ET> С ТЕКУЩЕЙ ПОЗИЦИИ ЭКРАНА (ВСЛЕД ЗА СИМВОЛОМ "7"), ТО Z ПОЛУЧАЕТ ЗНАЧЕНИЕ -536.0, K - ЗНАЧЕНИЕ -18, КУРСОР ПЕРЕМЕЩАЕТСЯ К НАЧАЛУ СЛЕДУЮЩЕЙ СТРОКИ ЭКРАНА ДИСПЛЕЯ.

1.9. ПРИМЕР. СОСТАВИТЬ ПРОГРАММУ, КОТОРАЯ СРЕДИ ВСЕХ НАТУРАЛЬНЫХ ЧИСЕЛ, БОЛЬШИХ 10 И ПРЕДСТАВИМЫХ В МАШИНЕ ИЩЕТ ЧИСЛА, ЯВЛЯЮЩИЕСЯ "ПЕРЕВЕРТЫШАМИ" В ДЕСЯТИЧНОМ ПРЕДСТАВЛЕНИИ; ВЫДАЕТ ИХ ПО 10 ЧИСЕЛ В СТРОКЕ.

```
PROGRAM ANEAR_BACK;
(* ПОИСК ЧИСЕЛ-ПЕРЕВЕРТЫШЕЙ В СИСТЕМЕ СЧИСЛЕНИЯ
С ОСНОВАНИЕМ Q *)
CONST
  Q=10; (* ОСНОВАНИЕ СИСТЕМЫ СЧИСЛЕНИЯ *)
  K=10; (* ЧИСЛО ЧИСЕЛ В СТРОКЕ *)
  L=6; (* КОЛИЧЕСТВО ПОЗИЦИЙ ДЛЯ ЧИСЛА *)
  MAXINT1=1000;
VAR I, I1, I2: INTEGER; (* ОЧЕРЕДНОЕ ЧИСЛО И ЕГО ЧАСТИ *)
    M, M1 : INTEGER; (* СТЕПЕНИ Q *)
    B : BOOLEAN;
    N : INTEGER; (* СЧЕТЧИК КОЛИЧЕСТВА ЧИСЕЛ В СТРОКЕ ЭКРАНА *)
    P, NP : INTEGER; (* КОЛИЧЕСТВО ПАР ЦИФР В ЧИСЛЕ, НОМЕР ПАРЫ *)
    ML, KP : INTEGER; (* ДЛИНА MAXINT1 И I В Q-ИЧНОЙ ЗАПИСИ *)
BEGIN
  (* ОПРЕДЕЛЕНИЕ СТАРШЕГО РАЗРЯДА И КОЛИЧЕСТВА ЦИФР
  В Q-ИЧНОЙ ЗАПИСИ MAXINT1 *)
  M:=1; ML:=1; WRITELN(M, MAXINT1, MAXINT1);
  WHILE (MAXINT1 DIV M) > 0 DO
    BEGIN M:=M*Q; ML:=ML+1 END;
  (* ПОИСК И ВЫДАЧА ПЕРЕВЕРТЫШЕЙ *)
  N:=0;
  FOR I:=Q TO MAXINT1 DO
    BEGIN
      (* ВЫДЕЛЕНИЕ СТАРШЕГО РАЗРЯДА И КОЛИЧЕСТВА
      РАЗРЯДОВ В Q-ИЧНОЙ ЗАПИСИ I *)
      M1:=M; KP:=ML;
      WHILE (I DIV M1) = 0 DO
```

```

BEGIN
  M1:=M1 DIV Q; KP:=KP-1
  END;
P:=KP DIV 2; (* КОЛИЧЕСТВО ПАР ЦИФР В I *)
(* ПРОВЕРКА ЧИСЛА НА ПЕРЕВЕРТЫШ *)
B:=TRUE; (* НЕ СОВПАВШИХ ПАР НЕ БЫЛО *)
I1:=I; I2:=I;
FOR NP:=1 TO P DO
  BEGIN
    IF (I1 MOD Q) <> (I2 DIV M1) THEN B:=FALSE;
    (* ЦИФРЫ В ПАРЕ НЕ СОВПАДАЮТ *)
    I1:=I1 DIV Q; I2:=I2 MOD Q; M1:=M1 DIV Q
    END; (* ПАРЫ *)
  IF B THEN (* ВЫДАЧА ПЕРЕВЕРТЫША *)
    BEGIN
      IF N>=K THEN BEGIN WRITELN; N:=0 END;
      N:=N+1; WRITE(I:L)
      END (* ВЫДАЧИ ПЕРЕВЕРТЫША *)
    END; (* ОБРАБОТКИ I *)
  WRITELN
END. (* AHEAD-BACK *)

```

### 1.10. ЗАДАНИЯ ПО ТЕМЕ

1. КОЗА ПРИВЯЗАНА К СТОЛБУ, РАСПОЛОЖЕННОМУ В ТОЧКЕ ОКРУЖНОСТИ РАДИУСА R, ОХВАТЫВАЮЩЕЙ ЛУГ. КАКОЙ ДОЛЖНА БЫТЬ ДЛИНА ВЕРЕВКИ, ЧТОБЫ ДЛЯ ВЫПАСА ИСПОЛЬЗОВАЛАСЬ ПОЛОВИНА ЛУГА?

2. РАЗЛОЖИТЬ ЧИСЛО НА ПРОСТЫЕ МНОЖИТЕЛИ — ЗНАЧИТ НАЙТИ ВСЕ ЕГО ДЕЛИТЕЛИ, ЯВЛЯЮЩИЕСЯ ПРОСТЫМИ ЧИСЛАМИ (НАПРИМЕР,  $792=2 \cdot 2 \cdot 2 \cdot 3 \cdot 3 \cdot 11$ ). СОСТАВИТЬ ПРОГРАММУ, КОТОРАЯ ДЛЯ КАЖДОГО ЦЕЛОГО ПОЛОЖИТЕЛЬНОГО ЧИСЛА ИЗ НЕКОТОРОЙ ПОСЛЕДОВАТЕЛЬНОСТИ НАХОДИТ ЕГО РАЗЛОЖЕНИЕ НА ПРОСТЫЕ МНОЖИТЕЛИ.

3. НАИМЕНЬШЕЕ ОБЩЕЕ КРАТНОЕ ЦЕЛЫХ ПОЛОЖИТЕЛЬНЫХ ЧИСЕЛ РАВНО ИХ ПРОИЗВЕДЕНИЮ, ДЕЛЕННОМУ НА НАИБОЛЬШИЙ ОБЩИЙ ДЕЛИТЕЛЬ. СОСТАВИТЬ ПРОГРАММУ, ОПРЕДЕЛЯЮЩУЮ НАИМЕНЬШЕЕ ОБЩЕЕ КРАТНОЕ НЕКОТОРОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ЧИСЕЛ.

4. ЦЕЛОЕ ПОЛОЖИТЕЛЬНОЕ ЧИСЛО НАЗЫВАЮТ СОВЕРШЕННЫМ, ЕСЛИ ОНО РАВНО СУММЕ ВСЕХ СВОИХ ДЕЛИТЕЛЕЙ, ВКЛЮЧАЮЩИХ 1, НО НЕ ВКЛЮЧАЮЩИХ САМО ЧИСЛО (НАПРИМЕР,  $6=1+2+3$ ;  $496=1+2+4+8+16+31+62+124+248$ ). СОСТАВИТЬ ПРОГРАММУ ВЫДАЧИ ВСЕХ СОВЕРШЕННЫХ ЧИСЕЛ, НЕ ПРЕВЫШАЮЩИХ ЧИСЛА N.

5. ДАТА ЗАДАЕТСЯ В ВИДЕ: ДЕНЬ, НОМЕР МЕСЯЦА, ГОД. ПО ТЕКУЩЕЙ ДАТЕ И ПО ДАТАМ РОЖДЕНИЯ ГРУППЫ ЛИЦ ОПРЕДЕЛИТЬ ДЛЯ КАЖДОГО ЧИСЛО ПРОЖИТЫХ ДНЕЙ.

6. СОСТАВИТЬ ПРОГРАММУ, КОТОРАЯ С ТОЧНОСТЬЮ  $\epsilon$  НАХОДИТ КОРНИ УРАВНЕНИЯ  $f(x)=0$ , РАСПОЛОЖЕННЫЕ НА  $[a, b]$  И УДАЛЕННЫЕ ДРУГ ОТ ДРУГА НЕ БЛИЖЕ, ЧЕМ НА  $h$ . ДЛЯ РЕШЕНИЯ ЗАДАЧИ СРЕДИ ОТРЕЗКОВ ДЛИНЫ  $h$ , НА КОТОРЫЕ РАЗБИВАЕТСЯ  $[a, b]$ , ВЫДЕЛИТЬ ТЕ, НА КОНЦАХ КОТОРЫХ  $f(x)$  ЗНАКОПРОТИВОПОЛОЖНА. НА КАЖДОМ ИЗ НИХ УТОЧНИТЬ КОРЕНЬ МЕТОДОМ ДЕЛЕНИЯ ОТРЕЗКА ПОПОЛАМ. ВЗЯТЬ  $f(x)=e^x+x^2-1,2$ .

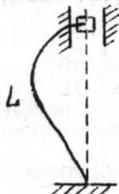
7. СОСТАВИТЬ ПРОГРАММУ, КОТОРАЯ ДЛЯ КАЖДОГО ИЗ  $k=1, 2, \dots, M$  С ТОЧНОСТЬЮ  $\epsilon$  НАХОДИТ СУММУ РЯДОВ (1) И (2), ВЫДАЕТ ЗНАЧЕНИЯ СУММ И КОЛИЧЕСТВА СЛАГАЕМЫХ, УЧАСТВУЮЩИХ В СУММИРОВАНИИ:

$$(1) \quad f(a) = \sum_{n=1}^{\infty} \frac{n+1}{n(a+n^2)} \quad (a = 0,05 \cdot k);$$

$$(2) G(a) = S(a) - \sum_{n=2}^{\infty} \frac{1}{n(n-1)} + \sum_{n=2}^{\infty} \frac{1}{n(n-1)} = \frac{2}{1+a} + 1 +$$

$$\sum_{n=2}^{\infty} \left( \frac{n+1}{n(n^2+a)} - \frac{1}{n(n-1)} \right) = \frac{3+a}{1+a} - (1+a) \cdot \sum_{n=2}^{\infty} \frac{1}{n(n-1)(n^2+a)}$$

В. КРИТИЧЕСКАЯ СИЛА Р ДЛЯ СТЕРЖНЯ, ОДИН КОНЕЦ КОТОРОГО ЗАЩЕМЛЕН, А ДРУГОЙ МОЖЕТ ПЕРЕМЕЩАТЬСЯ В ВЕРТИКАЛЬНОМ ПОЛОЖЕНИИ



УДОВЛЕТВОРЯЕТ УРАВНЕНИЮ (1)  $\operatorname{tg}(L \cdot \sqrt{\frac{P}{E}}) = L \cdot \sqrt{\frac{P}{E}}$

(L - ДЛИНА СТЕРЖНЯ, E - ИЗГИБАЮЩАЯ ЖЕСТКОСТЬ). ЕСЛИ  $x = L \cdot \sqrt{\frac{P}{E}}$ , ТОГДА УРАВНЕНИЕ (1) ПРИНИМАЕТ ВИД (2)  $\operatorname{tg} x - x = 0$ .

СОСТАВИТЬ ПРОГРАММУ ПОИСКА С ТОЧНОСТЬЮ  $\epsilon$  НУЖНОГО КОЛИЧЕСТВА КОРНЕЙ УРАВНЕНИЯ (2) МЕТОДОМ ХОРД (ПРОПОРЦИОНАЛЬНЫХ ЧАСТЕЙ).

МЕТОД ХОРД УТОЧНЕНИЯ КОРНЯ УРАВНЕНИЯ  $f(x) = 0$ , РАСПОЛОЖЕННОГО НА  $[a, b]$ : СТРОИТСЯ ПОСЛЕДОВАТЕЛЬНОСТЬ  $x_0, x_1, x_2, \dots$ , СХОДЯЩАЯСЯ К КОРНЮ УРАВНЕНИЯ, ПРИЧЕМ  $x_0 = a$ ;

$$x_{i+1} = x_i - \frac{f(x_i)}{f(b) - f(x_i)} \cdot (b - x_i).$$

ЕСЛИ  $|x_i - x_{i-1}| < \epsilon$ , СЧИТАЕТСЯ, ЧТО  $x_i$  - НУЖНОЕ ПРИБЛИЖЕНИЕ К РЕШЕНИЮ УРАВНЕНИЯ.

2. ЗАДАЧА КОШИ (1)  $y' = f(x, y)$ ,  $y(a) = y_0$  НА  $[a, b]$  МОЖЕТ БЫТЬ РЕШЕНА МЕТОДОМ ЭЙЛЕРА: ИЩЕТСЯ ТАБЛИЦА ПРИБЛИЖЕННОГО РЕШЕНИЯ  $y = y(x)$  В ТОЧКАХ  $x_0 = a$ ,  $x_{i+1} = x_i + h$ , ГДЕ

$$h = (b-a)/N, \quad y(x_0) = y_0, \quad y(x_i) = y_i, \quad y_{i+1} = y_i + h \cdot f(x_i, y_i),$$

$i = 0, 1, \dots, N-1$ .

СОСТАВИТЬ ПРОГРАММУ, КОТОРАЯ РЕШАЕТ ЗАДАЧУ КОШИ

$$y' = x \cdot \sin y + e^{-(x+y)}, \quad y(a) = y_0$$

ДЛЯ КАЖДОГО  $s = 0, 0.5, 1, \dots, M$  НА  $[a, b] = [0, 1]$  МЕТОДОМ ЭЙЛЕРА С ШАГОМ  $h = (b-a)/(10 \cdot L)$  И ВЫДАЕТ НА ЭКРАН ЗНАЧЕНИЯ С ШАГОМ  $h_s = (b-a)/L$  ( $M, L, a, b, L$  ВВЕСТИ).

19. МЕТОД НЬЮТОНА НАХОЖДЕНИЯ С ТОЧНОСТЬЮ  $\epsilon$  РЕШЕНИЯ СИСТЕМЫ

$$\begin{cases} F(x, y) = 0, \\ G(x, y) = 0 \end{cases} \quad (1)$$

ЗАКЛЮЧАЕТСЯ В ПОСТРОЕНИИ ПОСЛЕДОВАТЕЛЬНОСТИ  $(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots$  СХОДЯЩЕЙСЯ К РЕШЕНИЮ СИСТЕМЫ. ОБЫЧНО СЧИТАЮТ  $(x_n, y_n)$  ИСХОДНОЙ ТОЧКОЙ, ЕСЛИ РАССТОЯНИЕ ОТ НЕЕ ДО ТОЧКИ  $(x_{n-1}, y_{n-1})$  НЕ ПРЕВЫШАЕТ  $\epsilon$ . ПРИЕМ  $x_{j+1} = x_j - \Delta x_j / \Delta_j$ ,  $y_{j+1} = y_j - \Delta y_j / \Delta_j$ .

$$\text{ГДЕ } \Delta_x = \begin{vmatrix} F'_x & F'_y \\ G'_x & G'_y \end{vmatrix}, \quad \Delta_{x_0} = \begin{vmatrix} F & F'_y \\ G & G'_y \end{vmatrix}, \quad \Delta_{y_0} = \begin{vmatrix} F'_x & F \\ G'_x & G \end{vmatrix}$$

ПРИ  $x = x_i, y = y_i$ .

СОСТАВИТЬ ПРОГРАММУ, КОТОРАЯ МЕТОДОМ НЬЮТОНА НАХОДИТ РЕШЕНИЕ СИСТЕМЫ

$$\begin{cases} ch(0,3x^2+1) - x + y = 1, \\ (x-0,1)^2/0,21 - y^2/2 = 0,8. \end{cases} \quad (2)$$

РАСПОЛОЖЕННОЕ В ПЕРВОЙ КООРДИНАТНОЙ ЧЕТВЕРТИ.

ДЛЯ ОПРЕДЕЛЕНИЯ  $(x_0, y_0)$  РАЗРЕШИТЬ  $G(x, y) = 0$  ОТНОСИТЕЛЬНО  $y$ :  $y = \psi(x)$  (3). НАЙТИ ВЕРШИНУ ГИПЕРБОЛЫ (3). ДВИГАЯСЯ ПО ТОЧКАМ ГИПЕРБОЛЫ (3) ОТ ВЕРШИНЫ ВПРАВО С ШАГОМ  $h$  ПО ОСИ  $Ox$ , НАЙТИ ДВЕ ТАКИХ ТОЧКИ  $(x, \psi(x))$  И  $(\bar{x}, \psi(\bar{x}))$ , ЧТО ФУНКЦИЯ  $F(x, y)$  В НИХ ЗНАКОПРОТИВОПОЛОЖНА. ВЗЯТЬ  $x_0 = \bar{x} + h/2, y_0 = \psi(x_0)$  ( $\bar{x} = \bar{x} + h$ ).

11. ПРИБЛИЖЕННОЕ ЗНАЧЕНИЕ  $I(c) = \int_a^b f(x, c) dx$  (1)

ПРИ ФИКСИРОВАННОМ  $c$  МОЖЕТ БЫТЬ НАЙДЕНО МЕТОДОМ СИМПСОНА

(ПАРАБОЛ): 
$$I = \int_a^b \psi(x) dx \approx (b-a)/(3N) \cdot (\psi(x_0) + 4\psi(x_1) + 2\psi(x_2) + 4\psi(x_3) +$$

(2) 
$$+ 2\psi(x_4) + 4\psi(x_5) + \dots + 2\psi(x_{N-2}) + 4\psi(x_{N-1}) + \psi(x_N)).$$

ЗДЕСЬ  $x_0 = a$ ,  $h = (b-a)/N$  ( $N$  - ЧЕТНОЕ),  $x_{i+1} = x_i + h, i = 0, 1, \dots, N-1$ .

СОСТАВИТЬ ПРОГРАММУ, КОТОРАЯ ДЛЯ ЗАДАНЫХ  $c = 0,05 \cdot K, K = 0, 1, \dots, M$  НАХОДИТ ПРИБЛИЖЕННЫЕ ЗНАЧЕНИЯ  $I(c)$  С ТОЧНОСТЬЮ  $\epsilon$ . ДЛЯ ПОЛУЧЕНИЯ ЗНАЧЕНИЯ  $I(c)$  С ТОЧНОСТЬЮ  $\epsilon$  РАЗБИВАЮТ  $[a, b]$  НА 2 ЧАСТИ ( $N=2$ ), ВЫЧИСЛЯЮТ ПРИБЛИЖЕННОЕ ЗНАЧЕНИЕ  $I(c) = I_2$ , ВЫЧИСЛЯЮТ  $I(c)$  ПРИ  $N$  В ДВА РАЗА БОЛЬШЕМ ДО ТЕХ ПОР, ПОКА НОВОЕ ПРИБЛИЖЕННОЕ ЗНАЧЕНИЕ  $I(c)$  ОТЛИЧАЕТСЯ ОТ ПРЕДЫДУЩЕГО ПО АБСОЛЮТНОЙ ВЕЛИЧИНЕ БОЛЬШЕ, ЧЕМ НА  $\epsilon$ .

ВЗЯТЬ 
$$I(c) = \int_a^b \cos(e^{1/3} + cx) dx.$$

12. МЕТОД ИТЕРАЦИИ РЕШЕНИЯ УРАВНЕНИЯ (1)  $\varphi(x) = x$  ЗАКЛЮЧАЕТСЯ В ПОСТРОЕНИЕ ПОСЛЕДОВАТЕЛЬНЫХ ПРИБЛИЖЕНИЙ К РЕШЕНИЮ

(2)  $x_0, x_1, x_2, \dots, x_{M-1}, x_M, \dots$  ДО ТЕХ ПОР, ПОКА  $|x_M - x_{M-1}| > \delta$  ПО ФОРМУЛЕ  $x_{M+1} = \varphi(x_M)$ . МЕТОД ОЧЕНЬ ПРОСТ, НО НЕ ВСЕГДА ОБЕСПЕЧИВАЕТ СХОДИМОСТЬ ПОСЛЕДОВАТЕЛЬНОСТИ (2), ПОЭТОМУ ПРИ ЕГО ИСПОЛЬЗОВАНИИ СЛЕДУЕТ ПРЕДУСМАТРИВАТЬ КОНТРОЛЬ СХОДИМОСТИ И, ЕСЛИ ЗА УКАЗАННОЕ ЧИСЛО ШАГОВ СХОДИМОСТЬ НЕ ОБЕСПЕЧИВАЕТСЯ, ПРЕКРАЩАТЬ ПРОЦЕСС ПОСТРОЕНИЯ ПРИБЛИЖЕНИЙ.

СОСТАВИТЬ ПРОГРАММУ, КОТОРАЯ ДЛЯ КАЖДОГО  $c = 0,1K, K = 0, 1, \dots, L$  НАХОДИТ С ТОЧНОСТЬЮ  $\epsilon$  КОРЕНЬ УРАВНЕНИЯ  $4x - ch(x(1+c)) = 0$  МЕТОДОМ ИТЕРАЦИИ. ЗНАЧЕНИЕ  $x_0$  ОПРЕДЕЛИТЬ ГРАФИЧЕСКИ.

