

У 84081

Б-742

5156/83

Боголюбовская А.А. и др.

Б1-11-83-512



ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ

Б1-11-83-512

ДЕПОНИРОВАННАЯ ПУБЛИКАЦИЯ

Дубна 1983

БОГОЛЮБСКАЯ А.А., ЖИДКОВА И.Б., РОСТОВЦЕВ В.А.

51-11-83-512

СИСТЕМА ПРОГРАММИРОВАНИЯ

REDUCE-2

ВХОДНОЙ ЯЗЫК

РУКОВОДСТВО

ПОД РЕДАКЦИЕЙ В.А. РОСТОВЦЕВА

18 07 83

ДУБНА, О И Я И, Л В Т А, 1983.

Объединенный институт
ядерных исследований
БИБЛИОТЕКА

В ОСНОВУ ДАННОГО РУКОВОДСТВА ПОЛОЖЕНЫ
REDUCE 2 USER'S MANUAL [1]

REDUCE INTERACTIVE LESSONS [2].

В СИЛУ ОСОБЕННОСТЕЙ ПЕЧАТАЮЩЕГО УСТРОЙСТВА ЭВМ, НА КОТОРОМ ПОЛУЧЕНО НАСТОЯЩЕЕ РУКОВОДСТВО, НЕКОТОРЫЕ СИМВОЛЫ В ТЕКСТЕ ПРИШЛОСЬ ЗАМЕНИТЬ:

- 1) ВЕРТИКАЛЬНОЙ ЧЕРТЕ (РАЗДЕЛЯЮЩЕЙ АЛЬТЕРНАТИВЫ В МЕТАЛИНГВИСТИЧЕСКИХ ФОРМУЛАХ) СООТВЕТСТВУЕТ ЗНАК [
- 2) АПОСТРОФУ СООТВЕТСТВУЕТ ЗНАК ' "
- 3) ТВЕРДОМУ ЗНАКУ СООТВЕТСТВУЕТ ЗНАК "

КРОМЕ ЭТОГО, РУКОВОДСТВО ИМЕЕТ ЕЩЕ НЕКОТОРЫЕ ОСОБЕННОСТИ :

- 1) ТАК ЖЕ, КАК В [1], МЕТАЛИНГВИСТИЧЕСКИЕ ФОРМУЛЫ, ЗАДАЮЩИЕ СИНТАКСИЧЕСКИЕ ПРАВИЛА ЯЗЫКА **REDUCE**, НЕ ПРЕТЕНДУЮТ НА СТРОГОСТЬ, ОНИ СКОРЕЕ НАГЛЯДНЫ И УДОБНЫ В ИСПОЛЬЗОВАНИИ.
- 2) В ТЕКСТЕ РУКОВОДСТВА, ТАМ, ГДЕ ЗНАКИ ПРЕПИНАНИЯ МОГЛИ ПОМЕШАТЬ ИЗЛОЖЕНИЮ СИНТАКСИСА ЯЗЫКА **REDUCE**, МЫ ДЕЛАЛИ ВЫБОР НЕ В ПОЛЬЗУ РУССКОЙ ПУНКТУАЦИИ.
- 3) ГРАММАТИЧЕСКИЙ ПЕРЕНОС СЛОВ В ТЕКСТЕ ДЕЛАЕТ ПРОГРАММА **ТЕХТА [12]**. КАК УТВЕРЖДАЮТ АВТОРЫ ЭТОЙ ПРОГРАММЫ, ОНА ПОЧТИ НЕ ОШИБАЕТСЯ.

АВТОРЫ БЛАГОДАРИТ ВСЕХ, ЧЬИ ПРЕДЛОЖЕНИЯ СПОСОБСТВОВАЛИ УЛУЧШЕНИЮ РУКОВОДСТВА, И ОСОБЕННО - А.Г.ГРОЗИНА И А.А.РАСТОРГУЕВА. МЫ ИСКРЕННЕ ПРИЗНАТЕЛЬНЫ В.П.ШИРИКОВУ ЗА ИНТЕРЕС К РАБОТЕ И ПОДДЕРЖКУ.