## 2. РЯД ДОПОЛНИТЕЛЬНЫХ ВОЗМОЖНОСТЕЙ

### 2.1. ЛИТЕРНЫЙ ТИП (СНАР)

ЛИТЕРНЫЙ ТИП (СНАР) ЯВЛЯЕТСЯ ОДНИМ ИЗ ПРЕДОПРЕДЕЛЕННЫХ ТИПОВ ПАСКАЛЯ. ЗНАЧЕНИЕ ДАННОГО ЭТОГО ТИПА - ЛЮБАЯ ДОПУСТИМАЯ НА ПЭВМ

ЛИТЕРА, ПРЕДСТАВЛЕННАЯ В КОДЕ КОИ-7 В ОДНОМ БАЙТЕ ПАМЯТИ. КОДЫ КОИ-7 ЗАГЛАВНЫХ ЛАТИНСКИХ БУКВ (ЦИФР) УПОРЯДОЧЕНЫ В СООТВЕТСТВИИ С АЛФАВИТОМ (ЗНАЧЕНИЕМ ЦИФР) И ОБРАЗУЮТ СВЯЗНОЕ МНОЖЕСТВО.

СРЕДИ ЛИТЕР РАЗЛИЧАЮТ ПРЕДСТАВИМЫЕ ЛИТЕРЫ, КОТОРЫЕ ИМЕЮТ ГРАФИЧЕСКОЕ ИЗОБРАЖЕНИЕ (ИХ КОДЫ КОИ-7 ОТ 32 ДО 126), И ЛИТЕРЫ, НЕ ИМЕЮЩИЕ ГРАФИЧЕСКОГО ИЗОБРАЖЕНИЯ (ИХ КОДЫ КОИ-7 ОТ 0 ДО 31). ЭТИ ЛИТЕРЫ НАЗЫВАЮТ УПРАВЛЯЮЩИМИ.

КОНСТАНТА ТИПА CHAR МОЖЕТ БЫТЬ ЗАДАНА КОДОМ КОИ-7 ЛИТЕРЫ С ПРЕДШЕСТВУЮЩИМ СИМВОЛОМ "#": #65, #841 - ЛИТЕРА A; #830 - ЛИТЕРА O. ДЛЯ ПРЕДСТАВИМОЙ ЛИТЕРЫ КОНСТАНТА МОЖЕТ ИМЕТЬ ВИД ЭТОЙ ЛИТЕРЫ (В СЛУЧАЕ ЛИТЕРЫ - АПОСТРОФА ВИД ДВУХ СОСЕДНИХ АПОСТРОФОВ) ЗАКЛЮЧЕННОЙ В АПОСТРОФЫ 'A', 'C', 'Z', ';', '...' ДЛЯ УПРАВЛЯЮЩЕЙ ЛИТЕРЫ КОНСТАНТУ ДОПУСКАЕТСЯ ЗАДАВАТЬ КОМБИНАЦИЕЙ ЛИТЕРЫ "^" И ПРЕДСТАВИМОЙ ЛИТЕРЫ С КОИ-7 НА 64 БОЛЬШЕ, ЧЕМ КОИ-7 УПРАВЛЯЮЩЕЙ ЛИТЕРЫ: #12 И ^L ОПРЕДЕЛЯЮТ КОНСТАНТУ, ЗНАЧЕНИЕМ КОТОРОЙ ЯВЛЯЕТСЯ УПРАВЛЯЮЩАЯ ЛИТЕРА С КОИ-7, РАВНЫМ 12.

ДААННЫЕ ЛИТЕРНОГО ТИПА УПОРЯДОЧЕНЫ В СООТВЕТСТВИИ С ИХ КОДАМИ КОИ-7: #12<'A', 'A'> 'C'.

ДЛЯ ДАННЫХ ТИПА CHAR ОПРЕДЕЛЕНА ОПЕРАЦИЯ ОТНОШЕНИЯ С БУЛЕВСКИМ РЕЗУЛЬТАТОМ. ТИП CHAR ЯВЛЯЕТСЯ ОРДИНАЛЬНЫМ: ORD(X) ДЛЯ X ТИПА CHAR - КОД КОИ-7 ЛИТЕРЫ X; PRED(X) - ЛИТЕРА, КОД КОИ-7 КОТОРОЙ ПРЕДШЕСТВУЕТ КОДУ КОИ-7 ЛИТЕРЫ X; SUCC(X) - ЛИТЕРА, КОД КОИ-7 КОТОРОЙ СЛЕДУЕТ ЗА КОДОМ КОИ-7 ЛИТЕРЫ X.

ПРЕДОПРЕДЕЛЕННАЯ ФУНКЦИЯ CHR(I) ДЛЯ ЦЕЛОГО АРГУМЕНТА ИЗ [0,127] ИМЕЕТ ЗНАЧЕНИЕМ ЛИТЕРУ С КОДОМ КОИ-7, РАВНЫМ I.

ПЕРЕМЕННОЙ ТИПА CHR МОЖНО ПРИСВОИТЬ ЗНАЧЕНИЕ ВЫРАЖЕНИЯ ТИПА CHAR, ОНА МОЖЕТ БЫТЬ ПАРАМЕТРОМ ЦИКЛА В ОПЕРАТОРЕ FOR.

ДААННОЕ ТИПА CHAR МОЖЕТ БЫТЬ ОПРАНДОМ ВЫРАЖЕНИЯ. ЕСЛИ ЭЛЕМЕНТ СПИСКА ВВОДА ПРОЦЕДУРЫ WRITE/WRITELN ЯВЛЯЕТСЯ ВЫРАЖЕНИЕМ ТИПА CHAR, ВЫВОДУ ПОДЛЕЖИТ ЛИТЕРА - ЗНАЧЕНИЕ ВЫРАЖЕНИЯ (WRITE('A', '0':4) --> A0000). ПОПЫТКА ВЫДАТЬ НА ЭКРАН УПРАВЛЯЮЩУЮ ЛИТЕРУ ВЫЗЫВАЕТ ВЫПОЛНЕНИЕ НЕКОТОРЫХ ВСПОМОГАТЕЛЬНЫХ ДЕЙСТВИЙ. НАПРИМЕР, ВЫПОЛНЕНИЕ ПРОЦЕДУРЫ WRITE(^L) ИЛИ WRITELN(#12) ВЫЗЫВАЕТ ОЧИСТКУ ЭКРАНА ДИСПЛЕЯ С ПОДВОДОМ КУРСОРА В ЛЕВЫЙ ВЕРХНИЙ УГОЛ ЭКРАНА.

ЕСЛИ ПЕРЕМЕННАЯ ТИПА CHAR - ЭЛЕМЕНТ СПИСКА ВВОДА ПРОЦЕДУРЫ READ/READLN, ТО ЕЕ ЗНАЧЕНИЕМ ЯВЛЯЕТСЯ ОЧЕРЕДНАЯ ЛИТЕРА ПОСЛЕДОВАТЕЛЬНОСТИ ЛИТЕР, НАБРАННОЙ ПРИ ВВОДЕ. ВКЛЮЧАЯ В СПИСОК ВВОДА ЭЛЕМЕНТЫ РАЗЛИЧНЫХ ТИПОВ, СЛЕДУЕТ ПОМНИТЬ, ЧТО ПРИ ЧТЕНИИ ЧИСЛА ВЫБИРАЮТСЯ ОЧЕРЕДНЫЕ ЛИТЕРЫ ВВОДИМОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ЛИТЕР С ПРОПУСКОМ ЛЕВЫХ ПРОБЕЛОВ, ЕСЛИ ОНИ ЕСТЬ, ДО ЛИТЕРЫ ПРОБЕЛ ИЛИ НЕДОПУСТИМОЙ ЛИТЕРЫ. ЗНАЧЕНИЕ ПЕРЕМЕННОЙ ТИПА CHAR ВО ВВОДИМОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ДОЛЖНО СЛЕДОВАТЬ НЕПОСРЕДСТВЕННО ЗА ПОСЛЕДНЕЙ ЛИТЕРОЙ ЧИСЛА. ДЛЯ НЕКОТОРЫХ ДАННЫХ ТИПА CHAR ЭТО ВЫЗЫВАЕТ ОШИБКУ ПРИ ВЫПОЛНЕНИИ ВВОДА. НАПРИМЕР, VAR X, Y:REAL; S:CHAR; ... READ(X,Y,S); ЧТЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ЛИТЕР -1.1 2 <E> ДАЕТ X=-1.1, Y=2, S=' '; ЧТЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ЛИТЕР -1.1 2\*<E> ДАЕТ X=-1.1, Y=2, S='\*'; ЧТЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ЛИТЕР -1.1 2<E> ВЫЗЫВАЕТ ОШИБКУ И СНЯТИЕ ЗАДАЧИ. ПОЭТОМУ ЛИТЕРНЫЕ ЭЛЕМЕНТЫ СПИСКА ВВОДА СЛЕДУЕТ ПРЕДПОСЛАТЬ ЧИСЛОВЫМ ЭЛЕМЕНТАМ: READ(S,X,Y). ТОГДА ЧТЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ЛИТЕР + -1.1 2 ДАЕТ S='+', X=-1.1, Y=2.

ФРАГМЕНТ ПРОГРАММЫ

```
VAR C,S: CHAR; SN: INTEGER; B,B1: BOOLEAN;
```

```

READ(S); B:=FALSE; B1:=FALSE;
IF(S>='0') AND (S<='9') THEN B:=TRUE;
FOR C:='A' TO 'Z' DO IF S=C THEN B1:=TRUE;
WRITELN(S:2; B:7, B1:7);
ВВОДИТ ЛИТЕРУ S; ПРИСВАИВАЕТ В ЗНАЧЕНИЕ TRUE, ЕСЛИ S - ДЕСЯТИЧНАЯ
ЦИФРА; ПРИСВАИВАЕТ B1 ЗНАЧЕНИЕ TRUE, ЕСЛИ S - ЗАГЛАВНАЯ ЛАТИНСКАЯ
БУКВА (ТОТ ЖЕ РЕЗУЛЬТАТ БЫЛ БЫ ПОЛУЧЕН ПРИ ВЫПОЛНЕНИИ ОПЕРАТОРОВ
S1:=ORD(S); IF(ORD('A')<=S1) AND (ORD('Z')>=S1) THEN B1:=TRUE ИЛИ
IF ('A'<=S) AND ('Z'>=S) THEN B1:=TRUE); ВЫВОДИТ S,B,B1.

```

## 2.2. СТРОКОВЫЙ ТИП (STRING[*n*])

В ТУРЕО ПАСКАЛЕ СТРОКОВЫЙ ТИП ЯВЛЯЕТСЯ ПРЕДОПРЕДЕЛЕННЫМ. ЗНАЧЕНИЕ ДАННОГО ЭТОГО ТИПА - ПОСЛЕДОВАТЕЛЬНОСТЬ НЕ БОЛЕЕ *n* (*n* ∈ {0,255}) ЛЮБЫХ ДОПУСТИМЫХ НА МАШИНЕ ЛИТЕР (В ТОМ ЧИСЛЕ ПУСТАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ), ДЛЯ ХРАНЕНИЯ СТРОКОВОГО ЗНАЧЕНИЯ ИСПОЛЬЗУЕТСЯ ПОЛЕ ИЗ *n* +1 ПОСЛЕДОВАТЕЛЬНЫХ БАЙТОВ ПАМЯТИ, ЧУМЕРУЕМЫХ ОТ 0 ДО *n*. В НУЛЕВОМ БАЙТЕ ПОЛЯ РАЗМЕЩАЕТСЯ ТЕКУЩАЯ ДЛИНА ЗНАЧЕНИЯ, НАЧИНАЯ С ПЕРВОГО БАЙТА - ЛИТЕРЫ ЗНАЧЕНИЯ, ПРЕДСТАВЛЕННЫЕ В КОДЕ КОИ-7.

В ОТЛИЧИЕ ОТ ДРУГИХ ПРЕДОПРЕДЕЛЕННЫХ ТИПОВ СТРОКОВЫЙ ТИП ПОДЛЕЖИТ ОПРЕДЕЛЕНИЮ, ТАК КАК ЗНАЧЕНИЕ *n* НЕ ПРЕДОПРЕДЕЛЕНО. ОПРЕДЕЛЕНИЕ ТИПА - УКАЗАНИЕ ВЕРХНЕЙ ГРАНИЦЫ ДЛИН ПОСЛЕДОВАТЕЛЬНОСТЕЙ ЛИТЕР - ЗНАЧЕНИЯ ДАННЫХ ТИПА:

```
STRING[n],
```

ГДЕ *n* - ЦЕЛАЯ КОНСТАНТА ИЛИ ИДЕНТИФИКАТОР ЦЕЛОЙ КОНСТАНТЫ И 1 <= *n* <= 255.

ОПРЕДЕЛЕНИЕ СТРОКОВОГО ТИПА МОЖЕТ БЫТЬ ПРОИЗВЕДЕНО В РАЗДЕЛЕ ТУРЕ. ТОГДА ОПРЕДЕЛЯЕМЫЙ ТИП ПОЛУЧАЕТ ИМЯ (<ИМЯ ТИПА>=<ТИП>) И ПРИ ОПИСАНИИ ПЕРЕМЕННЫХ ЭТОГО ТИПА В РАЗДЕЛЕ VAR В КАЧЕСТВЕ ТИПА ДОСТАТОЧНО УКАЗАТЬ ИМЯ СТРОКОВОГО ТИПА. НАПРИМЕР,

```
TYPE ST=STRING[10];
```

```
VAR S,S1:ST; (*ОПИСАНЫ ПЕРЕМЕННЫЕ S,S1 СТРОКОВОГО ТИПА*)
```

ЯЗЫК ПАСКАЛЬ ДОПУСКАЕТ ИСПОЛЬЗОВАНИЕ АНОНИМНЫХ ТИПОВ.

ТИП СЧИТАЕТСЯ АНОНИМНЫМ (НЕ ИМЕЕТ ИМЕНИ), ЕСЛИ ОПРЕДЕЛЕНИЕ ТИПА УКАЗЫВАЕТСЯ В КАЧЕСТВЕ ТИПА ПРИ ОПИСАНИИ ПЕРЕМЕННЫХ В РАЗДЕЛЕ VAR. НАПРИМЕР, VAR C,C1:STRING[15]; (\*ОПИСАНЫ ПЕРЕМЕННЫЕ C,C1 СТРОКОВОГО ТИПА С ДЛИНАМИ ЗНАЧЕНИЯ, НЕ БОЛЬШИМИ 15. ТИП STRING[15] НЕ ИМЕЕТ ИМЕНИ \*)

СУЩЕСТВУЮТ СИТУАЦИИ, КОГДА ТИП ДОЛЖЕН БЫТЬ ИМЕНОВАННЫМ (НАПРИМЕР, ТИП ФОРМАЛЬНОГО ПАРАМЕТРА ПОДПРОГРАММЫ).

СТРОКОВАЯ КОНСТАНТА - ПОСЛЕДОВАТЕЛЬНОСТЬ ОТ 0 ДО 255 ПРЕДСТАВИМЫХ ЛИТЕР, ЗАКЛЮЧЕННАЯ В АПОСТРОФЫ (ЛИТЕРА АПОСТРОФ ("") ВКЛЮЧАЕТСЯ В ПОСЛЕДОВАТЕЛЬНОСТЬ В ВИДЕ ДВУХ СОСЕДНИХ АПОСТРОФОВ). НАПРИМЕР, 'КОНСТАНТА СТРОКОВОГО ТИПА' - СТРОКА ДЛИНЫ 25; 'Y'=F(X,Y)' - СТРОКА ДЛИНЫ 9; '' - СТРОКА ДЛИНЫ 0 (ПУСТАЯ СТРОКА).

УПРАВЛЯЮЩИЕ ЛИТЕРЫ ВКЛЮЧАЮТСЯ В СТРОКОВУЮ КОНСТАНТУ БЕЗ ОКРАИМЛЯЮЩИХ АПОСТРОФОВ: #12#10^B. ДОПУСТИМО СМЕШИВАНИЕ В СТРОКОВОЙ КОНСТАНТЕ ПРЕДСТАВИМЫХ И УПРАВЛЯЮЩИХ ЛИТЕР: #12#10^B^ВНИМАНИЕ! ПРИГОТОВИТЬСЯ К ВВОДУ ДАННЫХ'. ПРИ ВЫДАЧЕ НА ЭКРАН ДИСПЛЕЯ ЭТОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ЛИТЕР

А) ЭКРАН ОЧИЩАЕТСЯ С ПОДВОДОМ КУРСОРА В ЛЕВЫЙ ВЕРХНИЙ УГОЛ (ЛИТЕРА #12),

Б) ПЕРЕВОДИТСЯ СТРОКА (ЛИТЕРА #10) - КУРСОР ПЕРЕМЕЩАЕТСЯ В ПЕРВУЮ ПОЗИЦИЮ СЛЕДУЮЩЕЙ СТРОКИ ЭКРАНА ДИСПЛЕЯ,  
В) ВКЛЮЧАЕТСЯ ЗВОНОК (ЛИТЕРА ^L),  
Г) С ТЕКУЩЕЙ ПОЗИЦИИ ЭКРАНА ВЫДАЕТСЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ПРЕДСТАВИМЫХ ЛИТЕР КОНСТАНТЫ.

ДЛЯ СТРОКОВЫХ ДАННЫХ ОПРЕДЕЛЕНА ОПЕРАЦИЯ

А) КОНКАТЕНАЦИЯ (СЦЕПЛЕНИЯ) СТРОК (ЗАДАЕТСЯ ЗНАКОМ "+", ОПРЕДЕЛЯЕТ ПРИПИСЫВАНИЕ ЛИТЕР ВТОРОГО ОПЕРАНДА ВСЛЕД ЗА ПОСЛЕДНЕЙ ЛИТЕРОЙ ПЕРВОГО ОПЕРАНДА: 'A'+ 'BC' --> 'A BC'. ДЛИНА СТРОКИ РЕЗУЛЬТАТА НЕ ДОЛЖНА ПРЕВЫШАТЬ 255);

Б) ОТНОШЕНИЯ (ОПЕРАЦИИ ОТНОШЕНИЯ ИМЕЮТ МЕНЬШИЙ ПРИОРИТЕТ, ЧЕМ ОПЕРАЦИЯ КОНКАТЕНАЦИИ СТРОК). ПРИ ВЫЧИСЛЕНИИ ЗНАЧЕНИЯ ОТНОШЕНИЯ СЛЕВА НАПРАВО СРАВНИВАЮТСЯ КОДЫ КОИ-7 СООТВЕТСТВУЮЩИХ ЛИТЕР СТРОК ОПЕРАНДОВ. СТРОКИ РАВНЫ, ЕСЛИ СОВПАДАЮТ ПОСЛЕДОВАТЕЛЬНОСТИ ИХ ЛИТЕР И ДЛИНЫ СТРОК ОДИНАКОВЫ. БОЛЕЕ КОРОТКАЯ СТРОКА, ВСЕ ЛИТЕРЫ КОТОРОЙ СОВПАДАЮТ С СООТВЕТСТВУЮЩИМИ ЛИТЕРАМИ БОЛЕЕ ДЛИННОЙ СТРОКИ, СЧИТАЕТСЯ МЕНЬШЕ; ОТНОШЕНИЯ 'A' < 'A', 'ABC' > '012', '0' < 'CDF' ИМЕЮТ ЗНАЧЕНИЯ TRUE.

ДААННЫЕ ПРЕДОПРЕДЕЛЕННОГО ТИПА CHAR МОГУТ РАССМАТРИВАТЬСЯ КАК ЧАСТНЫЕ СЛУЧАИ СТРОКОВЫХ ДАННЫХ С ДЛИНОЙ СТРОКИ, РАВНОЙ 1.

ПРИ ПРИСВАИВАНИИ ЗНАЧЕНИЯ СТРОКОВОЙ ПЕРЕМЕННОЙ ЛИШЬ ПРАВЫЕ ЛИТЕРЫ ЗНАЧЕНИЯ СТРОКОВОГО ВЫРАЖЕНИЯ, ЕСЛИ ОНИ ЕСТЬ, ОТСЕКАЮТСЯ. НАПРИМЕР, ЕСЛИ ВЫПОЛНЯЮТСЯ ОПЕРАТОРЫ S:= 'ПРИМЕР'; S1:= 'УСЕЧЕНИЯ'; C:= 'ЗНАЧЕНИЯ', ТО ПОСЛЕ ВЫПОЛНЕНИЯ ОПЕРАТОРА C1:= S+ ' '+S1+ ' '+C; C1 ПОЛУЧАЕТ ЗНАЧЕНИЕ: 'ПРИМЕР УСЕЧЕНИЯ', ЛИТЕРЫ 'ЗНАЧЕНИЯ' ОТСЕЧЕНЫ (ДЛИНА СТРОКИ S РАВНА 6, ДЛИНА СТРОКИ S1 РАВНА 8, ДЛИНА СТРОКИ C РАВНА 8, ДЛИНА СТРОКИ C1 НЕ МОЖЕТ БЫТЬ БОЛЬШЕ 15).

ЗНАЧЕНИЕ СТРОКОВОЙ ПЕРЕМЕННОЙ - ЭЛЕМЕНТА СПИСКА ВВОДА ПРОЦЕДУРЫ READ/READLN НАБИРАЕТСЯ БЕЗ ОКЛАИМЛЯЮЩИХ АПОСТРОФОВ. КОЛИЧЕСТВО ЛИТЕР В НЕМ ДОЛЖНО БЫТЬ РАВНО , ЕСЛИ ЗА НИМ В НАБИРАЕМОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ЛИТЕР СЛЕДУЮТ ЗНАЧЕНИЯ ОЧЕРЕДНЫХ ЭЛЕМЕНТОВ СПИСКА ВВОДА, МОЖЕТ БЫТЬ МЕНЬШЕ В ПРОТИВНОМ СЛУЧАЕ.

ЕСЛИ VAR STR:STRING(10); X:REAL; TO, ЧТОБЫ STR ПОЛУЧИЛА ЗНАЧЕНИЕ 'ПРИМЕР', X - ЗНАЧЕНИЕ -7.2, ПРИ ВЫПОЛНЕНИИ ПРОЦЕДУРЫ READ(STR,X), НЕОБХОДИМО НАБРАТЬ ПРИМЕР -7.2<ET>, ПРИ ВЫПОЛНЕНИИ ПРОЦЕДУРЫ READ(X,STR)-->-7.2ПРИМЕР <ET>, ПРИ ВЫПОЛНЕНИИ ПРОЦЕДУР READ(STR); READ(X)--> ПРИМЕР <ET> И -7.2 <ET>.

ДЛЯ УДОБСТВА МОЖНО РЕКОМЕНДОВАТЬ СТРОКОВУЮ ПЕРЕМЕННУЮ ВКЛЮЧАТЬ В СПИСОК ВВОДА ПОСЛЕДНЕЙ ИЛИ ФОРМИРОВАТЬ СПИСОК ВВОДА С ОДНИМ ЭЛЕМЕНТОМ СТРОКОВОГО ТИПА.

ПРОЦЕДУРА WRITE/WRITELN ВЫДАЕТ ЗНАЧЕНИЕ СТРОКОВОГО ВЫРАЖЕНИЯ - ЭЛЕМЕНТА СПИСКА ВЫВОДА С ТЕКУЩЕЙ ПОЗИЦИИ ЭКРАНА БЕЗ ОКЛАИМЛЯЮЩИХ АПОСТРОФОВ. ЕСЛИ S='СТРОКА', ТО ПРОЦЕДУРА WRITE(S, ':', 'ТЕКСТ':?) ВЫДАЕТ СТРОКА: ТЕКСТ. ЕСЛИ ШИРИНА ПОЛЯ ДЛЯ ВЫВОДА СТРОКИ НЕДОСТАТОЧНА, ТО ПАРАМЕТР, УКАЗЫВАЮЩИЙ ШИРИНУ ПОЛЯ, ИГНОРИРУЕТСЯ.

#### ПРЕДОПРЕДЕЛЕННЫЕ ПРОЦЕДУРЫ

ОБЪЯВИМ ЧЕРЕЗ S СТРОКОВУЮ ПЕРЕМЕННУЮ; SV - СТРОКОВОЕ ВЫРАЖЕНИЕ; N,K - ЦЕЛЫЕ ВЫРАЖЕНИЯ; VR - ЧИСЛОВУЮ ПЕРЕМЕННУЮ; V - ЧИСЛОВОЕ ВЫРАЖЕНИЕ.

А) DELETE(S,N,K) - УДАЛИТЬ В СТРОКЕ S, НАЧИНАЯ С ЛИТЕРЫ С НОМЕРОМ N, К ЛИТЕР. РЕЗУЛЬТАТ ПРИСВОИТЬ ПЕРЕМЕННОЙ S. НАПРИМЕР,

S:= 'ВОДОПРОВОД'. ПОСЛЕ ВЫПОЛНЕНИЯ ПРОЦЕДУРЫ DELETE(S,4,3)  
S ИМЕЕТ ЗНАЧЕНИЕ 'ВОДОВОД'.  
ЗАМЕЧАНИЕ: СИМВОЛЫ ЗА ПРЕДЕЛАМИ СТРОКИ НЕ УДАЛЯЮТСЯ,  
1< N< 255.

Б) INSERT(SV,S,N) - НАЧИНАЯ С ПОЗИЦИИ N, ВСТАВИТЬ В СТРОКУ  
В ЗНАЧЕНИЕ SV(1< N< 255), А ЕСЛИ N БОЛЬШЕ ТЕКУЩЕЙ ДЛИНЫ S, ТО  
ПРОИЗВЕСТИ КОНКАТЕНАЦИЮ S И SV. РЕЗУЛЬТАТ ПРИСВОИТЬ S С  
УДАЛЕНИЕМ ПРАВЫХ ЛИТЕР, ЕСЛИ ЕСТЬ ЛИШНИЕ. НАПРИМЕР, ЕСЛИ  
S:= 'ВОДОПРОВОД', ТО ПОСЛЕ ВЫПОЛНЕНИЯ ПРОЦЕДУРЫ INSERT ('НОВЫЙ',  
S, 11) S='ВОДОПРОВОД НОВЫ', ЕСЛИ S ИМЕЕТ ТИП STRING[15].

В) STR(V,S) - ПРЕОБРАЗОВАТЬ ЗНАЧЕНИЕ ЧИСЛОВОГО ВЫРАЖЕНИЯ V  
В ПОСЛЕДОВАТЕЛЬНОСТЬ ЛИТЕР, ПРЕДСТАВЛЯЮЩИХ ЕГО В ДЕСЯТИЧНОЙ  
СИСТЕМЕ СЧИСЛЕНИЯ, С ПРИСВАИВАНИЕМ РЕЗУЛЬТАТА ПЕРЕМЕННОЙ S (ПРЕОБ-  
РАЗОВАНИЕ ВЫПОЛНЯЕТСЯ ПО ТЕМ ЖЕ ПРАВИЛАМ, ЧТО И ПРИ ВЫПОЛНЕНИИ  
ПРОЦЕДУРЫ WRITE(V), (WRITELN(V)). НАПРИМЕР, VAR L: INTEGER;  
SL: STRING[8]; SPI: STRING[20];. ПОСЛЕ ВЫПОЛНЕНИЯ L:=456;  
STR(L,SL); STR(PI/10,SPI) SL='456', SPI=' 3.1415926536E-01'.  
В КАЧЕСТВЕ ПЕРВОГО ПАРАМЕТРА ПЕРЕМЕННОЙ VR. ЦЕЛАЯ ПЕРЕМЕННАЯ КОНС-  
ТРУКЦИИ  $\alpha: \beta: \gamma$  С  $\alpha$  - ЧИСЛОВЫМ ВЫРАЖЕНИЕМ (СМ.1.8.1). ПРОЦЕ-  
ДУРА STR(PI/10:10:6,SPI) ПРИСВАИВАЕТ SPI ЗНАЧЕНИЕ '0.314159'  
(ЗНАЧЕНИЕ SPI СОВПАДАЕТ С ФОРМОЙ ВНЕШНЕГО ПРЕДСТАВЛЕНИЯ ЭЛЕМЕН-  
ТА СПИСКА ПРОЦЕДУРЫ WRITE/WRITELN).

Г) VAL(SV, VR, ERROR) - ПРЕОБРАЗОВАТЬ ЗНАЧЕНИЕ СТРОКОВО-  
ГО ВЫРАЖЕНИЯ SV, ЯВЛЯЮЩЕГОСЯ ВЕРНЫМ СИМВОЛЬНЫМ ПРЕДСТАВЛЕНИЕМ  
ДЕСЯТИЧНОГО ЧИСЛА БЕЗ НАЧАЛЬНЫХ И КОНЕЧНЫХ ПРОБЕЛОВ, ВО  
ВНУТРИМАШИННОЕ (В СООТВЕТСТВИИ С ТИПОМ VR) ПРЕДСТАВЛЕНИЕ ЧИСЛА  
С ПРИСВАИВАНИЕМ РЕЗУЛЬТАТА ПЕРЕМЕННОЙ VR. ЦЕЛАЯ ПЕРЕМЕННАЯ ERROR  
ПОЛУЧАЕТ ЗНАЧЕНИЕ 0, ЕСЛИ ПРЕОБРАЗОВАНИЕ ВЫПОЛНЕНО, ИЛИ  
НОМЕР НЕДОПУСТИМОЙ ЛИТЕРЫ В SV, В ПРОТИВНОМ СЛУЧАЕ. НАПРИМЕР,  
ЕСЛИ VAR X: REAL; M,L,L1: INTEGER, ТО ПОСЛЕ ВЫПОЛНЕНИЯ ПРОЦЕДУР  
VAL('1.37'X,L); VAL('1,37',M,L1) X=1.37, L=0, M ИМЕЕТ  
НЕОПРЕДЕЛЕННОЕ ЗНАЧЕНИЕ, L1=2.

### 2.3. ПЕРЕЧИСЛИМЫЙ ТИП

ПРЕОПРЕДЕЛЕННЫЙ ТИП BOOLEAN ОПРЕДЕЛЯЕТСЯ КАК ТИП СО  
ЗНАЧЕНИЯМИ ДАННЫХ, ВЫБИРАЕМЫХ ИЗ МНОЖЕСТВА (FALSE, TRUE). FALSE,  
TRUE - КОНСТАНТЫ-ИДЕНТИФИКАТОРЫ ТИПА.

ПАСКАЛЬ ДОПУСКАЕТ ПО ТАКОМУ ЖЕ ПРАВИЛУ СОЗДАВАТЬ В ПРОГРАММЕ  
НОВЫЕ ТИПЫ, ПЕРЕЧИСЛЯЯ ПРИ ОБЪЯВЛЕНИИ ТИПА ВСЕ ДОПУСТИМЫЕ  
ЗНАЧЕНИЯ ДАННЫХ ЭТОГО ТИПА - ПЕРЕЧИСЛИМЫЕ ТИПЫ. ДОПУСТИМЫЕ  
ЗНАЧЕНИЯ - ИДЕНТИФИКАТОРЫ, КОТОРЫЕ ЯВЛЯЮТСЯ КОНСТАНТАМИ - ИДЕНТИ-  
ФИКАТОРАМИ ТИПА. НАПРИМЕР, РАЗДЕЛ  
TYPE

```
FELINE=(CAT, TIGER, LION, PUMA);  
SPECTRUM=(RED, ORANGE, YELLOW, GREEN, BLUE, VIOLET);  
ОПРЕДЕЛЯЕТ ТИП FELINE (КОШАЧЬИХ), ДОПУСТИМЫМИ ЗНАЧЕНИЯМИ ДАННЫХ  
ЭТОГО ТИПА ЯВЛЯЮТСЯ CAT(КОТ), TIGER(ТИГР), LION(ЛЕВ), PUMA(ПУМА)  
- КОНСТАНТЫ - ИДЕНТИФИКАТОРЫ ТИПА; ТИП SPECTRUM(СПЕКТР), ДОПУС-  
ТИМЫМИ ЗНАЧЕНИЯМИ ДАННЫХ ТИПА ЯВЛЯЮТСЯ RED(КРАСНЫЙ), ORANGE(ОР-  
АНЖЕВЫЙ), YELLOW(ЖЕЛТЫЙ), GREEN(ЗЕЛЕНЫЙ), BLUE(СИНИЙ), VIOLET  
(ФИОЛЕТОВЫЙ) - КОНСТАНТЫ-ИДЕНТИФИКАТОРЫ ТИПА.
```

КАЖДООМ КОНСТАНТЕ ТИПА ПОСТАВЛЕН В СООТВЕТСТВИЕ ЕЕ НОМЕР В  
СПИСКЕ ОПРЕДЕЛЕНИЯ ТИПА. НУМЕРАЦИЯ НАЧИНАЕТСЯ С НУЛЯ. В ПАМЯТИ  
МАШИНЫ ЗНАЧЕНИЕ ДАННОГО ПЕРЕЧИСЛИМОГО ТИПА ХРАНИТСЯ НЕ В ВИДЕ

КОНСТАНТЫ-ИДЕНТИФИКАТОРА, А В ВИДЕ ЕЕ НОМЕРА В СПИСКЕ КОНСТАНТ ТИПА. ПОЭТОМУ ПЕРЕЧИСЛИМЫЙ ТИП ЯВЛЯЕТСЯ ОРДИНАЛЬНЫМ: ORD(CAT)=0, ORD(PUMA)=3, ORD(GREEN)=3, SUCC(CAT)=TIGER, PRED(VIOLET)=BLUE; И ДЛЯ ДАННЫХ ПЕРЕЧИСЛИМОГО ТИПА ОПРЕДЕЛЕНА ОПЕРАЦИЯ ОТНОШЕНИЯ: СРАВНИВАЮТСЯ ЗНАЧЕНИЯ ФУНКЦИИ ORD ОПЕРАНДОВ. НАПРИМЕР, CAT<PUMA, BLUE>RED.

РАЗДЕЛ VAR PRF:FELINE;

CR, CL, COLOR:SPECTRUM;

RD, DT:(FATHER, MOTHER, SON, DAUGHTER);

ОПИСЫВАЕТ ПЕРЕМЕННУЮ PRF ТИПА FELINE, ПЕРЕМЕННЫЕ CR, CL, COLOR ТИПА SPECTRUM, ИСПОЛЬЗУЯ ИМЕНОВАННЫЕ ПЕРЕЧИСЛИМЫЕ ТИПЫ, И ПЕРЕМЕННЫЕ RD, DT, ИСПОЛЬЗУЯ АНОНИМНЫЙ ПЕРЕЧИСЛИМЫЙ ТИП С КОНСТАНТАМИ-ИДЕНТИФИКАТОРАМИ FATHER(ОТЕЦ), MOTHER(МАТЬ), SON(СЫН), DAUGHTER(ДОЧЬ).

ПЕРЕМЕННОЙ ПЕРЕЧИСЛИМОГО ТИПА МОЖНО ПРИСВОИТЬ ЗНАЧЕНИЕ ВЫРАЖЕНИЯ ТОГО ЖЕ ТИПА:

CL:=RED; COLOR:=SUCC(CL); CR:=COLOR. ДАННЫЕ ПЕРЕЧИСЛИМОГО ТИПА МОГУТ ИСПОЛЬЗОВАТЬСЯ В ОПЕРАТОРЕ FOR: FOR DT:=FATHER TO DAUGHTER DO ...

ДАННЫЕ ПЕРЕЧИСЛИМЫХ ТИПОВ НЕ МОГУТ БЫТЬ ЭЛЕМЕНТАМИ СПИСКОВ ВВОДА-ВЫВОДА. МОЖНО ВЫДАТЬ ТОЛЬКО ЗНАЧЕНИЕ ФУНКЦИИ ORD ДЛЯ ДАННОГО ПЕРЕЧИСЛИМОГО ТИПА.

ОСНОВНОЕ НАЗНАЧЕНИЕ ВВЕДЕНИЯ ПЕРЕЧИСЛИМЫХ ТИПОВ - ПРИБЛИЖЕНИЕ ЯЗЫКА ПРОГРАММЫ К ЕСТЕСТВЕННОМУ ЯЗЫКУ - УЛУЧШЕНИЕ ЧИТАЕМОСТИ ПРОГРАММЫ.

<ОПРЕДЕЛЕНИЕ ПЕРЕЧИСЛИМОГО ТИПА>::=(**<КОНСТАНТА-ИДЕНТИФИКАТОР>** } , **<КОНСТАНТА-ИДЕНТИФИКАТОР>** } ).

ЗАМЕЧАНИЕ. КОНСТАНТА-ИДЕНТИФИКАТОР МОЖЕТ БЫТЬ ЭЛЕМЕНТОМ СПИСКА ОПРЕДЕЛЕНИЯ ТОЛЬКО ОДНОГО ПЕРЕЧИСЛИМОГО ТИПА. ДЛЯ ДРУГИХ ЦЕЛЕЙ ЭТОТ ИДЕНТИФИКАТОР ИСПОЛЬЗОВАТЬСЯ НЕ МОЖЕТ.

## 2.4. ТИП ДИАПАЗОН

ЯЗЫК ПАСКАЛЬ ПОЗВОЛЯЕТ ВЫДЕЛЯТЬ ГРУППУ ПОСЛЕДОВАТЕЛЬНЫХ ЗНАЧЕНИЙ ОРДИНАЛЬНОГО ТИПА И СМОТРЕТЬ НА НЕЕ КАК НА НОВЫЙ ТИП.

ТАКОЙ ТИП ОПРЕДЕЛЯЕТСЯ УКАЗАНИЕМ ПЕРВОГО И ПОСЛЕДНЕГО ЗНАЧЕНИЯ, ВХОДЯЩИХ В ГРУППУ: <КОНСТАНТА1>..<КОНСТАНТА2> И НАЗЫВАЕТСЯ ТИПОМ ДИАПАЗОН. ТИП КОНСТАНТ, ИСПОЛЬЗУЕМЫХ В ОПРЕДЕЛЕНИИ ТИПА ДИАПАЗОН, ДОЛЖЕН БЫТЬ ОБЩИМ (<КОНСТАНТА1> <= <КОНСТАНТА2>).

ЕГО НАЗЫВАЮТ **БАЗОВЫМ ТИПОМ** ДЛЯ ТИПА ДИАПАЗОН. НАПРИМЕР, РАЗДЕЛЫ

CONST DL=00;

TYPE SPECTRUM=(RED, ORANGE, YELLOW, GREEN, BLUE, VIOLET);

WARM=RED..YELLOW;

COLD=GREEN..VIOLET;

DN=1..DL;

VAR Z,Z1: WARM; C: COLD;

NP, N: DN; S: 'A'..'Z';

ОПРЕДЕЛЯЮТ ПЕРЕЧИСЛИМЫЙ ТИП SPECTRUM, ИМЕНОВАННЫЕ ТИПЫ ДИАПАЗОН WARM, COLD С БАЗОВЫМ ТИПОМ SPECTRUM, ИМЕНОВАННЫЙ ТИП DN С БАЗОВЫМ ПРЕДОПРЕДЕЛЕННЫМ ТИПОМ INTEGER, АНОНИМНЫЙ ТИП - ПОДМОЖЕСТВО ТИПА CHAR, ВКЛЮЧАЮЩЕЕ ЗАГЛАВНЫЕ ЛАТИНСКИЕ БУКВЫ; ОПИСЫВАЮТ ПЕРЕМЕННЫЕ Z, Z1 ТИПА WARM; С ТИПА COLD, NP, N ТИПА DN; В АНОНИМНОГО ТИПА.

ДЛЯ ДАННЫХ ТИПА ДИАПАЗОН ОПРЕДЕЛЕНА ТЕ ЖЕ ОПЕРАЦИИ, ЧТО И ДЛЯ БАЗОВОГО ТИПА. В ВЫРАЖЕНИЯХ ДОПУСТИМО СМЕШИВАНИЕ ОПЕРАНДОВ ТИПА ДИАПАЗОН И ОПЕРАНДОВ БАЗОВОГО ТИПА. ПЕРЕМЕННОЙ БАЗОВОГО ТИПА МОЖНО ПРИСВОИТЬ ЗНАЧЕНИЕ ВЫРАЖЕНИЯ ТИПА ДИАПАЗОН И НАОБОРОТ.

УКАЗАНИЕ О ДИАПАЗОНЕ ПЕРЕДАЕТ ДОПОЛНИТЕЛЬНУЮ ИНФОРМАЦИЮ О ЗНАЧЕНИЯХ ДАННЫХ. ЭТО ПОЗВОЛЯЕТ

А) ФОРМУЛИРОВАТЬ ПРОБЛЕМУ В БОЛЕЕ НАГЛЯДНОЙ ФОРМЕ, ЧТО СПОСОБСТВУЕТ УЛУЧШЕНИЮ ЧИТАЕМОСТИ ПРОГРАММЫ;

Б) ВВОДИТЬ КОНТРОЛЬ ЗА ДОПУСТИМОСТЬЮ ЗНАЧЕНИЙ ДАННЫХ КАК НА ЭТАПЕ ТРАНСЛЯЦИИ, ТАК И НА ЭТАПЕ ВЫПОЛНЕНИЯ ПРОГРАММЫ, ОБЛЕГЧИТЬ ЭТИМ ОТЛАДКУ ПРОГРАММЫ;

В) ЭКОНОМИТЬ ПАМЯТЬ ПРИ РАЗМЕЩЕНИИ ЗНАЧЕНИЙ ДАННЫХ (ДЛЯ ХРАНЕНИЯ ЗНАЧЕНИЙ N<sub>r</sub>, N ТИПА DN ДОСТАТОЧНО ОДНОГО БАЙТА, А НЕ ДВУХ, КАК ДЛЯ ЦЕЛОГО ДАННОГО).

## 2.5. ОПЕРАТОР ВЫБОРА (CASE)

УСЛОВНЫЙ ОПЕРАТОР (IF) ПОЗВОЛЯЕТ В УДОБНОЙ ФОРМЕ ОПИСЫВАТЬ АЛГОРИТМ, ТРЕБУЮЩИЙ ВЫБРАТЬ ОДНУ ИЗ ДВУХ ВОЗМОЖНЫХ АЛЬТЕРНАТИВ. ИСПОЛЬЗОВАНИЕ ВЛОЖЕННЫХ IF, ОБЕСПЕЧИВАЮЩИХ ВЫБОР ОДНОЙ ИЗ ЧИСЛА АЛЬТЕРНАТИВ ПРИ ИХ ОБЩЕМ ЧИСЛЕ БОЛЬШЕМ 3, ДЕЛАЕТ ПРОГРАММУ ГРОМОЗДКОЙ, ТРУДНО ВОСПРИНИМАЕМОЙ.

ОПЕРАТОР ВЫБОРА (ВАРИАНТА) ЯВЛЯЕТСЯ ОБОБЩЕНИЕМ УСЛОВНОГО ОПЕРАТОРА, ПОЗВОЛЯЕТ В НАГЛЯДНОЙ ФОРМЕ ОПИСАТЬ ВЫБОР ОДНОЙ ИЗ МНОГИХ АЛЬТЕРНАТИВ:

```
CASE <ВЫРАЖЕНИЕ> OF
  <СПИСОК КОНСТАНТ>: <ОПЕРАТОР>
{ ; <СПИСОК КОНСТАНТ>: <ОПЕРАТОР> }
[ ELSE <ОПЕРАТОР> { ; <ОПЕРАТОР> } ]
```

END

<СПИСОК КОНСТАНТ> ::= <ЭЛЕМЕНТ СПИСКА> { , <ЭЛЕМЕНТ СПИСКА> }

<ЭЛЕМЕНТ СПИСКА> ::= <КОНСТАНТА> ! <КОНСТАНТА1> .. <КОНСТАНТА2>

ВЫРАЖЕНИЕ И КОНСТАНТЫ СПИСКОВ КОНСТАНТ ДОЛЖНЫ БЫТЬ ОДНОГО ОРДИНАЛЬНОГО ТИПА.