ДЛЯ ПОЛУЧЕНИЯ РУКОВОДСТВА В ВИДЕ, УДОБНОМ ДЛЯ ЧТЕНИЯ И РАЗУМНОМ С ТОЧКИ ЗРЕНИЯ ЭКОНОМИИ БУМАГИ, МЫ ПОЛЬЗОВАЛИСЬ ПРОГРАММАМИ **РТВ [13]**, **VARB**, **ТЕХТА**, **DLIST [14]**. МЫ БЛАГОДАРИМ А.А.КОРНЕЙЧУКА ЗА КОНСУЛЬТАЦИИ ПО ЭТИМ ПРОГРАММАМ И А.В.ГУСЕВА ЗА ПОМОЩЬ ПРИ АДАПТАЦИИ ТЕКСТА ДЛЯ НОВОГО ВАРИАНТА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ "РУССКОЯ ПЕЧАТИ".

СИСТЕМА ПРОГРАММИРОВАНИЯ REDUCE ПРЕДСТАВЛЯЕТ ИНТЕРЕС ДЛЯ МАТЕМАТИКОВ, ФИЗИКОВ И ИНЖЕНЕРОВ. ЭТА СИСТЕМА ПОЗВОЛЯЕТ ПРОВОДИТЬ АЛГЕБРАИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ ШИРОКОГО КЛАССА, В ТОМ ЧИСЛЕ:

- 1) РАСКРЫВАТЬ СКОБКИ (УМНОЖАТЬ, ВОЗВОДИТЬ В ЦЕЛУЮ СТЕПЕНЬ), ПРИВОДИТЬ ПОДОБНЫЕ И УПОРЯДОЧИВАТЬ ЧЛЕНЫ В ПОЛИНОМАХ И РАЦИОНАЛЬНЫХ ФУНКЦИЯХ; ПРИВОДИТЬ РАЦИОНАЛЬНЫЕ ВЫРАЖЕНИЯ К ОБЩЕМУ ЗНАМЕНАТЕЛЮ;
- 2) НАХОДИТЬ НАИБОЛЬШИЙ ОБЩИЙ ДЕЛИТЕЛЬ ДВУХ ПОЛИНОМОВ;
- 3) ДИФФЕРЕНЦИРОВАТЬ АНАЛИТИЧЕСКИЕ ВЫРАЖЕНИЯ;
- 4) ВЫПОЛНЯТЬ ПОДСТАНОВКИ И АНАЛИЗИРОВАТЬ ВЫРАЖЕНИЯ (ВЫДЕЛЯТЬ ЧИСЛИТЕЛЬ И ЗНАМЕНАТЕЛЬ, НАХОДИТЬ КОЭФФИЦИЕНТЫ ПРИ РАЗЛИЧНЫХ СТЕПЕНЯХ, ПРЕНЕБРЕГАТЬ АСИМПТОТИЧЕСКИ МАЛЫМИ ЧЛЕНАМИ);
- 5) ПРЕДСТАВЛЯТЬ АЛГЕБРАИЧЕСКОЕ ВЫРАЖЕНИЕ ПРИ ВЫВОДЕ ЕГО НА ПЕЧАТЬ В РАЗЛИЧНЫХ ФОРМАХ;
- 6) ВЫПОЛНЯТЬ МАТЕМАТИЧЕСКИЕ ОПЕРАЦИИ НАД МАТРИЦАМИ С СИМВОЛЬНЫМИ ЭЛЕМЕНТАМИ, В ТОМ ЧИСЛЕ: ТРАНСПОНИРОВАТЬ МАТРИЦУ, ВЫЧИСЛЯТЬ ШПУР, ДЕТЕРМИНАНТ, НАХОДИТЬ ОБРАТНУЮ МАТРИЦУ;
- 7) ПРОВОДИТЬ АЛГЕБРАИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ ДЛЯ ЗАДАЧ ФИЗИКИ (ВЫСКИХ ЭНЕРГИИ) В ТОМ ЧИСЛЕ В РАМКАХ АЛГЕБРЫ СПИНОРОВ ДЛЯ ЧАСТИЦ СО СПИНОМ ПОЛОВИНА И АЛГЕБРЫ ВЕКТОРОВ ДЛЯ ЧАСТИЦ СО СПИНОМ 1;
- 8) ВЫПОЛНЯТЬ ТЕНЗОРНЫЕ ОПЕРАЦИИ;
- 9) ИНТЕГРИРОВАТЬ АНАЛИТИЧЕСКИЕ ВЫРАЖЕНИЯ (БРАТЬ НЕОПРЕДЕЛЕННЫЕ ИНТЕГРАЛЫ ОТ ПОЛИНОМОВ, ЛОГАРИФИЧЕСКИХ ФУНКЦИЙ, ЭКСПОНЕНЦИАЛЬНЫХ ФУНКЦИЙ, ФУНКЦИИ ТАНГЕНСА И АРКТАНГЕНСА).