ОПЕРАТОРЫ, ВКЛЮЧАЕМЫЕ В ОПЕРАТОР CASE, РЕАЛИЗУЮТ ВАРИАНТЫ (АЛЬТЕРНАТИВЫ) АЛГОРИТМА. КОНСТАНТЫ СПИСКОВ КОНСТАНТ НАЗЫВАЮТ МЕТКАМИ ВАРИАНТОВ, ВЫРАЖЕНИЕ - СЕЛЕКТОРОМ ВАРИАНТА.

ДЕЙСТВИЯ, ЗАДАВАЕМЫЕ ОПЕРАТОРОМ:

А) ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ СЕЛЕКТОРА ВАРИАНТА;

Б) ВЫДЕЛЕНИЕ СПИСКА КОНСТАНТ, ВКЛЮЧАЮЩЕГО ЗНАЧЕНИЕ СЕЛЕКТОРА (ОНО СОВПАДАЕТ С ОДНОЙ ИЗ КОНСТАНТ СПИСКА ИЛИ ПОПАДАЕТ В ОДИН ИЗ ДИАПАЗОНОВ СПИСКА);

В) ЕСЛИ СПИСОК КОНСТАНТ ОБНАРУЖЕН, ВЫПОЛНЕНИЕ СООТВЕТСТВУЮЩЕГО ОПЕРАТОРА ВАРИАНТА;

Г) ЕСЛИ ЗНАЧЕНИЕ СЕЛЕКТОРА НЕ ПРИНАДЛЕЖИТ НИ ОДНОМУ ИЗ СПИСКОВ КОНСТАНТ И В ОПЕРАТОРЕ УКАЗАНА ЧАСТЬ С ELSE, ВЫПОЛНЕНИЕ ОПЕРАТОРОВ, ЗАКЛЮЧЕННЫХ МЕЖДУ ELSE И END;

Д) ПЕРЕХОД К ВЫПОЛНЕНИЮ СЛЕДУЮЩЕГО ОПЕРАТОРА.

НАПРИМЕР, ПРОГРАММА

```
PROGRAM EXPRESSION;
```

```
(* ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ВЫРАЖЕНИЯ *)
```

```
VAR
```

```
  X, Y, Z: REAL;
```

```

S:CHAR;
B:BOOLEAN;
BEGIN
  READ(S,X,Y); B:=TRUE; (* ЗНАК ОПЕРАЦИИ ДОПУСТИМ *)
  CASE S OF
    '+': Z:=X+Y;
    '-': Z:=X-Y;
    '*': Z:=X*Y;
    '/': Z:=X/Y;
  ELSE WRITELN; WRITELN('ЗНАК ОПЕРАЦИИ ',S,
    ' НЕДОПУСТИМ '); B:=FALSE

  END; (* CASE *)
  IF B THEN
    BEGIN
      WRITELN('X=',X,' Y=',Y);
      WRITELN(' ',Z,'--ЗНАЧЕНИЕ ВЫРАЖЕНИЯ X',S,'Y')
    END
  END. (* EXPRESSION *)

```

ЗАПРАШИВАЕТ ЛИТЭРУ - ЗНАК ОПЕРАЦИИ И ДВА ВЕЩЕСТВЕННЫХ ЧИСЛА - ОПЕРАНДЫ ОПЕРАЦИИ; ВЫЧИСЛЯЕТ РЕЗУЛЬТАТ ОПЕРАЦИИ НАД ЧИСЛАМИ И ВЫДАЕТ ЕГО, ЕСЛИ ЗНАК ОПЕРАЦИИ ДОПУСТИМ, ВЫДАЕТ СООТВЕТСТВУЮЩЕЕ СООБЩЕНИЕ, ЕСЛИ ЗНАК ОПЕРАЦИИ НЕДОПУСТИМ.

СПИСКИ КОНСТАНТ ВАРИАНТОВ В ОПЕРАТОРЕ CASE НЕ ДОЛЖНЫ ПЕРЕСЕКАТЬСЯ. ВЫПОЛНЕНИЕ ОПЕРАТОРА CASE В СЛУЧАЕ ОТСУТСТВИЯ ЧАСТИ ELSE В НЕМ И НЕСОВПАДАЮЩЕГО ЗНАЧЕНИЯ СЕЛЕКТОРА ВАРИАНТА НИ С ОДНОЙ ИЗ КОНСТАНТ, ОПРЕДЕЛЯЕМЫХ СПИСКОМ КОНСТАНТ, ЭКВИВАЛЕНТНО ПУСТОМУ ОПЕРАТОРУ.

## 2.6. ПОДПРОГРАММЫ

ПАСКАЛЬ ДОПУСКАЕТ ИСПОЛЬЗОВАНИЕ ПОДПРОГРАММ ПРОЦЕДУР (PROCEDURE) (В ФОРТРАНЕ ИХ АНАЛОГАМИ ЯВЛЯЮТСЯ ПОДПРОГРАММЫ SUBROUTINE) И ПОДПРОГРАММ ФУНКЦИИ (FUNCTION). КАЖДАЯ ПОДПРОГРАММА ПОДЛЕЖИТ ОПИСАНИЮ <ОПИСАНИЕ ПОДПРОГРАММЫ>::=<ЗАГОЛОВОК><БЛОК>; БЛОК ИМЕЕТ ТУ ЖЕ СТРУКТУРУ И НАЗНАЧЕНИЕ, ЧТО И РАССМОТРЕННЫЙ РАНЕЕ ПРОГРАММНЫЙ БЛОК (СМ.1.6). ЗАГОЛОВОК ОПРЕДЕЛЯЕТ ИМЯ, ВИД (ПРОЦЕДУРА ИЛИ ФУНКЦИЯ) И СПИСОК ФОРМАЛЬНЫХ ПАРАМЕТРОВ ПОДПРОГРАММЫ. СПИСОК ФОРМАЛЬНЫХ ПАРАМЕТРОВ МОЖЕТ ОТСУТСТВОВАТЬ.

```

<ИМЯ ПОДПРОГРАММЫ>::=<ИДЕНТИФИКАТОР>
<СПИСОК ФОРМАЛЬНЫХ ПАРАМЕТРОВ>::=<РАЗДЕЛ ФОРМАЛЬНЫХ ПАРАМЕТРОВ>
{;<РАЗДЕЛ ФОРМАЛЬНЫХ ПАРАМЕТРОВ>}
<РАЗДЕЛ ФОРМАЛЬНЫХ ПАРАМЕТРОВ>::=<ГРУППА ФОРМАЛЬНЫХ ПАРАМЕТРОВ>!
УВВ <ГРУППА ФОРМАЛЬНЫХ ПАРАМЕТРОВ>
<ГРУППА ФОРМАЛЬНЫХ ПАРАМЕТРОВ>::=<ИДЕНТИФИКАТОР>
{,<ИДЕНТИФИКАТОР>} :<ИДЕНТИФИКАТОР ТИПА>

```

ТО ЕСТЬ ГРУППА ФОРМАЛЬНЫХ ПАРАМЕТРОВ ОПРЕДЕЛЯЕТ ОДИН ИЛИ НЕСКОЛЬКО ФОРМАЛЬНЫХ ПАРАМЕТРОВ ОДНОГО ТИПА. НАПРИМЕР, (X, Y:REAL; B:BOOLEAN) - ДВЕ ГРУППЫ. ПЕРВАЯ ГРУППА ОПРЕДЕЛЯЕТ ДВА ФОРМАЛЬНЫХ ПАРАМЕТРА ВЕЩЕСТВЕННОГО ТИПА (X,Y), ВТОРАЯ ГРУППА ОПРЕДЕЛЯЕТ ОДИН ФОРМАЛЬНЫЙ ПАРАМЕТР БУЛЕВСКОГО ТИПА.

```

<ЗАГОЛОВОК ПОДПРОГРАММЫ ПРОЦЕДУРЫ>::=PROCEDURE
<ИМЯ ПОДПРОГРАММЫ><СПИСОК ФОРМАЛЬНЫХ ПАРАМЕТРОВ>;
НАПРИМЕР, ЗАГОЛОВОК PROCEDURE P(X,Y:REAL; VAR B:BOOLEAN);
ОПРЕДЕЛЯЕТ ПОДПРОГРАММУ ПРОЦЕДУРУ P С ФОРМАЛЬНЫМИ ПАРАМЕТРАМИ
X, Y, B.

```

АЛГОРИТМ ПОДПРОГРАММЫ МОЖЕТ БЫТЬ ОПИСАН В ВИДЕ ПОДПРОГРАММЫ-ФУНКЦИИ ЛИШЬ В ТОМ СЛУЧАЕ, ЕСЛИ ОН ИМЕЕТ ОДИН РЕЗУЛЬТАТ ПРОСТОГО (ПРЕДПОДЕЛЕННОГО, ПЕРЕЧИСЛИМОГО ТИПА ИЛИ ТИПА ДИАПАЗОН) ТИПА ИЛИ ССЫЛОЧНОГО ТИПА (ЗНАЧЕНИЕ ФУНКЦИИ).

<ЗАГОЛОВОК ПОДПРОГРАММЫ ФУНКЦИИ>:=FUNCTION<ИМЯ ПОДПРОГРАММЫ><СПИСОК ФОРМАЛЬНЫХ ПАРАМЕТРОВ>:<ИДЕНТИФИКАТОР ТИПА ЗНАЧЕНИЯ ФУНКЦИИ>;

НАПРИМЕР, ЗАГОЛОВОК FUNCTION F (I:INTEGER):REAL; ОПРЕДЕЛЯЕТ ФУНКЦИЮ F С ОДНИМ ФОРМАЛЬНЫМ ПАРАМЕТРОМ I ЦЕЛОГО ТИПА И ЗНАЧЕНИЕМ ФУНКЦИИ ВЕЩЕСТВЕННОГО ТИПА.

В РАЗДЕЛЕ ОПЕРАТОРОВ БЛОКА ФУНКЦИИ ДОЛЖЕН БЫТЬ ХОТЯ БЫ ОДИН ОПЕРАТОР, КОТОРЫЙ ПРИСВАИВАЕТ ЗНАЧЕНИЕ ИДЕНТИФИКАТОРУ ФУНКЦИИ (ИМЯ ФУНКЦИИ СОВПАДАЕТ С ИМЕНЕМ РЕЗУЛЬТАТА). НАПРИМЕР, IF I=1 THEN F:=SIN(I/30.0) ELSE F:=COS(I).

ДЛЯ ВЫЗОВА ПОДПРОГРАММЫ УКАЗЫВАЕТСЯ ЕЕ ИМЯ И СПИСОК ФАКТИЧЕСКИХ ПАРАМЕТРОВ (ЕСЛИ ПОДПРОГРАММА ИМЕЕТ СПИСОК ФОРМАЛЬНЫХ ПАРАМЕТРОВ).

<СПИСОК ФАКТИЧЕСКИХ ПАРАМЕТРОВ>:=(<ФАКТИЧЕСКИЙ ПАРАМЕТР>{ , <ФАКТИЧЕСКИЙ ПАРАМЕТР>} )  
<ФАКТИЧЕСКИЙ ПАРАМЕТР>:=<ВЫРАЖЕНИЕ>!<ПЕРЕМЕННАЯ>

МЕЖДУ СПИСКАМИ ФОРМАЛЬНЫХ И ФАКТИЧЕСКИХ ПАРАМЕТРОВ УСТАНОВЛИВАЕТСЯ ВЗАИМНООДНОЗНАЧНОЕ СООТВЕТСТВИЕ. ДОЛЖНЫ СОВПАДАТЬ КОЛИЧЕСТВО, ПОРЯДОК СЛЕДОВАНИЯ, ТИПЫ ФОРМАЛЬНЫХ И ФАКТИЧЕСКИХ ПАРАМЕТРОВ. НАПРИМЕР, ДЛЯ ПРОЦЕДУР P И F СПИСКИ ФАКТИЧЕСКИХ ПАРАМЕТРОВ МОГУТ ИМЕТЬ ВИД (-7.2, N/2, W) И (2) СООТВЕТСТВЕННО, ЕСЛИ VAR W:BOOLEAN; R,N:REAL;.

ОБРАЩЕНИЕ К ПОДПРОГРАММЕ ЗАДАЕТ ВЫЗОВ ПОДПРОГРАММЫ:  
<ВЫЗОВ ПОДПРОГРАММЫ>:=<ИМЯ ПОДПРОГРАММЫ><СПИСОК ФАКТИЧЕСКИХ ПАРАМЕТРОВ>

ОБРАЩЕНИЕ К ПОДПРОГРАММЕ ПРОЦЕДУРЕ ИМЕЕТ ВИД САМОСТОЯТЕЛЬНОГО ОПЕРАТОРА - ОПЕРАТОРА ВЫЗОВА ПРОЦЕДУРЫ (КАК READ, WRITE). ОБРАЩЕНИЕ К ПОДПРОГРАММЕ ФУНКЦИИ УКАЗЫВАЕТСЯ В КАЧЕСТВЕ ОПЕРАНДА ВЫРАЖЕНИЯ И НАЗЫВАЕТСЯ ВЫЗОВОМ (УКАЗАТЕЛЕМ, ОБРАЩЕНИЕМ К) ФУНКЦИИ. P(-7.2, N/2,W); - ОПЕРАТОР ВЫЗОВА ПРОЦЕДУРЫ P. ОПЕРАТОР R:=SQR(F(2)) ЗАДАЕТ ВЫЧИСЛЕНИЕ КВАДРАТА ЗНАЧЕНИЯ ФУНКЦИИ F ДЛЯ ФАКТИЧЕСКОГО ПАРАМЕТРА 2 С ПРИСВАИВАНИЕМ РЕЗУЛЬТАТА ПЕРЕМЕННОЙ R.

ПАСКАЛЬ ДОПУСКАЕТ ДВА СПОСОБА ПРИЕМА ПОДПРОГРАММОЙ ФАКТИЧЕСКИХ ПАРАМЕТРОВ: ПО ЗНАЧЕНИЮ И ПО ССЫЛКЕ.

В ПОДПРОГРАММУ ПЕРЕДАЕТСЯ АДРЕС ФАКТИЧЕСКОГО ПАРАМЕТРА (ЗНАЧЕНИЯ ВЫРАЖЕНИЯ ИЛИ ПЕРЕМЕННОЙ). ЕСЛИ ПОДПРОГРАММА ПРИНИМАЕТ ФАКТИЧЕСКИЙ ПАРАМЕТР ПО ЗНАЧЕНИЮ, В ЕЕ ТЕЛЕ ВЫДЕЛЯЕТСЯ ПОЛЕ ПАМЯТИ (ПОЛЕ ДЛЯ ПЕРЕМЕННОЙ - ФОРМАЛЬНОГО ПАРАМЕТРА), В КОТОРОЕ ПЕРЕСЫЛАЕТСЯ ЗНАЧЕНИЕ ФАКТИЧЕСКОГО ПАРАМЕТРА. ВСЕ ДЕЙСТВИЯ, ПРЕДУСМОТРЕННЫЕ В ПОДПРОГРАММЕ ДЛЯ ФОРМАЛЬНОГО ПАРАМЕТРА, ВЫПОЛНЯЮТСЯ С ИСПОЛЬЗОВАНИЕМ ЭТОГО ПОЛЯ, ПОЭТОМУ ВОЗМОЖНЫЕ ИЗМЕНЕНИЯ ФОРМАЛЬНОГО ПАРАМЕТРА НИКАК НЕ СКАЗЫВАЮТСЯ НА ЗНАЧЕНИИ ФАКТИЧЕСКОГО ПАРАМЕТРА. ПАРАМЕТРЫ, ПРИНИМАЕМЫЕ ПОДПРОГРАММОЙ ТАКИМ ОБРАЗОМ, НАЗЫВАЮТ ПАРАМЕТРАМИ ЗНАЧЕНИЙ. В ПРОЦЕДУРЕ P ПАРАМЕТРАМИ ЗНАЧЕНИЯМИ ЯВЛЯЮТСЯ X, Y, В ФУНКЦИИ F - ПАРАМЕТР I.

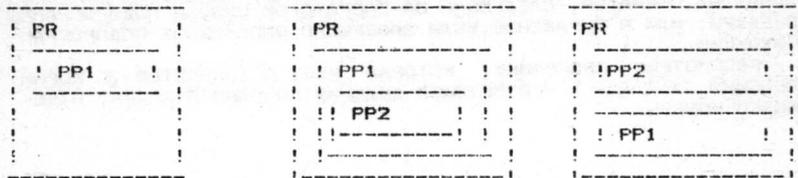
ЕСЛИ ФАКТИЧЕСКИЙ ПАРАМЕТР ПРИНИМАЕТСЯ ПОДПРОГРАММОЙ ПО ССЫЛКЕ, ВСЕ ДЕЙСТВИЯ, ПРЕДУСМОТРЕННЫЕ В ПОДПРОГРАММЕ ДЛЯ ФОРМАЛЬНОГО ПАРАМЕТРА, ПРОИЗВОДЯТСЯ С ИСПОЛЬЗОВАНИЕМ ПОЛЯ ФАКТИЧЕСКОГО ПАРАМЕТРА. ПОЭТОМУ ЛЮБОЕ ИЗМЕНЕНИЕ ФОРМАЛЬНОГО ПАРАМЕТРА ВЛЕЧЕТ ИЗМЕНЕНИЕ СООТВЕТСТВУЮЩЕГО ФАКТИЧЕСКОГО ПАРАМЕТРА. ПАРАМЕТРЫ, ПРИНИ-

МАЕМЫЕ ТАКИМ СПОСОБОМ, НАЗЫВАЮТ ПАРАМЕТРАМИ-ПЕРЕМЕННЫМИ. В СПИСКАХ ФОРМАЛЬНЫХ ПАРАМЕТРОВ ОПИСАНИЮ ПАРАМЕТРОВ-ПЕРЕМЕННЫХ ДОЛЖЕН ПРЕДШЕСТВОВАТЬ ПРЕФИКС VAR.

ПАРАМЕТР "B" В ПРОЦЕДУРЕ P ЯВЛЯЕТСЯ ПАРАМЕТРОМ ПЕРЕМЕННОЙ.

ФАКТИЧЕСКИЙ ПАРАМЕТР, СООТВЕТСТВУЮЩИЙ ФОРМАЛЬНОМУ ПАРАМЕТРУ ЗНАЧЕНИЕМ МОЖЕТ БЫТЬ КОНСТАНТОЙ, ПЕРЕМЕННОЙ, ВЫРАЖЕНИЕМ ( $X \leftarrow -7.2$ ,  $Y \leftarrow X/2$ ,  $I \leftarrow 2$ ). ФАКТИЧЕСКИЙ ПАРАМЕТР, СООТВЕТСТВУЮЩИЙ ФОРМАЛЬНОМУ ПАРАМЕТРУ ПЕРЕМЕННОЙ, ДОЛЖЕН БЫТЬ ПЕРЕМЕННОЙ ( $X \leftarrow X$ ).

ПОДПРОГРАММА В ПАСКАЛЕ МОЖЕТ БЫТЬ ТОЛЬКО ВНУТРЕННЕЙ, Т.Е. ОПИСАННОЙ В РАЗДЕЛЕ ПОДПРОГРАММ ТОГО БЛОКА, РАЗДЕЛ ОПЕРАТОРОВ КОТОРОГО СОДЕРЖИТ ЕЕ ВЫЗОВЫ. ОБОЗНАЧИМ ЧЕРЕЗ PR ПРОГРАММУ, PP1, PP2 - ПОДПРОГРАММЫ. ПУСТЬ ПРОГРАММА PR ИМЕЕТ ОДНУ ИЗ СТРУКТУР:



В СЛУЧАЕ (A) ОПИСАНИЕ ПОДПРОГРАММЫ PP1 РАЗМЕЩЕНО В БЛОКЕ ПРОГРАММЫ PR, ПОЭТОМУ ОБРАЩАТЬСЯ К ВЫПОЛНЕНИЮ ПОДПРОГРАММЫ PP1 МОЖНО ИЗ ЛЮБОГО МЕСТА РАЗДЕЛА ОПЕРАТОРОВ ПРОГРАММЫ.

В СЛУЧАЕ (B) ПОДПРОГРАММА PP1 ИСПОЛЬЗУЕТСЯ ТАК ЖЕ, КАК И В СЛУЧАЕ (A). НО БЛОК ПОДПРОГРАММЫ PP1 СОДЕРЖИТ ОПИСАНИЕ ПОДПРОГРАММЫ PP2. ПОДПРОГРАММА PP2 ДОСТУПНА ДЛЯ ВЫПОЛНЕНИЯ (ВИДИМА) ТОЛЬКО В РАЗДЕЛЕ ОПЕРАТОРОВ ПОДПРОГРАММЫ PP1 (ПОДПРОГРАММА PP2 ЛОКАЛИЗОВАНА В ПОДПРОГРАММЕ PP1). ОБРАЩЕНИЕ К ПОДПРОГРАММЕ PP2 ИЗ РАЗДЕЛА ОПЕРАТОРОВ ПРОГРАММЫ PR НЕДОПУСТИМО.

В СЛУЧАЕ (B) БЛОК ПРОГРАММЫ PR ВКЛЮЧАЕТ ОПИСАНИЕ ПОДПРОГРАММ PP1, PP2. ОБЕ ОНИ ДОСТУПНЫ ДЛЯ ВЫЗОВА В РАЗДЕЛЕ ОПЕРАТОРОВ ПРОГРАММЫ PR. КРОМЕ ТОГО, ПОДПРОГРАММЕ PP1 ДОСТУПНА ДЛЯ ОБРАЩЕНИЯ ПОДПРОГРАММА PP2, ОПИСАННАЯ ДО ПОДПРОГРАММЫ PP1.

ВСЕ ОБЪЕКТЫ, ОПИСАННЫЕ ИЛИ ОПРЕДЕЛЕННЫЕ В БЛОКЕ ПОДПРОГРАММЫ (МЕТКИ, ИДЕНТИФИКАТОРЫ КОНСТАНТ, ТИПЫ, ПЕРЕМЕННЫЕ, ПОДПРОГРАММЫ) ДОСТУПНЫ (ВИДИМЫ) ТОЛЬКО В ЭТОМ БЛОКЕ И НЕДОСТУПНЫ ВНЕ ЕГО - ЛОКАЛЬНЫЕ ОБЪЕКТЫ ПОДПРОГРАММЫ. ФОРМАЛЬНЫЕ ПАРАМЕТРЫ ТОЖЕ ЛОКАЛИЗОВАНЫ В БЛОКЕ ПОДПРОГРАММЫ (ОНИ ОПИСЫВАЮТСЯ В ЗАГОЛОВКЕ ПОДПРОГРАММЫ).

ПОДПРОГРАММА ВСЕГДА РАЗМЕЩЕНА В НЕКОТОРОМ ОБЪЕМЛЯЮЩЕМ ЕЕ БЛОКЕ И МОЖЕТ ИСПОЛЬЗОВАТЬ ИДЕНТИФИКАТОРЫ КОНСТАНТ, ТИПЫ, ПЕРЕМЕННЫЕ ЭТОГО БЛОКА - ГЛОБАЛЬНЫЕ ОБЪЕКТЫ ПОДПРОГРАММЫ.

ПРОЦЕДУРА

```
PROCEDURE PP (A,B:REAL; VAR Z:REAL);
```

```
  BEGIN Z:=(A+B)/2 END;
```

ПРЕДНАЗНАЧЕНА ДЛЯ ВЫЧИСЛЕНИЯ СРЕДНЕГО АРИФМЕТИЧЕСКОГО ДВУХ ВЕЩЕСТВЕННЫХ ЧИСЕЛ, ИСПОЛЬЗУЕТ ТОЛЬКО ФОРМАЛЬНЫЕ ПАРАМЕТРЫ.

ФУНКЦИИ

```
FUNCTION REGHT (X:REAL):REAL;
```

```
  BEGIN REGHT:=K*X+B END;
```

```
FUNCTION REGHT1 (X:REAL):REAL;
```

```
  VAR K,B:REAL;
```

```

BEGIN
K:=2.5; B:=-0.7;
REBNT1:=-K*X+B
END;

```

ПРЕДНАЗНАЧЕНЫ ДЛЯ ВЫЧИСЛЕНИЯ ОРДИНАТЫ ТОЧКИ, ЛЕЖАЩЕЙ НА ПРЯМОЙ  $Y=KX+B$ , ПО ЗНАЧЕНИЮ АБСЦИССЫ ТОЧКИ. ОБЕ ФУНКЦИИ ИСПОЛЗУЮТ ФОРМАЛЬНЫЙ ПАРАМЕТР ЗНАЧЕНИЕ X (АБСЦИССА ТОЧКИ). ФУНКЦИЯ REBNT ПОЛЗУЕТСЯ ПЕРЕМЕННЫМИ K,B, НЕ ОПИСАННЫМИ В НЕЙ - ГЛОБАЛЬНЫМИ ПЕРЕМЕННЫМИ. ФУНКЦИЯ REBNT1 ИСПОЛЗУЕТ ЛОКАЛЬНЫЕ ПЕРЕМЕННЫЕ K,B, ВЫЧИСЛЯЕТ ОРДИНАТУ ТОЧКИ ПРЯМОЙ  $Y=2,5X - 0,7$ .

ПОДПРОГРАММА ВЫПОЛНЯЕТСЯ ПРИ ЕЕ ВЫЗОВЕ. ВЫПОЛНЕНИЕ ПОСЛЕДНЕГО ОПЕРАТОРА ПОДПРОГРАММЫ СОПРОВОЖДАЕТСЯ ВОЗВРАТОМ В ТОЧКУ ВЫЗОВА: НА ОПЕРАТОР, СЛЕДУЮЩИЙ ЗА ОПЕРАТОРОМ ВЫЗОВА ПОДПРОГРАММЫ ПРОЦЕДУРЫ, ИЛИ В ВЫРАЖЕНИЕ, ЕСЛИ ЗАВЕРШЕНО ВЫПОЛНЕНИЕ ПОДПРОГРАММЫ-ФУНКЦИИ.

РАССМОТРИМ ПРОГРАММУ, КОТОРАЯ ИЩЕТ С ТОЧНОСТЬЮ  $\epsilon$  КОРНИ УРАВНЕНИЯ  $e^x \sin x = 0$  МЕТОДОМ ДЕЛЕНИЯ ПОПОЛАМ ОТРЕЗКА, ОТДЕЛЯЮЩЕГО КОРЕНЬ.

---

(\* ПРОГРАММА ПРЕДНАЗНАЧЕНА ДЛЯ ПОИСКА КОРНЕЙ ФУНКЦИИ  $Y=F(T)$  С ТОЧНОСТЬЮ EPS МЕТОДОМ ДЕЛЕНИЯ ОТРЕЗКА, ОТДЕЛЯЮЩЕГО КОРЕНЬ, ПОПОЛАМ \*)  
PROGRAM HALF;

```

VAR
EPS:REAL; (* ТРЕБУЕМАЯ ТОЧНОСТЬ *)
X:REAL; (* КОРЕНЬ *)
L,P:REAL; (* КОНЦЫ ОТРЕЗКА, ОТДЕЛЯЮЩЕГО КОРЕНЬ *)
ERR:BOOLEAN; (* ПРИЗНАК ОТСУТСТВИЯ КОРНЯ НА [L,P] *)
DTV:CHAR; (* ОТВЕТ НА ВОПРОС О ПРОДОЛЖЕНИИ
ПОИСКА КОРНЕЙ *)

```

```

(* ПРОЦЕДУРА УТОЧНЕНИЯ КОРНЯ *)
PROCEDURE S(A,B:REAL; (*КОНЦЫ ОТРЕЗКА*))
VAR X:REAL; (*КОРЕНЬ*)
VAR ER:BOOLEAN(*КОРНЯ НЕТ НА [A,B]*)
);

```

```

VAR
H, (*ДЛИНА ОТРЕЗКА*)
FA,FB,FX(*ЗНАЧЕНИЯ ФУНКЦИИ*)
:REAL;

```

```

(* ФУНКЦИЯ, ВЫЧИСЛЯЮЩАЯ F(T)*)
FUNCTION F(T:REAL):REAL;

```

```

BEGIN F:=EXP(T)-SIN(T) END;

```

```

BEGIN

```

```

FA:=F(A); FB:=F(B); H:=B-A; X:=A+H/2; ER:=FALSE;

```

```

IF FA*FB>0 THEN ER:=TRUE; (* КОРНЯ НЕТ НА [A,B] *)

```

```

WHILE (H>EPS) AND NOT ER DO

```

```

BEGIN

```

```

FX:=F(X);

```

```

IF FA*FX<=0 THEN BEGIN B:=X; FB:=FX END

```

```

ELSE BEGIN A:=X; FA:=FX END;

```

```

(* END IF *)

```

```

H:=B-A; X:=A+H/2

```

```

END; (* WHILE *)

```

```

END; (*PROCEDURE S *)

```

```

BEGIN
WRITE(#12); (* ОЧИСТКА ЭКРАНА *)
WRITELN(' ':10,'РЕШЕНИЕ УРАВНЕНИЯ EXP(T)-SIN(T)=0',
        ' С ТОЧНОСТЬЮ EPS');
WRITE('EPS=?'); READLN(EPS);
REPEAT
WRITE('УКАЖИТЕ ЛЕВЫЙ,ПРАВЫЙ КОНЦЫ ОТРЕЗКА, ОТДЕЛЯЮЩЕГО ',
      'КОРЕНЬ'); READLN(L,P);
S(L,P,X;ERR);
IF ERR THEN WRITELN('НА [',L,',',P,'] КОРЕНЬ НЕ НАЙДЕН')
ELSE WRITELN(X,'-КОРЕНЬ, РАСПОЛОЖЕННЫЙ НА [',
            L,',',P,']');
WRITELN('ПРОДОЛЖИТЬ ПОИСК КОРНЕЙ? Y/N'); READLN(OTV)
UNTIL OTV='N'
END. (* HALF *)

```

ПРОГРАММА HALF ИСПОЛЬЗУЕТ ПОДПРОГРАММУ S, КОТОРАЯ, КРОМЕ ФОРМАЛЬНЫХ ПАРАМЕТРОВ (A,B - ПАРАМЕТРЫ ЗНАЧЕНИЯ, X,ER - ПАРАМЕТРЫ ПЕРЕМЕННЫЕ), ЛОКАЛЬНЫХ ПЕРЕМЕННЫХ (H, FA, FB, FX), ЛОКАЛЬНОЙ ФУНКЦИИ F С ПАРАМЕТРОМ ЗНАЧЕНИЕМ T, ИСПОЛЬЗУЕТ ГЛОБАЛЬНУЮ ПЕРЕМЕННУЮ EPS.

В ПОДПРОГРАММАХ ФУНКЦИЯХ СИНТАКСИС ЯЗЫКА НЕ ЗАПРЕЩАЕТ ПРИСВАИВАТЬ ЗНАЧЕНИЯ ГЛОБАЛЬНЫМ ПЕРЕМЕННЫМ, ФОРМАЛЬНЫМ ПАРАМЕТРАМ ПЕРЕМЕННЫМ. ИСПОЛЬЗОВАНИЕ ЭТИХ ВОЗМОЖНОСТЕЙ ВЕДЕТ ПРИ ОБРАЩЕНИИ К ФУНКЦИИ, НАРЯДУ С ПОЛУЧЕНИЕМ ЗНАЧЕНИЯ ФУНКЦИИ, К ИЗМЕНЕНИЮ ЗНАЧЕНИЯ ГЛОБАЛЬНОЙ ПЕРЕМЕННОЙ ИЛИ ФАКТИЧЕСКОГО ПАРАМЕТРА, НАЗЫВАЕМОМУ ПОБОЧНЫМ ЭФФЕКТОМ ФУНКЦИИ. НАПРИМЕР, В ПРОГРАММЕ

```

PROGRAM SECONDARY_EFFECT;
VAR
K,P,V:REAL;
FUNCTION G(X:REAL):REAL;
BEGIN
K:=2.5;
G:=K*X+0.5;
END;
BEGIN
READLN(K,P);
V:=K+G(P)*(K-1);
WRITELN('K=',K,P,V)
END.

```

ПРИ ВЫЧИСЛЕНИИ ЗНАЧЕНИЯ V, ЕСЛИ ВВЕДЕНО K, РАВНОЕ 1, ПОСЛЕ ОБРАЩЕНИЯ К ФУНКЦИИ G K ИЗМЕНИТСЯ И БУДЕТ НАЙДЕНО  $V=1+G(P)*(2.5-1)$ .

ПОБОЧНЫЙ ЭФФЕКТ ЗАТРУДНЯЕТ ОТЛАДКУ ПРОГРАММ, ДОКАЗАТЕЛЬСТВО ПРАВИЛЬНОСТИ ИХ РАБОТЫ. РЕКОМЕНДУЕТСЯ ИЗБЕГАТЬ ИСПОЛЬЗОВАНИЯ В ФУНКЦИЯХ ФОРМАЛЬНЫХ ПАРАМЕТРОВ ПЕРЕМЕННЫХ, ПРИСВАИВАНИЯ ЗНАЧЕНИЯ ГЛОБАЛЬНЫМ ПЕРЕМЕННЫМ.

## 2.7. РЕКУРСИВНЫЕ ПОДПРОГРАММЫ

РЕКУРСИЕЙ НАЗЫВАЮТ СИТУАЦИЮ, КОГДА ПОДПРОГРАММА ВЫЗЫВАЕТ САМА СЕБЯ НЕПОСРЕДСТВЕННО ИЛИ ЧЕРЕЗ ДРУГИЕ ПОДПРОГРАММЫ. ПО СУТИ ЭТО ОЗНАЧАЕТ, ЧТО АЛГОРИТМ ПОДПРОГРАММЫ СВОДИТСЯ К ТАКОМУ ЖЕ АЛГОРИТМУ, НО С ДРУГИМИ ИСХОДНЫМИ ДАННЫМИ. НАПРИМЕР,  $N!=N*(N-1)!$ ,

ЕСЛИ  $N > 2$  И  $N! = 1$ , ЕСЛИ  $N < 2$ ; ДЛЯ ВЫЧИСЛЕНИЯ  $S = \sum_{i=1}^M f(i)$  МОЖНО  
 ИСПОЛЬЗОВАТЬ АЛГОРИТМ  $S = f(M) + \sum_{i=1}^{M-1} f(i)$ , ЕСЛИ

$M > 0$  И  $S = 0$ , ЕСЛИ  $M \leq 0$ .

ПРИ ТРАНСЛЯЦИИ ПРОГРАММЫ С РЕКУРСИВНЫМИ ПОДПРОГРАММАМИ ОПЦИЯ "A" ТРАНСЛЯТОРА ДОЛЖНА БЫТЬ ОТКЛЮЧЕНА: (\*\*A-\*) .

НАПРИМЕР. ПРОГРАММА

(\*\*A-\*)

PROGRAM SKL;

VAR

K, L: INTEGER;

C: INTEGER; (\*ЧИСЛО СОЧЕТАНИЙ ИЗ K ПО L\*)

FUNCTION FACTORIAL(N: INTEGER): INTEGER;

BEGIN

IF N < 2 THEN FACTORIAL := 1

ELSE FACTORIAL := N \* FACTORIAL(N-1)

END;

BEGIN

READLN(K, L);

WRITELN('ЧИСЛО СОЧЕТАНИЙ ИЗ ' , K, ' ПО ' , L, ' РАВНО ' ,

(FACTORIAL(K) DIV FACTORIAL(L)) DIV

FACTORIAL(K-L))

END.

ВЫЧИСЛЯЕТ И ВЫДАЕТ ЗНАЧЕНИЕ  $C_k^L = K! / (L! \cdot (K-L)!)$  С ИСПОЛЬЗОВАНИЕМ РЕКУРСИВНОЙ ФУНКЦИИ FACTORIAL, ВЫЧИСЛЯЮЩЕЙ  $N!$ .

ПРИ СОСТАВЛЕНИИ РЕКУРСИВНОЙ ПОДПРОГРАММЫ СЛЕДУЕТ ЗАБОТИТЬСЯ, ЧТОБЫ ОНА НЕ ВЫЗЫВАЛА СЕБЯ БЕСКОНЕЧНО. В НАШЕМ ПРИМЕРЕ ВЫХОД ИЗ РЕКУРСИИ ДЛЯ ФУНКЦИИ FACTORIAL ЗАДАЕТ ВЫПОЛНЕНИЕ УСЛОВИЯ N < 2.

РЕАЛИЗАЦИЯ РЕКУРСИВНЫХ ПОДПРОГРАММ СВЯЗАНА С БОЛЬШИМИ ТРУДНОСТЯМИ. ПОЭТОМУ РЕКУРСИЕЙ СЛЕДУЕТ ПОЛЬЗОВАТЬСЯ ОСТОРОЖНО И ТАМ, ГДЕ ЭТО НЕОБХОДИМО.

В ПАСКАЛЕ ИМЕЕТСЯ ПРАВИЛО: ПОДПРОГРАММА ДОЛЖНА БЫТЬ ОПИСАНА ДО ОБРАЩЕНИЯ К НЕЙ. В СЛУЧАЕ КОСВЕННОЙ РЕКУРСИИ, КОГДА ПОДПРОГРАММА А ОБРАЩАЕТСЯ К ПОДПРОГРАММЕ В, А ТА ВЫЗЫВАЕТ А, СОБЛИСТИ ЭТО ПРАВИЛО НЕВОЗМОЖНО. ВЫХОДОМ ИЗ ЭТОЙ СИТУАЦИИ ЯВЛЯЕТСЯ ИСПОЛЬЗОВАНИЕ ПРЕДВАРИТЕЛЬНОГО ОПИСАНИЯ ПОДПРОГРАММЫ, КОТОРОЕ СОДЕРЖИТ ТОЛЬКО ЗАГОЛОВОК, А ВМЕСТО БЛОКА - СЛУЖЕБНОЕ СЛОВО FORWARD. ПРЕДВАРИТЕЛЬНОЕ ОПИСАНИЕ ПОДПРОГРАММЫ РАЗМЕЩАЕТСЯ ДО ОПИСАНИЙ ВСЕХ ПОДПРОГРАММ, СОДЕРЖАЩИХ ОБРАЩЕНИЯ К НЕЙ. ФАКТИЧЕСКОЕ ОПИСАНИЕ ПОДПРОГРАММЫ СЛЕДУЕТ В ТОМ ЖЕ РАЗДЕЛЕ ПОСЛЕ ОПИСАНИЙ ВСЕХ ПОДПРОГРАММ, К КОТОРЫМ ОНА ОБРАЩАЕТСЯ, И ОТЛИЧАЕТСЯ ОТ ОБЫЧНОГО ОПИСАНИЯ ЛИШЬ ТЕМ, ЧТО ЕЕ ЗАГОЛОВОК ВКЛЮЧАЕТ ТОЛЬКО ИМЯ ПОДПРОГРАММЫ (ОТЛОЖЕННОЕ ОПИСАНИЕ ПОДПРОГРАММЫ).

НАПРИМЕР, ЕСЛИ ПРИ ВЫПОЛНЕНИИ ПРОГРАММЫ

PROGRAM SATCH;

VAR

D: INTEGER;

FUNCTION UP(VAR I: INTEGER): INTEGER;

FORWARD; (\*ПРЕДВАРИТЕЛЬНОЕ ОПИСАНИЕ ФУНКЦИИ \*)

FUNCTION DOWN(VAR I: INTEGER): INTEGER;

BEGIN

I := I DIV 2; WRITELN(I);

IF I <> 1 THEN I := UP(I)

END;

FUNCTION UP; (\* ОТЛОЖЕННОЕ ОПИСАНИЕ ФУНКЦИИ \*)

```
BEGIN
  WHILE I MOD 2<>0 DO
    BEGIN I:=I*3+1; WRITELN(I) END;
  I:=DOWN(I)
END;
BEGIN
  WRITE('ВВЕДИТЕ ЦЕЛОЕ ЧИСЛО ');
  READLN(D); WRITELN('ИСХОДНОЕ ЧИСЛО:',D);
  D:=UP(D);
  WRITELN('КОНЕЦ')
END.
```

ВВЕДЕНО ЧИСЛО 5, ТО ВЫДАЧА ИМЕЕТ ВИД:  
ИСХОДНОЕ ЧИСЛО: 5

```
16
8
4
2
1
КОНЕЦ
```

## 2.8. ЗАДАНИЯ ПО ТЕМЕ

1. ТЕКСТ НА РУССКОМ ЯЗЫКЕ СОСТОИТ ИЗ СЛОВ, РАЗДЕЛЕННЫХ ОДНИМ ИЛИ НЕСКОЛЬКИМИ ПРОБЕЛАМИ, И ЗНАКОВ ПРЕПИНАНИЯ; РАЗМЕЩЕН В ОДНОЙ ИЛИ НЕСКОЛЬКИХ СТРОКАХ. ПЕРЕНОС СЛОВ ЗАПРЕЩЕН.

ВЫДЕЛИТЬ И ВЫДАТЬ ВСЕ СЛОВА ТЕКСТА,

- 1.1) НАЧИНАЮЩИЕСЯ С ЗАДАННОЙ БУКВЫ,
- 1.2) СОДЕРЖАЩИЕ ЗАДАННУЮ БУКВУ,
- 1.3) ОКАНЧИВАЮЩИЕСЯ НА ЗАДАННУЮ БУКВУ,
- 1.4) СОДЕРЖАЩИЕ СДВОЕННЫЕ БУКВЫ,
- 1.5) ЯВЛЯЮЩИЕСЯ ПОЛИНДРОМАМИ ("ПЕРЕВЕРТЫШАМИ") (ДЕД, ТОПОТ) ОДНОСИМВОЛЬНЫЕ СЛОВА "ПЕРЕВЕРТЫШАМИ" НЕ СЧИТАТЬ).
- 1.6) НАЙТИ СЛОВО С МАКСИМАЛЬНОЙ ДЛИНОЙ.

ПЕРЕКОДИРОВАТЬ ТЕКСТ,

- 1.7) ИЗМЕНЯЯ В КАЖДОМ ЕГО СЛОВЕ ПОРЯДОК БУКВ НА ПРОТИВОПОЛОЖНЫЙ,
- 1.8) ЗАМЕНЯЯ ЗВОНКИЕ СОГЛАСНЫЕ НА ПАРНЫЕ ГЛУХИЕ И НАОБОРОТ.