1. КРАТКИЙ ОБЗОР СИСТЕМЫ REDUCE.

СИСТЕМА ПРОГРАММИРОВАНИЯ REDUCE (СЛОВОТВОР СИСТЕМЫ - А.С. HEARN) ОТНОСИТСЯ К УНИВЕРСАЛЬНЫМ СИСТЕМАМ ДЛЯ АНАЛИТИЧЕСКИХ ВЫЧИСЛЕНИЙ (САВ)[3], ПОД "АНАЛИТИЧЕСКИМИ ВЫЧИСЛЕНИЯМИ" ПРИМЕНИТЕЛЬНО К САВ ПОНИМАЮТСЯ ДЕЙСТВИЯ НАД АЛГЕБРАИЧЕСКИМИ ВЫРАЖЕНИЯМИ И ТОЖДЕСТВЕННЫЕ АЛГЕБРАИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ, РЕЗУЛЬТАТОМ КОТОРЫХ ЯВЛЯЕТСЯ НЕ ЧИСЛО, А АНАЛИТИЧЕСКОЕ ВЫРАЖЕНИЕ.

СИСТЕМА REDUCE ПРЕДСТАВЛЯЕТ СОБОЙ КОМПЛЕКС ВЗАИМОСВЯЗАННЫХ ПРОГРАММ, В КОТОРЫХ РЕАЛИЗОВАНЫ АЛГОРИТМЫ АНАЛИТИЧЕСКИХ ВЫЧИСЛЕНИЙ - ПОЛИНОМИАЛЬНЫЕ АЛГОРИТМЫ, АЛГОРИТМЫ МАТРИЧНОЙ АЛГЕБРЫ И Т.Д. СИСТЕМА НАПИСАНА НА ЯЗЫКЕ RLISP (ВНУТРЕННИЙ REDUCE ИЛИ СИМВОЛЬНАЯ МОДА ЯЗЫКА REDUCE), ДЛЯ ТОГО, ЧТОБЫ ОБЛЕГЧИТЬ ВЗАИМОДЕЙСТВИЕ ПОЛЬЗОВАТЕЛЕЙ-НЕПРОГРАММИСТОВ С СИСТЕМОЙ, БЫЛ РАЗРАБОТАН ВХОДНОЙ ЯЗЫК - ПРОСТО REDUCE ИЛИ АЛГЕБРАИЧЕСКАЯ МОДА ЯЗЫКА REDUCE. ЗНАНИЯ АЛГЕБРАИЧЕСКОЙ МОДЫ ВПОЛНЕ ДОСТАТОЧНО ДЛЯ ТОГО, ЧТОБЫ УСПЕШНО ИСПОЛЬЗОВАТЬ ИМЕЮЩИЕСЯ АЛГЕБРАИЧЕСКИЕ ВОЗМОЖНОСТИ СИСТЕМЫ. ЕСЛИ ЖЕ ПОЛЬЗОВАТЕЛЬ ХОЧЕТ РАСШИРИТЬ СИСТЕМУ (ДОБАВИТЬ В НЕЕ НЕКОТОРЫЕ ПРОЦЕДУРЫ ДЛЯ СВОЕГО КЛАССА ЗАДАЧ) ИЛИ ПРОВЕСТИ КАКИЕ-ЛИБО СИМВОЛЬНЫЕ (ПОДОБНЫЕ ЛИСПОВСКИМ) ВЫЧИСЛЕНИЯ, ОН ДОЛЖЕН ОВЛАДЕТЬ СИМВОЛЬНОЙ МОДОЙ.