2. ТЕКСТ НА РУССКОМ ЯЗЫКЕ СОСТОИТ ИЗ СЛОВ, РАЗДЕЛЕННЫХ ОДНИМ ИЛИ НЕСКОЛЬКИМИ ПРОБЕЛАМИ, И ЗНАКОВ ПРЕПИНАНИЯ; РАЗМЕЩЕН В ОДНОЙ СТРОКЕ.

ВЫДЕЛИТЬ И ВЫДАТЬ ПАРЫ СЛОВ ТЕКСТА,

- 2.1) ОТЛИЧАЮЩИХСЯ ТОЛЬКО ПОРЯДКОМ БУКВ,
- 2.2) ОДНО ИЗ КОТОРЫХ ПОЛУЧАЕТСЯ ИЗ ДРУГОГО УДАЛЕНИЕМ ОДНОЙ ИЗ БУКВ (КЛУБОК-КУБОК, БЛУЗА-ЛУЗА).
- 2.3) НАЙТИ И ВЫДАТЬ ВСЕ СЛОВА МУТАНТЫ. ДВА СЛОВА ТЕКСТА ОБРАЗУЮТ СЛОВО МУТАНТ, ЕСЛИ ПОСЛЕДНИЕ БУКВЫ ПЕРВОГО СОВПАДАЮТ С ПЕРВЫМИ БУКВАМИ ВТОРОГО. В МУТАНТ ОБЩАЯ ЧАСТЬ СЛОВ ВКЛЮЧАЕТСЯ ОДИН РАЗ (БУИВОЛ+ВОЛК—>БУИВОЛК; КРОКОДИЛ+ЛОШАДЬ-->КРОКОДИЛОШАДЬ).

3. СТРОКА СОДЕРЖИТ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ, ПРЕДСТАВЛЯЮЩУЮ ЧИСЛО В РИМСКОЙ СИСТЕМЕ СЧИСЛЕНИЯ (I(1), V(5), X(10), L(50), C(100), D(500), M(1000)).

СОФОРМИРОВАТЬ ВНУТРИМАШИННОЕ ПРЕДСТАВЛЕНИЕ ЧИСЛА (INTEGER).  
ВЫДАТЬ ЧИСЛО.

4. СТРОКА СОДЕРЖИТ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ, ИЗБРАЖАЮЩИХ ДЕСЯТИЧНОЕ ЧИСЛО С ПЛАВАЮЩЕЙ ТОЧКОЙ (1ЕЗ, -4.7Е-5, 18Е+9). ПОЛУЧИТЬ ВНУТРИНАШИННОЕ ПРЕДСТАВЛЕНИЕ ЧИСЛА (REAL). ВЫДАТЬ ЧИСЛО.

5. ДАН ТЕКСТ, РАЗМЕЩЕННЫЙ В ОДНОЙ ИЛИ НЕСКОЛЬКИХ СТРОКАХ ТЕКСТА. ИСКЛЮЧИТЬ ИЗ ТЕКСТА КОММЕНТАРИИ (ПАРА ЛИТЕР "(" - НАЧАЛО КОММЕНТАРИЯ, ")" - КОНЕЦ КОММЕНТАРИЯ).

6. ПОСЛАНИЕ ОТ ВНЕЗЕМНОЙ ЦИВИЛИЗАЦИИ ПРЕДСТАВЛЯЕТ СОБОЙ НАБОР N СИМВОЛОВ "0" И "1". ПРИ ПОПЫТКЕ РАСШИФРОВАТЬ ЕГО ОБНАРУЖЕНО, ЧТО N ЯВЛЯЕТСЯ ПРОИЗВЕДЕНИЕМ ДВУХ ПРОСТЫХ ЧИСЕЛ K И L. УЧЕНЫЕ ПРЕДПОЛАГАЮТ, ЧТО ПОСЛАНИЕ - КАРТИНКА РАЗМЕРОВ K\*L ИЛИ L\*K.

ОПРЕДЕЛИТЬ N, K, L. ПЕРЕКОДИРОВАТЬ ПОСЛАНИЕ, ЗАМЕНЯЯ

А) "0" НА "\_", "1" НА "\*";

Б) "0" НА "\*", "1" НА "\_".

ВЫДАТЬ КАРТИНКИ РАЗМЕРОВ

А) K\*L,

Б) L\*K

С ВЫБОРОМ СИМВОЛОВ ИЗ ПЕРЕКОДИРОВАННОГО ПОСЛАНИЯ ПО

А) СТРОКАМ,

Б) СТОЛБЦАМ.

ПОСЛАНИЕ: 10010010111101001100011100100000

11010010001010010010111

7. ФУНКЦИЯ БЕССЕЛЯ  $I_{N+0,5}(z)$  ДЛЯ ВЕЩЕСТВЕННЫХ  $z$  И ЦЕЛЫХ ПОЛОЖИТЕЛЬНЫХ  $N$  ОПРЕДЕЛЯЕТСЯ РЕКУРРЕНТНЫМ СООТНОШЕНИЕМ:

$$z \cdot I_{N+0,5}(z) + z \cdot I_{N-1,5}(z) = z \cdot (N-0,5) \cdot I_{N-0,5}(z);$$

$$I_{0,5}(z) = \sin z \cdot \sqrt{\frac{z}{\pi}}, \quad I_{-0,5}(z) = \cos z \cdot \sqrt{\frac{z}{\pi}}.$$

СОСТАВИТЬ ПРОГРАММУ ВЫЧИСЛЕНИЯ ТАБЛИЦЫ ЗНАЧЕНИЯ ФУНКЦИИ

$I_{N+0,5}(z)$  НА  $[a, b]$  С ШАГОМ  $h = (b-a)/M$  С ИСПОЛЬЗОВАНИЕМ

РЕКУРСИВНОЙ ФУНКЦИИ, ВЫЧИСЛЯЮЩЕЙ ЗНАЧЕНИЕ  $I_{k+0,5}(z)$ .

### 3. ИНСТРУКЦИЯ ПО РАБОТЕ В СРЕДЕ ТУРБО ПАСКАЛЬ

СИСТЕМА ТУРБО ПАСКАЛЬ ПРИМЕНЯЕТСЯ В СОЧЕТАНИИ С ОПЕРАЦИОННОЙ СИСТЕМОЙ SCP НА ПЭВМ ROBOTRON-1715. ОНА ЯВЛЯЕТСЯ ЭФФЕКТИВНЫМ ИНСТРУМЕНТОМ, ПРЕДУСМАТРИВАЮЩИМ КОМПИЛЯЦИЮ В ТЕЧЕНИЕ СЕКУНД. СИСТЕМА ВКЛЮЧАЕТ ТРАНСЛЯТОР С ЯЗЫКА ПАСКАЛЬ, ВСТРОЕННЫЙ ЭКРАННЫЙ РЕДАКТОР, СЕРВИСНЫЕ И ПРИКЛАДНЫЕ ПРОГРАММЫ. ТРАНСЛЯТОР РЕАЛИЗУЕТ СТАНДАРТ ЯЗЫКА С НЕБОЛЬШИМИ ОТЛИЧИЯМИ И ПРЕДОСТАВЛЯЕТ РЯД ДОПОЛНИТЕЛЬНЫХ ВОЗМОЖНОСТЕЙ. ПРОГРАММЫ СИСТЕМЫ ХРАНЯТСЯ В ДИСКОВЫХ ФАЙЛАХ.

ИМЯ ФАЙЛА В СИСТЕМЕ SCP - ПОСЛЕДОВАТЕЛЬНОСТЬ НЕ БОЛЕЕ В БУКВ И ЦИФР, НАЧИНАЮЩАЯСЯ С БУКВЫ. ОНО ДОПУСКАЕТ НЕ БОЛЕЕ ЧЕМ ТРЕХСИМВОЛЬНОЕ РАСШИРЕНИЕ - ТИП ФАЙЛА. PAS - РАСШИРЕНИЕ ДЛЯ ИМЕН ФАЙЛОВ, ХРАНЯЩИХ ТЕКСТЫ ПАСКАЛЬ-ПРОГРАММ (НАПРИМЕР, PROB1.PAS).

ОБРАБАТЫВАЕМАЯ СИСТЕМОЙ ПРОГРАММА ДОЛЖНА РАЗМЕЩАТЬСЯ В РАБОЧЕМ ФАЙЛЕ. ДЛЯ РАБОЧЕГО ФАЙЛА ВЫДЕЛЯЕТСЯ ОБЛАСТЬ ОСНОВНОЙ ПАМЯТИ ПЭВМ. ПРОГРАММА МОЖЕТ ПОСТУПИТЬ В НЕГО С ДИСКА, ЕСЛИ РАНЕЕ БЫЛА ПОМЕЩЕНА В ДИСКОВЫЙ ФАЙЛ, ИЛИ НАБИРАЕТСЯ ПОЛЬЗОВАТЕЛЕМ С ИС-

ПОЛЬЗОВАНИЕМ ЭКРАННОГО РЕДАКТОРА. РАБОТА С ПРОГРАММОЙ ИЗ РАБОЧЕГО ФАЙЛА ВКЛЮЧАЕТ ДВА ЭТАПА - ТРАНСЛЯЦИЮ И ВЫПОЛНЕНИЕ.

СОДЕРЖИМОЕ РАБОЧЕГО ФАЙЛА МОЖЕТ БЫТЬ ЗАПИСАНО В ДИСКОВЫЙ ФАЙЛ. ПРИ ЭТОМ, ЕСЛИ НА ДИСКЕ УЖЕ ЕСТЬ ФАЙЛ, ИМЯ КОТОРОГО СОВПАДАЕТ С ИМЕНЕМ РАБОЧЕГО ФАЙЛА, РАСШИРЕНИЕ ИМЕНИ СТАРОЙ КОПИИ ФАЙЛА ЗАМЕНЯЕТСЯ НА ВАР.

ДЛЯ РАБОТЫ В СИСТЕМЕ ТУРБО ПАСКАЛЬ СЛЕДУЕТ ПОЛЬЗОВАТЬСЯ ДИСКОМ, ХРАНЯЩИМ НЕОБХОДИМЫЕ ПРОГРАММНЫЕ КОМПОНЕНТЫ СИСТЕМЫ. ТИПИЧНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ДЕЙСТВИЙ ТАКОВА:

1. ВКЛЮЧАЕМ ПЭВМ.
2. УСТАНОВЛИВАЕМ ДИСК НА ДИСКОВОД А. МАШИНА АВТОМАТИЧЕСКИ ОСУЩЕСТВЛЯЕТ НАЧАЛЬНУЮ ЗАГРУЗКУ ОПЕРАЦИОННОЙ СИСТЕМЫ SCP.
3. ЧТОБЫ СДЕЛАТЬ ДОСТУПНОЙ СИСТЕМУ ТУРБО ПАСКАЛЬ, ВЫЗЫВАЕМ ПО КОМАНДЕ TURBO (ОПЕРАЦИОННОЙ СИСТЕМЫ SCP) В ОСНОВНУЮ ПАМЯТЬ ЕЕ КОМПОНЕНТЫ (ТРАНСЛЯТОР, ВСТРОЕННЫЙ РЕДАКТОР). КОМАНДУ НАБИРАЕМ С ПОЗИЦИИ КУРСОРА ПОСЛЕ СИМВОЛОВ "A": A>TURBO. НАЖАТИЕМ КЛАВИШИ <ET> (ENTER) ПЕРЕДАЕМ КОМАНДУ НА ВЫПОЛНЕНИЕ.
4. СИСТЕМА ВЫДАЕТ НА ЭКРАН ВОПРОС INCLUDE ERROR MESSAGES (Y/N)? (ДОЗАГРУЖАТЬ В ОСНОВНУЮ ПАМЯТЬ ТЕКСТОВЫЙ ФАЙЛ С ДИАГНОСТИЧЕСКИМИ СООБЩЕНИЯМИ?). НА НЕГО, КАК ПРАВИЛО, ОТВЕЧАЕМ ПОЛОЖИТЕЛЬНО - "Y".
5. НА ЭКРАН ВЫДАЕТСЯ ГЛАВНОЕ МЕНЮ СИСТЕМЫ ТУРБО ПАСКАЛЬ, СОДЕРЖАЩЕЕ КОМАНДЫ, КОТОРЫЕ МОЖЕТ ВЫПОЛНЯТЬ СИСТЕМА. ВЫПОЛНЕНИЕ КОМАНДЫ ЗАДАЕМ НАЖАТИЕМ В ПОЗИЦИИ КУРСОРА НА НИЖНЕМ РЕГИСТРЕ ЛАТИНСКОЙ БУКВЫ, ПОДЧЕРКНУТОЙ В НАЗВАНИИ КОМАНДЫ.

```
LOGGED DRIVE: A
WORK FILE:
MAIN FILE:
EDIT COMPILE RUN SAVE
EXECUTE DIR QUIT COMPILER OPTIONS
>
```

#### 5.1. L (LOGGED DRIVE)

- СМЕНИТЬ АКТИВНЫЙ ДИСКОВОД (ВЫПОЛНЯЕТСЯ ПРИ СМЕНЕ АКТИВНОГО ДИСКОВОДА, ПРИ ЗАМЕНЕ ДИСКА НА АКТИВНОМ ДИСКОВОДЕ).

#### 5.2. W (WORK FILE)

- УСТАНОВИТЬ РАБОЧИЙ ФАЙЛ. КОМАНДА ЗАПРАШИВАЕТ ИМЯ РАБОЧЕГО ФАЙЛА:

WORK FILE NAME: -

С ПОЗИЦИИ КУРСОРА НАБИРАЕМ ИМЯ ФАЙЛА, ЕСЛИ ОН ЕСТЬ НА ДИСКЕ, ИЛИ КОТОРЫМ ХОТИМ НАЗВАТЬ ВНОВЬ НАБИРАЕМЫЙ ФАЙЛ, И НАЖИМАЕМ КЛАВИШУ <ET>. ЕСЛИ ФАЙЛ С УКАЗАННЫМ ИМЕНЕМ ИМЕЕТСЯ НА ДИСКЕ, ОН ВЫЗЫВАЕТСЯ В РАБОЧИЙ ФАЙЛ, ИНАЧЕ СИСТЕМА ВЫДАЕТ ИНФОРМАЦИОННОЕ СООБЩЕНИЕ

NEW FILE.

#### 5.3. E (EDIT)

- НАЧАТЬ РЕДАКТИРОВАНИЕ РАБОЧЕГО ФАЙЛА.

РАБОТОЙ ЭКРАННОГО РЕДАКТОРА УПРАВЛЯЕМ С ПОМОЩЬЮ ЕГО КОМАНД. КОМАНДА ЗАДАЕТСЯ НАЖАТИЕМ СООТВЕТСТВУЮЩЕЙ БУКВЕННОЙ КЛАВИШИ НА НИЖНЕМ РЕГИСТРЕ ПРИ УТОПЛЕННОЙ КЛАВИШЕ <CTRL>. НАПРИМЕР, УДАЛИТЬ

СИМВОЛ ЭКРАНА, ПОДСВЕЧЕННЫЙ КУРСОРОМ, МОЖНО ПО КОМАНДЕ В. НАБОР СИМВОЛА ПРИ УТОПЛЕННОЙ КЛАВИШЕ <CTRL> УСЛОВНО ОБОЗНАЧАЮТ ^В.

РЕДАКТОР МОЖЕТ РАБОТАТЬ В ОДНОМ ИЗ ДВУХ РЕЖИМОВ: INSERT (ПОМЕСТИТЬ ВНОВЬ ПОСТУПАЮЩИЙ С КЛАВИАТУРЫ СИМВОЛ В ПОЗИЦИЮ ЭКРАНА, ПОДСВЕЧЕННУЮ КУРСОРОМ, С ПРЕДВАРИТЕЛЬНЫМ СДВИГОМ ВПРАВО НА 1 ПОЗИЦИЮ ЧАСТИ СТРОКИ, НАЧИНАЮЩЕЙСЯ С ПОДСВЕЧЕННОГО СИМВОЛА - ВСТАВИТЬ СИМВОЛ В СТРОКУ) ИЛИ OVERWRITE (ВНОВЬ ПОСТУПАЮЩИЙ СИМВОЛ ПОМЕСТИТЬ В ПОЗИЦИЮ СТРОКИ, ПОДСВЕЧЕННУЮ КУРСОРОМ - ЗАМЕНИТЬ СИМВОЛ). ПЕРЕКЛЮЧЕНИЕ РЕЖИМА ПРОИЗВОДИТСЯ ПО КОМАНДЕ ^V.

ВО ВРЕМЯ РАБОТЫ РЕДАКТОРА ПЕРВАЯ СТРОКА ЭКРАНА ХРАНИТ СЛУЖЕБНУЮ ИНФОРМАЦИЮ:

LINE M COL N РЕЖИМ INDENT A: ИМЯ ФАЙЛА.

ЗДЕСЬ M - НОМЕР СТРОКИ, N - НОМЕР ПОЗИЦИИ В СТРОКЕ ЭКРАНА, ГДЕ РАЗМЕЩЕН КУРСОР.

ЕСЛИ РАБОЧИЙ ФАЙЛ ПУСТ, РЕДАКТОР НАЧИНАЕТ РАБОТУ С ОЧИСТКИ ЭКРАНА ДЛЯ ЗАНЕСЕНИЯ В ЕГО СТРОКИ ВНОВЬ НАБИРАЕМОЙ ПРОГРАММЫ. НАБИРАЕМ ТЕКСТ ПРОГРАММЫ С ЗАВЕРШЕНИЕМ НАБОРА КАЖДОЙ СТРОКИ НАЖАТИЕМ КЛАВИШИ <ET>. ДЛЯ УДАЛЕНИЯ В НАБИРАЕМОЙ СТРОКЕ ПОСЛЕДНЕГО НЕВЕРНОГО СИМВОЛА МОЖЕМ ПОЛЬЗОВАТЬСЯ НЕ ТОЛЬКО КОМАНДОЙ "^^B", НО И КЛАВИШЕЙ <DEL> (СДВИГ НА ОДНУ ПОЗИЦИЮ ВЛЕВО С УНИЧТОЖЕНИЕМ СИМВОЛА).

КОНЕЦ НАБОРА ТЕКСТА ДОЛЖЕН СОПРОВОЖДАТЬСЯ ВЫХОДОМ ИЗ РЕДАКТОРА В ГЛАВНОЕ МЕНЮ СИСТЕМЫ ПО КОМАНДЕ ^KD.

ЕСЛИ РЕДАКТИРУЕМ СТАРЫЙ ФАЙЛ, РЕДАКТОР ВЫЗЫВАЕТ НА ЭКРАН ЕГО ПЕРВУЮ СТРАНИЦУ. ВЫПОЛНЯЕМ РЕДАКТИРОВАНИЕ В ПРЕДЕЛАХ СТРАНИЦЫ. ДЛЯ ЛИСТАНИЯ РАБОЧЕГО ФАЙЛА ИСПОЛЬЗУЕМ КОМАНДЫ ^С (ЛИСТАНИЕ ВПЕРЕД) ИЛИ ^R (ЛИСТАНИЕ НАЗАД К НАЧАЛУ ФАЙЛА). ЗАВЕРШАЕМ РЕДАКТИРОВАНИЕ ФАЙЛА ВЫХОДОМ ИЗ РЕДАКТОРА ПО КОМАНДЕ ^KD.

НЕКОТОРЫЕ КОМАНДЫ РЕДАКТОРА:

^V СМЕНИТЬ РЕЖИМ (ВСТАВКА/ЗАМЕНА СИМВОЛА);

^^B УДАЛИТЬ СИМВОЛ, ПОДСВЕЧЕННЫЙ КУРСОРОМ;

^N РАЗДЕЛИТЬ СТРОКУ НА 2 (ЧАСТЬ СТРОКИ, НАХОДЯЩУЮСЯ СПРАВА ОТ КУРСОРА, ПЕРЕНЕСТИ НА НОВУЮ СТРОКУ. ЕСЛИ КУРСОР УКАЗЫВАЕТ НА ПОСЛЕДНИЙ СИМВОЛ СТРОКИ, КОМАНДА ВКЛЮЧАЕТ В ТЕКСТ НОВУЮ ПУСТУЮ СТРОКУ);

^Y УДАЛИТЬ СТРОКУ, В КОТОРОЙ РАЗМЕЩЕН КУРСОР;

^C ПЕРЕЙТИ НА СЛЕДУЮЩУЮ СТРАНИЦУ;

^R ПЕРЕЙТИ НА ПРЕДЫДУЩУЮ СТРАНИЦУ.

СМЕШАТЬ КУРСОР ПО ЭКРАНУ МОЖНО КОМАНДАМИ РЕДАКТОРА ИЛИ КЛАВИШАМИ УПРАВЛЕНИЯ ПЕРЕМЕЩЕНИЕМ КУРСОРА

(^D ИЛИ ---, ^^B ИЛИ <---, ^E ИЛИ ^, ^X ИЛИ ^).

^KD ВЫХОД ИЗ РЕДАКТОРА В ГЛАВНОЕ МЕНЮ.

ВЫХОД ИЗ РЕДАКТОРА СОПРОВОЖДАЕТСЯ ВЫДАЧЕЙ НА ЭКРАНЕ

>..

ДЛЯ ВЫЗОВА НА ЭКРАН ГЛАВНОГО МЕНЮ СИСТЕМЫ ДОСТАТОЧНО НАБРАТЬ ЛЮБОЙ СИМВОЛ, НЕ СОВПАДАЮЩИЙ НИ С ОДНИМ ИЗ СИМВОЛОВ - КОМАНД СИСТЕМЫ, НАПРИМЕР, СИМВОЛ ПРОБЕЛ (КЛАВИША <SPACE>).

5.4. С (COMPILE)

- НАЧАТЬ ТРАНСЛЯЦИЮ ПАСКАЛЬ-ПРОГРАММЫ, РАЗМЕЩЕННОЙ В РАБОЧЕМ ФАЙЛЕ.

ПРИ УСПЕШНОМ ЗАВЕРШЕНИИ ТРАНСЛЯЦИИ НА ЭКРАН ВЫДАЕТСЯ ИНФОРМАЦИЯ О КОЛИЧЕСТВЕ СТРОК В ПРОГРАММЕ, ЕЕ ДЛИНЕ, МЕСТЕ В ОСНОВНОЙ ПАМЯТИ, ОСУЩЕСТВЛЯЕТСЯ ВЫХОД В ГЛАВНОЕ МЕНЮ С ВЫДАЧЕЙ

" > .."

ОБНАРУЖЕНИЕ ПЕРВОЙ СИНТАКСИЧЕСКОЙ ОШИБКИ ВЫЗЫВАЕТ ВЫДАЧУ НА ЭКРАН СООБЩЕНИЯ ОБ ОШИБКЕ С ПРИГЛАШЕНИЕМ НАЖАТЬ КЛАВИШУ <ESC>.

НАЖАТИЕ КЛАВИШИ <ESC> ВЕДЕТ К ВЫЗОВУ ЭКРАННОГО РЕДАКТОРА, КОТОРЫЙ ВЫДАЕТ НА ЭКРАН ЧАСТЬ РАБОЧЕГО ФАЙЛА, СОДЕРЖАЩУЮ ОШИБКУ. ПРИ ЭТОМ КУРСОР (ОБЫЧНО) РАЗМЕЩАЕТСЯ ПРАВЕЕ ОШИБКИ. ФАЙЛ ПОДЛЕЖИТ РЕДАКТИРОВАНИЮ (СМ. 5.3). ПОСЛЕ ВЫХОДА ИЗ РЕДАКТОРА РАБОЧИЙ ФАЙЛ МОЖНО СНОВА ПЕРЕДАТЬ НА ТРАНСЛЯЦИЮ.

#### 5.5. R (RUN)

- ВЫПОЛНИТЬ ПРОГРАММУ, ПОЛУЧЕННУЮ ПОСЛЕ ТРАНСЛЯЦИИ РАБОЧЕГО ФАЙЛА.

НА ЭКРАН ВЫДАЕТСЯ "RUNNING", НАЧИНАЕТСЯ ВЫПОЛНЕНИЕ ПРОГРАММЫ. ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ НА ЭТАПЕ ВЫПОЛНЕНИЯ ПРОГРАММЫ ОПРЕДЕЛЯЮТСЯ ЭТОЙ ПРОГРАММОЙ.

УСПЕШНОЕ ЗАВЕРШЕНИЕ ВЫПОЛНЕНИЯ ПРОГРАММЫ СОПРОВОЖДАЕТСЯ ВЫХОДОМ В ГЛАВНОЕ МЕНЮ СИСТЕМЫ (">\_").

ДЛЯ ВЫХОДА ИЗ ПРОГРАММЫ В ГЛАВНОЕ МЕНЮ (ПРИ ЗАЦИКЛИВАНИИ ПРОГРАММЫ) СЛЕДУЕТ ИСПОЛЬЗОВАТЬ КОМАНДУ ^S^C.

ОБНАРУЖЕНИЕ ОШИБКИ ВРЕМЕНИ ВЫПОЛНЕНИЯ ПРОГРАММЫ ВЫЗЫВАЕТ АВАРИЙНОЕ СНЯТИЕ ПРОГРАММЫ (PROGRAM ABORTED), ВЫДАЧУ СООБЩЕНИЯ ОБ ОШИБКЕ И В БОЛЬШИНСТВЕ СЛУЧАЕВ ПРИГЛАШЕНИЯ НАЖАТЬ КЛАВИШУ <ESC>. НАЖАТИЕ КЛАВИШИ <ESC>, КАК И В СЛУЧАЕ ТРАНСЛЯЦИИ, ВЫЗЫВАЕТ ВХОД В ЭКРАННЫЙ РЕДАКТОР С ВЫДАЧЕЙ СТРАНИЦЫ ПРОГРАММЫ, СОДЕРЖАЩЕЙ ОШИБКУ.

#### 5.6. S (SAVE)

- ЗАПИСАТЬ РАБОЧИЙ ФАЙЛ В ФАЙЛ НА ДИСКЕ С ИМЕНЕМ РАБОЧЕГО ФАЙЛА.

5.7. ПЕРЕХОД К ОБРАБОТКЕ НОВОЙ ПРОГРАММЫ ТРЕБУЕТ УКАЗАНИЯ НОВОГО РАБОЧЕГО ФАЙЛА (КОМАНДА W). ЕСЛИ ТЕКУЩИЙ РАБОЧИЙ ФАЙЛ НЕ БЫЛ ЗАПИСАН НА ДИСК, СИСТЕМА ЗАПРАШИВАЕТ НАДО ЛИ СОХРАНИТЬ ЕГО НА ДИСКЕ.

### 6. ДОПОЛНИТЕЛЬНЫЕ КОМАНДЫ СИСТЕМЫ.

#### 6.1. D (DIR)

- ВЫДАТЬ НА ЭКРАН КАТАЛОГ ДИСКОВЫХ ФАЙЛОВ И ИНФОРМАЦИЮ ОБ ОБЪЕМЕ СВОБОДНОЙ ПАМЯТИ ДИСКА АКТИВНОГО ДИСКОВОДА. КОМАНДА ЗАПРАШИВАЕТ

DIR MASK:

В ОТВЕТ МОЖНО УКАЗАТЬ, НАПРИМЕР,

\*.\* ИЛИ ПУСТО (ВЫДАТЬ КАТАЛОГ ВСЕХ ФАЙЛОВ),

\*.PAS (ВЫДАТЬ КАТАЛОГ ВСЕХ ФАЙЛОВ С РАСШИРЕНИЯМИ ИМЕН - PAS) И НАЖАТЬ КЛАВИШУ <E>.

#### 6.2. O (COMPILER OPTIONS)

- УКАЗАТЬ ИНФОРМАЦИЮ О ТОМ, КУДА НАПРАВИТЬ РЕЗУЛЬТАТ ТРАНСЛЯЦИИ:

MEMORY

- РЕЗУЛЬТАТ ТРАНСЛЯЦИИ В ВИДЕ ГОТОВОЙ К ВЫПОЛНЕНИЮ ПРОГРАММЫ ОСТАВИТЬ В ОСНОВНОЙ ПАМЯТИ (ПРИНИМАЕТСЯ "ПО УМОЛЧАНИЮ");

COM-FILE

- РЕЗУЛЬТАТ ТРАНСЛЯЦИИ В ВИДЕ ГОТОВОЙ К ВЫПОЛНЕНИЮ ПРОГРАММЫ РАЗМЕСТИТЬ НА ДИСКЕ В ФАЙЛЕ С ИМЕНЕМ РАБОЧЕГО ФАЙЛА И РАСШИРЕНИЕМ COM;

CNN-FILE

- РЕЗУЛЬТАТ ТРАНСЛЯЦИИ БЕЗ ДОПОЛНЕНИЯ ЕГО ТАКИМИ ЧАСТЯМИ, КАК, НАПРИМЕР, ПРОЦЕДУРЫ ВВОДА-ВЫВОДА, ПОМЕСТИТЬ В ФАЙЛ НА ДИСКЕ С ИМЕНЕМ РАБОЧЕГО ФАЙЛА, НО С РАСШИРЕНИЕМ CNN. ДО ВЫПОЛНЕНИЯ ФАЙЛ

ТИПА ЦНН ДОЛЖЕН БЫТЬ ОБРАБОТАН ЗАГРУЗЧИКОМ. ТАКИЕ ФАЙЛЫ ОБЫЧНО ДОБАВЛЯЮТСЯ К СОМ-ФАЙЛАМ.

COMPILE --> MEMORY  
СОМ-FILE  
ЦНН-FILE

ИНФОРМАЦИЯ КОМАНДЕ ПЕРЕДАЕТСЯ НАЖАТИЕМ ОДНОЙ ИЗ КЛАВИШ M, C, H В СООТВЕТСТВИИ С ТЕМ, КУДА СЛЕДУЕТ НАПРАВИТЬ РЕЗУЛЬТАТ ТРАНСЛЯЦИИ. ВЫХОД ИЗ КОМАНДЫ В ГЛАВНОЕ МЕНЮ СИСТЕМЫ ЗАДАЕМ НАЖАТИЕМ КЛАВИШИ Q.

6.3. X (EXECUTE)

- ВЫПОЛНИТЬ ПРОГРАММУ ИЗ СОМ-ФАЙЛА.

КОМАНДА ЗАПРАШИВАЕТ

PROGRAM:

В ОТВЕТ НАБИРАЕМ СООТВЕТСТВУЮЩЕЕ ИМЯ СОМ-ФАЙЛА И НАЖИМАЕТ КЛАВИШУ <E>. НАПРИМЕР, НАБОР PRINT ЗАДАЕТ ВЫПОЛНЕНИЕ ПРОГРАММЫ ПОСТРАНИЧНОЙ ВЫДАЧИ НА ПРИНТЕР ТЕКСТА ПАСКАЛЬ-ПРОГРАММЫ, ХРАНЯЩЕЯСЯ В ФАЙЛЕ ТИПА PAS НА ДИСКЕ.

В ОТВЕТ НА ЗАПРОС ПРОГРАММЫ

ИМЯ ФАЙЛА ДЛЯ ПЕЧАТИ:

НАБИРАЕМ ИМЯ ФАЙЛА И НАЖИМАЕМ КЛАВИШУ <E>. ПОСЛЕ ВЫДАЧИ ОДНОЙ СТРАНИЦЫ ТЕКСТА ПРОГРАММА СПРАШИВАЕТ ПРЕКРАТИТЬ ИЛИ ПРОДОЛЖИТЬ ВЫДАЧУ. ДО ОТВЕТА НА ВОПРОС, ЕСЛИ ПЕЧАТЬ НЕ ЗАВЕРШЕНА, НЕОБХОДИМО ПОЗАБОТИТЬСЯ О ЗАМЕНЕ БУМАГИ НА ПРИНТЕРЕ. ПЕРЕД ВЫЗОВОМ ПРОГРАММЫ ПРИНТЕР ДОЛЖЕТ БЫТЬ ВКЛЮЧЕН И НА НЕМ УСТАНОВЛЕНА БУМАГА.

7. УДАЛЕНИЕ ФАЙЛА С ДИСКА

УДАЛЕНИЕ ДИСКОВОГО ФАЙЛА ОСУЩЕСТВЛЯЕТ СИСТЕМА SCP ПО КОМАНДЕ A>ERA < РАСШИРЕННОЕ ИМЯ УДАЛЯЕМОГО ФАЙЛА><E>.

УНИЧТОЖИТЬ ФАЙЛ МОЖНО ПОСЛЕ НАЧАЛЬНОЙ ЗАГРУЗКИ СИСТЕМЫ SCP ДО ВХОДА В ТУРБО ПАСКАЛЬ ИЛИ ПОСЛЕ ВЫХОДА ИЗ СИСТЕМЫ ТУРБО ПАСКАЛЬ В ОПЕРАЦИОННУЮ СИСТЕМУ SCP, ОСУЩЕСТВЛЯЕМОГО ПО КОМАНДЕ Q(QUIT) ГЛАВНОГО МЕНЮ.

8. СООБЩЕНИЕ ОБ ОШИБКАХ ВРЕМЕНИ ТРАНСЛЯЦИИ

ПРИ ОБНАРУЖЕНИИ ОШИБКИ ТРАНСЛЯТОР ВЫДАЕТ НА ЭКРАН НОМЕР ДИАГНОСТИЧЕСКОГО СООБЩЕНИЯ. ПОЯСНЯЮЩИЙ ОШИБКУ ТЕКСТ ВЫДАЕТСЯ, ЕСЛИ ПРИ ЗАГРУЗКЕ СИСТЕМЫ ТУРБО В ОСНОВНУЮ ПАМЯТЬ ПОМЕЩЕН ФАЙЛ С ТЕКСТАМИ СООБЩЕНИЙ.

НОМЕР СООБЩ.	ОЖИДАЕТСЯ	!!	НОМЕР СООБЩ.	!	ОЖИДАЕТСЯ
06	" = "	!!	23	!	ВЫРАЖЕНИЕ ЦЕЛОГО ТИПА
07	" ; "	!!	24	!	ПЕРЕМЕННАЯ ЦЕЛОГО ТИПА
08	" [ "	!!	25	!	КОНСТАНТА ЦЕЛОГО ИЛИ ВЕ-
		!!		!	ЩЕСТВЕННОГО ТИПА
09	" ] "	!!	26	!	ВЫРАЖЕНИЕ ТИПА INTEGER
		!!		!	ИЛИ REAL
10	" . "	!!	27	!	ПЕРЕМЕННАЯ ТИПА INTEGER
		!!		!	ИЛИ REAL
11	" .. "	!!	28	!	ПЕРЕМЕННАЯ ССЫЛОЧНОГО ТИПА
12	WEGIN	!!	29	!	ПЕРЕМЕННАЯ КОМБИНИРОВАННОГО
		!!		!	ТИПА
13	DO	!!	30	!	ПРОСТОЙ ТИП (ЛЮБОЙ СКАЛЯРНЫЙ,
		!!		!	КРОМЕ REAL)
14	END	!!	31	!	ПРОСТОЕ ВЫРАЖЕНИЕ

15	!	OF	!!	32	!	СТРОКОВАЯ КОНСТАНТА
16	!	ПРОЦЕДУРА ИЛИ	!!	33	!	СТРОКОВОЕ ВЫРАЖЕНИЕ
	!	ФУНКЦИЯ	!!		!	
17	!	THEN	!!	34	!	СТРОКОВАЯ ПЕРЕМЕННАЯ
18	!	TO ИЛИ DOWNTO	!!	35	!	ТЕКСТОВЫЙ ФАЙЛ
20	!	ЛОГИЧЕСКОЕ	!!	36	!	ИДЕНТИФИКАТОР ТИПА
	!	ВЫРАЖЕНИЕ	!!		!	
21	!	ФАЙЛОВАЯ ПЕРЕ-	!!	37	!	ФАЙЛ БЕЗ ТИПА
	!	МЕННАЯ	!!		!	
22	!	КОНСТАНТА ЦЕЛО-	!!		!	
	!	ГО ТИПА	!!		!	

НОМЕР	!	ПОЯСНЕНИЯ
СООБЩЕНИЯ	!	
40	!	НЕОПРЕДЕЛЕННАЯ МЕТКА
41	!	НЕОПРЕДЕЛЕННЫЙ ИДЕНТИФИКАТОР
42	!	ПРИ ОБЪЯВЛЕНИИ ТИПА ИСПОЛЬЗУЕТСЯ НЕОПРЕДЕЛЕННЫЙ ИДЕНТИФИКАТОР ТИПА
43	!	ДУБЛИРОВАНИЕ В БЛОКЕ ИДЕНТИФИКАТОРА ИЛИ МЕТКИ
44	!	НЕСООТВЕТСТВИЕ ТИПА (ЛЕВОЙ И ПРАВОЙ ЧАСТИ ОПЕРАТОРА); ПРИСВАИВАНИЯ; ФОРМАЛЬНОГО ИЛИ ФАКТИЧЕСКОГО ПАРАМЕТРА; НЕДОПУСТИМЫЙ ТИП ИНДЕКСА; ТИПЫ ОПЕРАНДОВ ВЫРАЖЕНИЯ (НЕСОВМЕСТИМЫ)
45	!	ЗНАЧЕНИЕ КОНСТАНТЫ НЕДОПУСТИМО ДЛЯ ТИПА
46	!	ТИП МЕТКИ ВАРИАНТА В ОПЕРАТОРЕ CASE НЕ СООТВЕТСТВУЕТ ТИПУ СЕЛЕКТОРА ВАРИАНТА
47	!	НЕДОПУСТИМЫЙ ДЛЯ ОПЕРАЦИИ ТИП ОПЕРАНДА
48	!	ТИП РЕЗУЛЬТАТА НЕ ДОЛЖЕН БЫТЬ СЛОЖНЫМ
49	!	ДЛИНА СТРОКИ ВНЕ ДИАПАЗОНА 1..255
50	!	НЕДОПУСТИМАЯ ДЛИНА СТРОКОВОЙ КОНСТАНТЫ
51	!	НЕДОПУСТИМЫЙ БАЗОВЫЙ ТИП (ДЛЯ ДИАПАЗОНА)
52	!	НИЖНЯЯ ГРАНИЦА ПРЕВЫШАЕТ ВЕРХНЮЮ ГРАНИЦУ
53	!	НЕДОПУСТИМОЕ ИСПОЛЬЗОВАНИЕ ЗАРЕЗЕРВИРОВАННОГО СЛОВА
54	!	НЕДОПУСТИМОЕ НАЗНАЧЕНИЕ
55	!	СТРОКОВАЯ КОНСТАНТА ПРЕВЫШАЕТ ДЛИНУ СТРОКИ
56	!	ОШИБКА В КОНСТАНТЕ ТИПА INTEGER
57	!	ОШИБКА В КОНСТАНТЕ ТИПА REAL
58	!	ИДЕНТИФИКАТОР СОДЕРЖИТ НЕДОПУСТИМЫЙ СИМВОЛ
60	!	НЕДОПУСТИМОЕ ИСПОЛЬЗОВАНИЕ КОНСТАНТЫ
61	!	НЕДОПУСТИМОЕ ИСПОЛЬЗОВАНИЕ ФАЙЛА ИЛИ ССЫЛКИ
62	!	НЕДОПУСТИМОЕ ИСПОЛЬЗОВАНИЕ ДАННОГО КОМБИНИРОВАННОГО ТИПА
63	!	НЕДОПУСТИМОЕ ИСПОЛЬЗОВАНИЕ ТЕКСТОВОГО ФАЙЛА
66	!	НЕДОПУСТИМЫЙ ТИП ЭЛЕМЕНТА СПИСКА ВВОДА-ВЫВОДА
67	!	ФАЙЛ НЕ МОЖЕТ БЫТЬ ФОРМАЛЬНЫМ ПАРАМЕТРОМ-ПЕРЕМЕННОЙ
68	!	НЕДОПУСТИМЫЙ ТИП ЭЛЕМЕНТОВ ФАЙЛА
70	!	БАЗОВЫЙ ТИП ДЛЯ МНОЖЕСТВА ВКЛЮЧАЕТ БОЛЬШЕ 256 ЗНАЧЕНИЙ
71	!	НЕДОПУСТИМЫЙ ПЕРЕХОД I/O GOTO
72	!	МЕТКА ОПРЕДЕЛЕНА ВНЕ СЛОКА
73	!	НЕТ ОТЛОЖЕННОГО ОПИСАНИЯ ПРОЦЕДУРЫ
90	!	ФАЙЛ НЕ НАЙДЕН

91 ! НЕОЖИДАННЫЙ КОНЕЦ ПРОГРАММЫ (ВЕРОЯТНО, В ПРОГРАММЕ  
! BEGIN БОЛЬШЕ, ЧЕМ END)  
93 ! НЕДОПУСТИМАЯ ДИРЕКТИВА КОМПИЛЯТОРА  
97 ! ЧРЕЗМЕРНА ГЛУБИНА ВЛОЖЕНИЯ ОПЕРАТОРОВ WITH  
98 ! ПЕРЕПОЛНЕНИЕ ПАМЯТИ (ПОПЫТКА РАСПРЕДЕЛИТЬ ДЛЯ ПЕ-  
! РЕМЕННЫХ ПАМЯТИ БОЛЬШЕ, ЧЕМ ДОСТУПНО)  
99 ! ПЕРЕПОЛНЕНИЕ ПАМЯТИ, ДОСТУПНОЙ ТРАНСЛЯТОРУ (РАЗ-  
! БИТЬ ПРОГРАММУ НА СЕГМЕНТЫ, ВЫЗЫВАЕМЫЕ С ПОМОЩЬЮ  
! ФАЙЛОВ ВКЛЮЧЕНИЯ)

## 2. ОШИБКИ ВРЕМЕНИ ВЫПОЛНЕНИЯ ПРОГРАММЫ

НОМЕР СООБЩЕНИЯ !	ПОЯСНЕНИЯ
01 !	ПЕРЕПОЛНЕНИЕ ДЛЯ ДАННЫХ REAL
02 !	ДЕЛЕНИЕ НА НУЛЬ
03 !	АРГУМЕНТ ФУНКЦИИ SQRT ОТРИЦАТЕЛЕН
04 !	АРГУМЕНТ ФУНКЦИИ LN НЕ ЯВЛЯЕТСЯ ПОЛОЖИТЕЛЬНЫМ
10 !	ДЛИНА СТРОКИ НЕДОПУСТИМА (БОЛЬШЕ 256 ИЛИ ОТЛИЧНА ! ОТ 1 ДЛЯ СТРОК, СОВМЕСТИМЫХ С CHAR)
11 !	В ПРОЦЕДУРЕ COPY, DELETE, INSERT ИНДЕКС ВНЕ ДИАПА- ! ЗОНА 1..255
90 !	ЗНАЧЕНИЕ ИНДЕКСА ВСЕ ДОПУСТИМОГО ДИАПАЗОНА
91 !	ЗНАЧЕНИЕ ПЕРЕМЕННОЙ ТИПА ДИАПАЗОН ВНЕ ДИАПАЗОНА
92 !	ДЛЯ ВЫПОЛНЕНИЯ ПРОЦЕДУРЫ NEW НЕДОСТАТОЧНО СВОБОД- ! НОЙ ПАМЯТИ