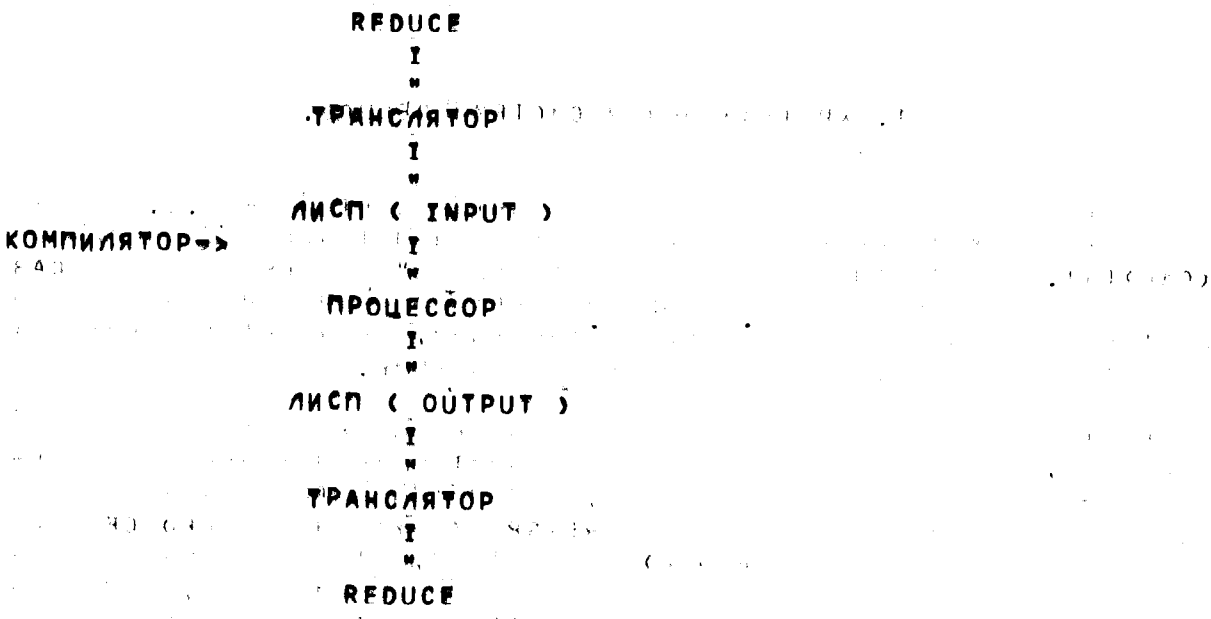
ДАННОЕ РУКОВОДСТВО (ЗА ИСКЛЮЧЕНИЕМ ПОСЛЕДНЕЙ ГЛАВЫ) ПОСВЯЩЕ-

НО АЛГЕБРАИЧЕСКОЯ МОДЕ.

ОДНА ИЗ ОСОБЕННОСТЕЙ СИСТЕМЫ - ЕЕ ИНТЕРАКТИВНЫЙ ХАРАКТЕР. ПО ЗАМЫСЛУ РАЗРАБОТЧИКОВ СИСТЕМА REDUCE ПРЕДНАЗНАЧЕНА ПРЕЖДЕ ВСЕГО ДЛЯ ИНТЕРАКТИВНОГО ИСПОЛЬЗОВАНИЯ (Т.Е. ДЛЯ РАБОТЫ В ДИАЛОГОВОМ РЕЖИМЕ). ХОТЯ, КОНЕЧНО, ОНА МОЖЕТ ИСПОЛЬЗОВАТЬСЯ И В ПАКЕТНОМ РЕЖИМЕ. ДИАЛОГОВЫЙ РЕЖИМ ПОЗВОЛЯЕТ КОНТРОЛИРОВАТЬ И КОРРЕКТИРОВАТЬ ХОД ВЫПОЛНЕНИЯ ПРОГРАММЫ.