## 10. СООБЩЕНИЯ ОБ ОШИБКАХ ВВОДА-ВЫВОДА

НОМЕР СООБЩЕНИЯ !	ПОЯСНЕНИЯ
01 !	ФАЙЛА НЕТ НА ДИСКЕ
02 !	ФАЙЛ НЕ ОТКРЫТ ДЛЯ ЧТЕНИЯ (НЕ БЫЛО ПРОЦЕДУРЫ RESET, ! ПОПЫТКА ЧИТАТЬ С УСТРОЙСТВА LST)
03 !	ФАЙЛ НЕ ОТКРЫТ ДЛЯ ВЫВОДА
04 !	ФАЙЛ НЕ ОТКРЫТ
10 !	В ТЕКСТОВОМ ФАЙЛЕ ЧИСЛОВОЕ ЗНАЧЕНИЕ СОДЕРЖИТ НЕДО- ! ПУСТЫЕ СИМВОЛЫ
22 !	ОБ'ЯВЛЕНИЕ И НАЗНАЧЕНИЕ СТАНДАРТНОМУ ФАЙЛУ ЗАПРЕЩЕНО
90 !	ТИП КОМПОНЕНТОВ ФАЙЛА НЕ СООТВЕТСТВУЕТ ТИПУ КОМПО- ! НЕНТОВ ФАЙЛОВОЙ ПЕРЕМЕННОЙ
91 !	ПОИСК ПОСЛЕ КОНЦА ФАЙЛА
99 !	НЕПРЕДВИДЕННЫЙ КОНЕЦ ФАЙЛА ! 1. ФИЗИЧЕСКИЙ КОНЕЦ ФАЙЛА ПРЕДШЕСТВУЕТ СИМВОЛУ КОНЦА ! ФАЙЛА (CTRL-Z) ! 2. ЧТЕНИЕ ЗА ПРЕДЕЛАМИ ФАЙЛА ! 3. ОШИБКА В ФАЙЛЕ
F0 !	ДИСК ПЕРЕПОЛНЕН
F1 !	ПЕРЕПОЛНЕНО ОГЛАВЛЕНИЕ ДИСКА
F2 !	КОЛИЧЕСТВО КОМПОНЕНТОВ ФАЙЛА БОЛЬШЕ 65535
F3 !	ЧРЕЗМЕРНОЕ КОЛИЧЕСТВО ОДНОВРЕМЕННО ОТКРЫТЫХ ФАЙЛОВ

FF ! ПРИ ПОПЫТКЕ ЗАКРЫТЬ ФАЙЛ ОН НЕ ОБНАРУЖЕН (НАПРИМЕР,  
! ИЗ-ЗА СМЕНЫ ДИСКА)

### 11. ДИРЕКТИВЫ КОМПИЛЯТОРА

НЕКОТОРЫМИ ВОЗМОЖНОСТЯМИ ЯЗЫКА В СИСТЕМЕ ТУРБО ПАСКАЛЬ УПРАВЛЯЮТ ЧЕРЕЗ ДИРЕКТИВЫ КОМПИЛЯТОРА. ДИРЕКТИВА ВКЛЮЧАЕТСЯ В ТЕКСТ ПРОГРАММЫ В ВИДЕ КОММЕНТАРИЯ СПЕЦИАЛЬНОГО ВИДА. ВСЕ ДИРЕКТИВЫ ИМЕЮТ ЗНАЧЕНИЯ "ПО УМОЛЧАНИЮ", КОТОРЫЕ ВЫБРАНЫ, ИСХОДЯ ИЗ ТРЕБОВАНИЯ УВЕЛИЧЕНИЯ СКОРОСТИ ВЫПОЛНЕНИЯ ПРОГРАММЫ И УМЕНЬШЕНИЯ ДЛИНЫ ПРОГРАММЫ В МАШИННЫХ КОДАХ.

(\*XR\*\*) ВКЛЮЧИТЬ, (\*XR-) ВЫКЛЮЧИТЬ ПРОВЕРКУ ПРИНАДЛЕЖНОСТИ ЗНАЧЕНИЯ СООТВЕТСТВУЮЩИМ ДИАПАЗОНАМ. РЕКОМЕНДУЕТСЯ ИСПОЛЬЗОВАТЬ "R+" ПРИ ОТЛАДКЕ ПРОГРАММЫ И "R-" В ОТЛАЖЕННОЙ ПРОГРАММЕ.

(\*XV\*\*) ВКЛЮЧИТЬ, (\*XV-) ВЫКЛЮЧИТЬ ПРОВЕРКУ СТРОГОГО СООТВЕТСТВИЯ ФОРМАЛЬНЫХ И ФАКТИЧЕСКИХ ПАРАМЕТРОВ ТИПА STRING(L) (ДЛИНЫ ФОРМАЛЬНОГО И ФАКТИЧЕСКОГО ПАРАМЕТРОВ ДОЛЖНЫ СОВПАДАТЬ).

(\*XA\*\*) ГЕНЕРИРОВАТЬ НЕРЕКУРСИВНЫЙ КОД, (\*XA-) ГЕНЕРИРОВАТЬ КОД, ДОПУСКАЮЩИЙ РЕКУРСИВНЫЕ ВЫЗОВЫ.

(\*XW \*) УСТАНОВИТЬ МАКСИМАЛЬНУЮ ГЛУБИНУ ВЛОЖЕНИЯ ОПЕРАТОРОВ WITH, РАВНУЮ (1<=N<=9).

(\*XI < ИМЯ ФАЙЛА\*) ВКЛЮЧИТЬ В КОМПИЛЯЦИЮ ФАЙЛ С УКАЗАННЫМ ИМЕНЕМ.

ДИРЕКТИВА "I" ПОЗВОЛЯЕТ ВКЛЮЧАТЬ В ТЕКСТ ПАСКАЛЬ-ПРОГРАММЫ РАНЕЕ РАЗРАБОТАННЫЕ ПОДПРОГРАММЫ, РАЗМЕЩЕННЫЕ В СВОИХ ФАЙЛАХ.

КОММЕНТАРИЙ ДИРЕКТИВЫ I ДОЛЖЕН РАЗМЕЩАТЬСЯ В ОТДЕЛЬНОЙ СТРОКЕ.

### ПРИЛОЖЕНИЕ 1

#### СПИСОК ЗАРЕЗЕРВИРОВАННЫХ СЛОВ

ABSOLUTE	EXTERNAL	NIL	SHR
AND	FILE	NOT	STRING
ARRAY	FOR	OF	THEN
BEGIN	FORWARD	OR	TO
CASE	FUNCTION	PACKED	TYPE
CONST	GOTO	PROCEDURE	UNTIL
DIV	IF	PROGRAM	VAR
DO	IN	RECORD	WHILE
DOWNTD	INLINE	REPEAT	WITH
ELSE	LABEL	SET	XOR
END	MOD	SHL	

### ПРИЛОЖЕНИЕ 2

#### СПИСОК ПРЕДОПРЕДЕЛЕННЫХ ИДЕНТИФИКАТОРОВ

ARCTAN	DELAY	LN	RENAME
ASSIGN	DELETE	LO	RESET
AUX	EOF	LOWVIDED	REWRITE
AUXINPTR	EOLN	LST	ROUND
AUXOUTPTR	ERASE	LSTOUTPTR	SEEK
BLOCKREAD	EXECUTE	MARK	SIN
BLOCKWRITE	EXP	MAXINT	SIZEOF

BOOLEAN	FALSE	MEM	SQR
BUFLEN	FILEPOS	MEMAVAIL	SQRT
BYTE	FILESIZE	MOVE	STR
CHAIN	FILLCHAR	NEW	SUCC
CHAR	FLUSH	NORMVIDEO	SWAP
CLOSE	FRAC	ODD	TEXT
CLREOL	GETMEM	ORD	TRM
CLRSCR	GOTOXY	OUTPUT	TRUE
CON	HEAPPTR	PI	TRUNC
CONINPTR	HI	POPRT	UPCASE
CONOUTPTR	IORESULT	POS	USR
CONCAT	INPUT	PRED	USRINPTR
CONSTPTR	INSLINE	PTR	USROUTPTR
COPY	INSERT	RANDOM	VAL
COS	INT	RANDOMIZE	WRITE
CRTEXTIT	INTEGER	READ	WRITELN
CRTINIT	KBD	READLN	
DELLINE	KEYPRESSED	REAL	
	LENGTH	RELEASE	

ПРИЛОЖЕНИЕ 3

ПРЕДОПРЕДЕЛЕННЫЕ ФУНКЦИИ

НАЗВАНИЕ ГРУППЫ	НАЗНАЧЕНИЕ ФУНКЦИИ	ОБРАЩЕНИЕ К ФУНКЦИИ	ТИП АРГУМЕНТА	ТИП РЕЗУЛЬТАТА	ПРИМЕЧАНИЯ
1	2	3	4	5	6
	!X!	!ABS(X)	!INTEGER	!INTEGER	!ТИП РЕЗУЛЬТАТА-
			!REAL	!REAL	!СОВПАДАЕТ С ТИ-
					!ПОМ АРГУМЕНТА
МАТЕМАТИЧЕСКИЕ	$x^2$	SQR(X)	!-"-"	!-"-"	!-"-"
	$\sqrt{x}$	SQRT(X)	!-"-"	!REAL	!X>=0
	$e^x$	EXP(X)	!-"-"	!-"-"	!-"-"
	$\ln x$	LN(X)	!-"-"	!-"-"	!X>0
	cos x	COS(X)	!-"-"	!-"-"	!X В РАДИАННОЙ
					!МЕРЕ
	SIN X	SIN(X)	!-"-"	!-"-"	!-"-"
	arctg x	ARCTAN(X)	!-"-"	!-"-"	!РЕЗУЛЬТАТ В РА-
					!ДИАННОЙ МЕРЕ
	!ВЗЯТИЕ ЦЕЛОЙ ЧАСТИ	!INT(X)	!-"-"	!-"-"	!INT(5.9)=5
	!ВЗЯТИЕ ДРОНОЙ ЧАСТИ	!FRAC(X)	!-"-"	!-"-"	!INT(-4.1)=-4
					!FRAC(X)=
					!X-INT(X)
ФУНКЦИИ ОРДИНАЛЬНЫХ ТИПОВ	!ОБРАЗОВАНИЕ ПРЕДШТУВУЮЩАГО ЭЛЕМЕНТА	!PRED(X)	!ОРДИНАЛЬНАЯ	!ОРДИНАЛЬНЫЙ	!РЕЗУЛЬТАТ-ЭЛЕМЕНТ ОРДИНАЛЬНОГО ТИПА, ПРЕДШТУВУЮЩИЙ X
	!ОБРАЗОВАНИЕ СЛЕДУЮЩАГО ЭЛЕМЕНТА	!SUCC(X)	!-"-"	!-"-"	!РЕЗУЛЬТАТ-ЭЛЕ-

	! НИЕ ПОСЛЕ-	!	"-"	!	"-"	!	МЕНТА ОРДИНАЛЬ-
	! ДУЩЕГО	!		!		!	! НОГО ТИПА, СЛЕ-
	! ЭЛЕМЕНТА	!		!		!	! ДУЩИЯ ЗА X
	! ПОЛУЧЕНИЕ	! CHR (X)	!	! INTEGER	!	! CHAR	! РЕЗУЛЬТАТ--ЛИТЕ-
	! ЛИТЕРЫ	!	!	! BYTE	!	!	! РА, КОД КОИ-7
	!	!	!	!	!	!	! КОТОРОИ РАВЕН X
ПРЕОБРА-	! ПОЛУЧЕНИЕ	! ORD (X)	!	! ОРДИНАЛЬ-	!	! INTEGER	! РЕЗУЛЬТАТ--НОМЕР
ЗОВАНИЯ	! ПОРЯДКОВОГО!	!	!	! НЫЯ	!	!	! ЗНАЧЕНИЯ X В
	! НОМЕРА X	!	!	!	!	!	! СПИСКЕ ДОПУСТИ-
	!	!	!	!	!	!	! МЫХ ЗНАЧЕНИИ
	!	!	!	!	!	!	! ТИПА
	!	!	!	!	!	!	! ORD ('A')=65;
	!	!	!	!	!	!	! ORD (20)=20;
	!	!	!	!	!	!	! ORD (-31)=-31;
	!	!	!	!	!	!	! ORD (FALSE)=0
	! ПРЕОБРАЗ-	! TRUNC (X)	!	! REAL	!	! INTEGER	! TRUNC (5.9)=5;
	! ВАНИЕ ОТ	!	!	!	!	!	! TRUNC (4.1)=-4
	! REAL K	!	!	!	!	!	!
	! INTEGER	!	!	!	!	!	!
	! ОКРУГЛЕНИЕ	! ROUND (X)	!	! REAL	!	! INTEGER	! ROUND (5.9)=6
	! ДО БЛИЖАЙШЕ-	!	!	!	!	!	!
	! ГО ЦЕЛОГО	!	!	!	!	!	! ROUND (5.2)=5
	!	!	!	!	!	!	! ROUND (-4.1)=-4
	!	!	!	!	!	!	! ROUND (-4.9)=-5
	! ПРОВЕРКА НА	! ODD (X)	!	! INTEGER	!	! BOOLEAN	! РЕЗУЛЬТАТ TRUE,
ЛОГИЧЕ-	! ЧЕТНОСТЬ	!	!	!	!	!	! ЕСЛИ X НЕЧЕТНОЕ;
СКИЕ	!	!	!	!	!	!	! FALSE, ЕСЛИ X
	!	!	!	!	!	!	! ЧЕТНОЕ
	! ОПРОС КЛА-	! KEYPRES-	!	! NET	!	! BOOLEAN	! РЕЗУЛЬТАТ TRUE,
	! ВИАТУРЫ	! SED	!	!	!	!	! ЕСЛИ НАЖАТА
	!	!	!	!	!	!	! КАКАЯ--ЛИБО
	!	!	!	!	!	!	! КЛАВИША КЛАВИА-
	!	!	!	!	!	!	! ТУРЫ
	! ПРОВЕРКА НА	! EOF (F)	!	! ФАЙЛОВЫЯ	!	! BOOLEAN	! РЕЗУЛЬТАТ TRUE,
	! КОНЕЦ ФАЙЛА	! EOF	!	!	!	!	! ЕСЛИ ОБНАРУЖЕН
	!	!	!	!	!	!	! КОНЕЦ ФАЙЛА
	!	!	!	!	!	!	! F (INPUT);
	!	!	!	!	!	!	! FALSE--В ПРОТИВ-
	!	!	!	!	!	!	! НОМ СЛУЧАЕ
ФАЙЛО-	! ПРОВЕРКА НА	! EOLN (F)	!	! ФАЙЛОВЫЯ	!	! BOOLEAN	! РЕЗУЛЬТАТ TRUE,
ВНЫЕ	! КОНЕЦ СТРОКИ!	! EOLN	!	!	!	!	! ЕСЛИ ОБНАРУЖЕН
	!	!	!	!	!	!	! КОНЕЦ СТРОКИ В
	!	!	!	!	!	!	! ФАЙЛЕ F (INPUT);
	!	!	!	!	!	!	! FALSE-- В ПРО-
	!	!	!	!	!	!	! ТИВНОМ СЛУЧАЕ
	! ОПРЕДЕЛЕНИЕ	! FILEPOS (F)	!	! ФАЙЛОВЫЯ	!	! INTEGER	! КОМПОНЕНТЫ НУ-
	! НОМЕРА ТЕКУ-	!	!	!	!	!	! МЕРУЮТСЯ НАЧИ-
	! ЩЕГО КОМПО-	!	!	!	!	!	! НАЯ С 0
	! НЕНТА ФАЙЛА	!	!	!	!	!	!
	! ОПРЕДЕЛЕНИЕ	! FILESIZE	!	!"-"	!	!"-"	! ЕСЛИ ФАЙЛ ПУСТ,
	! ЧИСЛА КОМПО-	! (F)	!	!	!	!	! РЕЗУЛЬТАТ РАВЕН
	! НЕНТОВ ФАЙЛА!	!	!	!	!	!	! 0

	! ПОЛУЧЕНИЕ	! RANDOM	! NET	! REAL	! РЕЗУЛЬТАТ-СЛУ-
	! СЛУЧАЙНОГО	!	!	!	! ЧАЙНОЕ ЧИСЛО В
	! ЧИСЛА	!	!	!	! ДИАПАЗОНЕ ОТ
ПРОЧИЕ	!	!	!	!	! НУЛЯ ДО ЕДИНИЦЫ
	! -"-"-"	! RANDOM(X)	! INTEGER	! INTEGER	! РЕЗУЛЬТАТ-СЛУ-
	!	!	!	!	! ЧАЙНОЕ ЧИСЛО ИЗ
	!	!	!	!	! ДИАПАЗОНА ОТ 0
	!	!	!	!	! ДО X
	! ПОЛУЧЕНИЕ	! SIZEOF(I)	! ЛЮБОЙ	! INTEGER	! РЕЗУЛЬТАТ -ЧИСЛО
	! ДЛИНЫ ПОЛЯ,	!	!	!	! БАЙТОВ, ЗАНИМАЕ-
	! ЗАНИМАЕМОГО	!	!	!	! МЫХ ДАННЫМИ ТИПА
	! ДАННЫМ	!	!	!	! I

ЗДЕСЬ

X - ВЫРАЖЕНИЕ СООТВЕТСТВУЮЩЕГО ТИПА;

F - ПЕРЕМЕННАЯ ФАЙЛОВОГО ТИПА (ДО ИСПОЛЬЗОВАНИЯ ПЕРЕМЕННОЙ ПРОЦЕДУРОЙ ASSIGN ДОЛЖНА БЫТЬ УСТАНОВЛЕНА СВЯЗЬ МЕЖДУ ПЕРЕМЕННОЙ И ФАЙЛОМ НА ДИСКЕ);

I - ПЕРЕМЕННАЯ.

#### ЛИТЕРАТУРА

1. ГРОГОНО П. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ПАСКАЛЬ. М.: МИР, 1982.
2. ИЕНСЕН К., ВИРТ Н. ПАСКАЛЬ. РУКОВОДСТВО ДЛЯ ПОЛЬЗОВАТЕЛЯ И ОПИСАНИЕ ЯЗЫКА. М.: ФИНАНСЫ И СТАТИСТИКА, 1982.
3. ФОРСАЙТ Р. ПАСКАЛЬ ДЛЯ ВСЕХ. М.: МАШИНОСТРОЕНИЕ, 1988.
4. ПЕРМИНОВ О.Н. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ПАСКАЛЬ. М.: РАДИО И СВЯЗЬ, 1988.
5. СЕМАШКО Г.Л., САЛТЫКОВ А.И. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ПАСКАЛЬ. М.: НАУКА, 1988.
6. ПРАЙС Д. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ПАСКАЛЬ: ПРАКТИЧЕСКОЕ РУКОВОДСТВО /ПЕР. С АНГЛ.; ПОД РЕД. О.Н.ПЕРМИНОВА. М.: МИР, 1987.
7. АБРАМОВ С.А., ЗИМА Е.В. НАЧАЛА ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ПАСКАЛЬ. М.: НАУКА, 1987.
8. ВИРТ Н. АЛГОРИТМЫ + СТРУКТУРЫ ДАННЫХ = ПРОГРАММЫ /ПЕР. С АНГЛ.; ПОД РЕД. Д.Б.ПОДШИВАЛОВА. М.: МИР, 1985.

## СО Д Е Р Ж А Н И Е

	Стр.
ВВЕДЕНИЕ	3
1. ОСНОВНЫЕ ЭЛЕМЕНТЫ ЯЗЫКА	3
1.1. Алфавит	3
1.2. Элементарные конструкции	3
1.3. Типы данных	4
1.4. Переменные	5
1.5. Предопределенные типы данных	5
1.5.1. Вещественный тип (REAL)	5
1.5.2. Целый тип (INTEGER)	5
1.5.3. Байтовый тип (BYTE)	5
1.5.4. Булевский тип (BOOLEAN)	5
1.5.5. Выражение	6
1.6. Структура программы	6
1.6.1. Раздел описания меток	7
1.6.2. Раздел определения констант	7
1.6.3. Раздел определения типов	8
1.6.4. Раздел описания переменных	8
1.6.5. Раздел описания подпрограмм	8
1.6.6. Раздел операторов	8
1.7. Основные операторы	9
1.7.1. Оператор присваивания	9
1.7.2. Оператор перехода	9
1.7.3. Условный оператор	9
1.7.4. Оператор цикла	10
1.8. Организация диалога «Машина-пользователь»	11
1.8.1. Предопределенные процедуры вывода	11
1.8.2. Предопределенные процедуры ввода	12
1.9. Пример	13
1.10. Задания по теме	14
2. РЯД ДОПОЛНИТЕЛЬНЫХ ВОЗМОЖНОСТЕЙ	16
2.1. Литерный тип (CHAR)	16
2.2. Строковый тип (STRING [n])	16
2.3. Перечислимый тип	20
2.4. Тип диапазон	21
2.5. Оператор выбора (CASE)	22
2.6. Подпрограммы	23
2.7. Рекурсивные подпрограммы	27
2.8. Задания по теме	29
3. ИНСТРУКЦИЯ ПО РАБОТЕ В СРЕДЕ ТУРБО-ПАСКАЛЬ	30
ПРИЛОЖЕНИЯ	
1. Список зарезервированных слов	37
2. Список предопределенных идентификаторов	37
3. Предопределенные функции	38
ЛИТЕРАТУРА	40