ИНТЕРАКТИВНЫМ ХАРАКТЕРОМ СИСТЕМЫ ОБЪЯСНЯЮТСЯ НЕКОТОРЫЕ ЕЕ КОНСТРУКТИВНЫЕ ОСОБЕННОСТИ. В ЯЗЫКЕ REDUCE, НАПРИМЕР, НЕТ ПОНЯТИЯ ОСНОВНОЙ ПРОГРАММЫ, КАК В ФОРТРАНЕ. ПРОГРАММА, НАПИСАННАЯ НА ЯЗЫКЕ REDUCE ПРЕДСТАВЛЯЕТ СОБОЙ ПОСЛЕДОВАТЕЛЬНОСТЬ КОМАНД (КОМАНДА - ПРИКАЗ СИСТЕМЕ СДЕЛАТЬ ЧТО-ЛИБО). КАЖДАЯ КОМАНДА СРАЗУ ЖЕ ПЕЧАТАЕТСЯ, ВЫПОЛНЯЕТСЯ ЭВМ, А ЗАТЕМ ПЕЧАТАЕТСЯ ОТВЕТ СИСТЕМЫ НА ЭТУ КОМАНДУ. ПОСЛЕ ЭТОГО СИСТЕМА ГОТОВА ВЫПОЛНИТЬ СЛЕДУЮЩУЮ КОМАНДУ ПОЛЬЗОВАТЕЛЯ.

В ДЕЙСТВИТЕЛЬНОСТИ ПРОГРАММА ПОЛЬЗОВАТЕЛЯ ВЫПОЛНЯЕТСЯ ПОСРЕДСТВОМ ЯЗЫКА STANDARD LISP, КОТОРЫЙ ЯВЛЯЕТСЯ ПРОМЕЖУТОЧНЫМ МЕЖДУ СИСТЕМОЙ И ЭВМ. КАЖДАЯ КОМАНДА ПРОГРАММЫ ВНАЧАЛЕ ЧИТАЕТСЯ СИСТЕМОЙ И ПРЕДСТАВЛЯЕТСЯ В КАНОНИЧЕСКОЙ РЕЛЬСОВСКОЙ ФОРМЕ. ЗАТЕМ ВЫПОЛНЯЮТСЯ НЕОБХОДИМЫЕ ДЕЙСТВИЯ НА УРОВНЕ АЛГЕБРАИЧЕСКОГО ПРОЦЕССОРА СИСТЕМЫ, ПОЛУЧЕННЫЙ "ПОЛУФАБРИКАТ" ТРАНСЛИРУЕТСЯ В ЛИСП И ВЫПОЛНЯЕТСЯ ЛИСП-ИНТЕРПРЕТОРОМ /*/. РЕЗУЛЬТАТ, ВЫДАННЫЙ ЭВМ, ТРАНСЛИРУЕТСЯ ИЗ ЛИСПОВСКОЙ ФОРМЫ В РЕЛЬСОВСКУЮ, ПРЕДСТАВЛЯЕТСЯ СИСТЕМОЙ В УДОБНОМ ДЛЯ ЧТЕНИЯ ВИДЕ И ВЫДАЕТСЯ НА ПЕЧАТЬ. СХЕМА ВЫПОЛНЕНИЯ КАЖДОЙ КОМАНДЫ ТАКОВА :



ЕСЛИ ПРЕДПОЛАГАЕТСЯ МНОГОКРАТНОЕ ИСПОЛЬЗОВАНИЕ ПРОГРАММЫ, СЛЕДУЕТ СКОМПИЛИРОВАТЬ ЕЕ И РАБОТАТЬ ДАЛЕЕ СО СКОМПИЛИРОВАННЫМ ВАРИАНТОМ - ЭТО ЗНАЧИТЕЛЬНО СЭКОНОМИТ МАШИННОЕ ВРЕМЯ.

ИСПОЛЬЗОВАНИЕ ЯЗЫКА ЛИСП В КАЧЕСТВЕ ПРОМЕЖУТОЧНОГО - ТАКЖЕ ОДНА ИЗ ОСОБЕННОСТЕЙ СИСТЕМЫ REDUCE. ВХОДНОЙ ЯЗЫК (АЛГЕБРА-

* ИНТЕРПРЕТАЦИЯ - НЕПОСРЕДСТВЕННОЕ ПОШАГОВОЕ ВЫПОЛНЕНИЕ ЭВМ КАЖДОЙ КОМАНДЫ ПРОГРАММЫ ПОЛЬЗОВАТЕЛЯ. КОМПИЛЯЦИЯ - ПРЕДВАРИТЕЛЬНЫЙ ПЕРЕВОД ВСЕЙ ПРОГРАММЫ НА МАШИННЫЙ ЯЗЫК.

ИЧЕСКАЯ МОДА) СОХРАНЯЕТ МНОГИЕ ХАРАКТЕРНЫЕ ЧЕРТЫ ЛИСПА, СИМВОЛЬНАЯ МОДА (RLISP) - ЭТО ПРОСТО АЛГОЛОПОДОБНЫЙ ВАРИАНТ ЯЗЫКА ЛISP.

ВСЕ КОМАНДЫ ПОЛЬЗОВАТЕЛЯ МОЖНО РАЗДЕЛИТЬ НА ДВА КЛАССА: ВЫЧИСЛЯЕМЫЕ КОМАНДЫ (Т.Е. КОМАНДЫ, ТРЕБУЮЩИЕ ОТВЕТА СИСТЕМЫ И НЕВЫЧИСЛЯЕМЫЕ (СООТВЕТСТВЕННО, НЕ ТРЕБУЮЩИЕ ОТВЕТА). ОТМЕТИМ, ЧТО ДАННАЯ КЛАССИФИКАЦИЯ КОМАНД НЕ ЯВЛЯЕТСЯ ОБЩЕПРИНЯТОЙ И НЕ СООТВЕТСТВУЕТ, НАПРИМЕР, [1]. КОМАНДА ОБЯЗАТЕЛЬНО ДОЛЖНА ОКОНЧИВАТЬСЯ ОГРАНИЧИТЕЛЕМ.

<ВЫЧИСЛЯЕМАЯ КОМАНДА> ::= <ВЫРАЖЕНИЕ><ОГРАНИЧИТЕЛЬ> & <ОПЕРАТОР><ОГРАНИЧИТЕЛЬ>

ВЫПОЛНЯЯ ВЫЧИСЛЯЕМУЮ КОМАНДУ, СИСТЕМА ПРОИЗВЕДЕТ УКАЗАННЫЕ ДЕЙСТВИЯ, ВЫЧИСЛИТ ЗНАЧЕНИЕ ВЫРАЖЕНИЯ ИЛИ ОПЕРАТОРА (ОПЕРАТОРЫ В ЯЗЫКЕ ИМЕЮТ ЗНАЧЕНИЯ) И ВЕРНЕТ ЕГО (КАК РЕЗУЛЬТАТ).

ВЫЧИСЛЯЕМЫЕ КОМАНДЫ РАССМАТРИВАЮТСЯ В ГЛАВЕ 3 НАСТОЯЩЕГО РУКОВОДСТВА, НЕВЫЧИСЛЯЕМЫЕ - В ГЛАВЕ 4. ГЛАВА 2 ПОСВЯЩАЕТСЯ ДАННЫМ, ОПЕРАЦИЯМ И СВЯЗАННЫМ С НИМИ ВОПРОСАМ, ГЛАВА 5 - МАТРИЧНОЙ АЛГЕБРЕ, ГЛАВА 6 - ВЫЧИСЛЕНИЯМ ДЛЯ ЗАДАЧ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ, ГЛАВА 7 - СИМВОЛЬНОЙ МОДЕ. В ПРИЛОЖЕНИИ А ПРИВОДИТСЯ СПИСОК ЗАРЕЗЕРВИРОВАННЫХ ИДЕНТИФИКАТОРОВ, НЕВЫЧИСЛЯЕМЫХ КОМАНД, СООБЩЕНИЯ ОБ ОШИБКАХ И ДИАГНОСТИЧЕСКИХ СООБЩЕНИЯ. ПРИЛОЖЕНИЕ Б ОТНОСИТСЯ К РАБОТЕ С СИСТЕМОЙ НА ЭВМ ЭИЯ.

2. ПРИМЕРЫ ПРОГРАММ НА ЯЗЫКЕ REDUCE, ФОРМАТ ПРОГРАММ,

ПРОГРАММА, НАПИСАННАЯ НА ЯЗЫКЕ REDUCE ИМЕЕТ СВОБОДНЫЙ ФОРМАТ - Т.Е. КОМАНДА ЯЗЫКА МОЖЕТ ЗАНИМАТЬ КАК ОДНУ ТАК И НЕСКОЛЬКО СТАНДАРТНЫХ ПЕРФОКАРТ (КОЛОНКИ С 1 ПО 72 ВКЛЮЧИТЕЛЬНО), ПЕРЕНОС НА ДРУГУЮ КАРТУ ОСУЩЕСТВЛЯЕТСЯ БЕЗ ВСЯКИХ СИМВОЛОВ ПРОДОЛЖЕНИЯ (НО ИДЕНТИФИКАТОР ПРИ ПЕРЕНОСЕ НЕ ДОЛЖЕН РАЗРЫВАТЬСЯ). КОНЕЦ КОМАНДЫ СИСТЕМА РАСПОЗНАЕТ ПО ОГРАНИЧИТЕЛЮ. ЕСЛИ В КАЧЕСТВЕ ОГРАНИЧИТЕЛЯ ИСПОЛЬЗУЕТСЯ ЗНАК ";", ОТВЕТ СИСТЕМЫ (ЕСЛИ ДАННАЯ КОМАНДА ТРЕБУЕТ ОТВЕТА) ПЕЧАТАЕТСЯ, ЗНАК ";" В КОНЦЕ КОМАНДЫ ВЫКЛЮЧАЕТ ПЕЧАТЬ. НА ОДНОЙ СТРОКЕ МОЖНО ЗАПИСАТЬ И НЕСКОЛЬКО КОМАНД.

В ПРИМЕРАХ И ВО ФРАГМЕНТАХ ПРОГРАММ КОМАНДЫ ПРОГРАММИСТА И ОТВЕТЫ СИСТЕМЫ ПЕЧАТАЮТСЯ С НАЧАЛА СТРОКИ (ТАКЖЕ, КАК И НА ИСТИНГАХ ВЫДАЧИ). КАК И В СТАНДАРТНОЙ ВЫДАЧЕ, КОМАНДЫ ОКОНЧИВАЮТСЯ ОГРАНИЧИТЕЛЯМИ, ОТВЕТЫ СИСТЕМЫ (ЗНАЧЕНИЯ ВЫРАЖЕНИЙ) ПЕЧАТАЮТСЯ БЕЗ ОГРАНИЧИТЕЛЕЙ.

РАССМОТРИМ НЕСКОЛЬКО ПРОСТЫХ ПРИМЕРОВ, ИЛЛЮСТРИРУЮЩИХ ВОЗМОЖНОСТИ СИСТЕМЫ. ПРИМЕРЫ, ПРИВЕДЕННЫЕ В НАСТОЯЩЕМ РУКОВОДСТВЕ, ПРОВЕРЯЛИСЬ С ВЕРСИЕЙ СИСТЕМЫ REDUCE 2 ОТ 15 АПРЕЛЯ 1979 Г.

(1) ЭТОТ ПРИМЕР ПОКАЗЫВАЕТ, КАК СИСТЕМА ВЫПОЛНЯЕТ ДЕЙСТВИЯ НАД АЛГЕБРАИЧЕСКИМИ ВЫРАЖЕНИЯМИ: РАСКРЫВАЕТ СКОБКИ, ПРИВОДИТ ПОДОБНЫЕ, ПРИВОДИТ РАЦИОНАЛЬНЫЕ ВЫРАЖЕНИЯ К ОБЩЕМУ ЗНАМЕНАТЕЛЮ. ПЕРЕМЕННЫЕ В ВЫРАЖЕНИИ СИСТЕМА РАСПОЛАГАЕТ В ОПРЕДЕЛЕННОМ ПОРЯДКЕ. ОБРАТИТЕ ВНИМАНИЕ НА ТО, ЧТО ПОЛЬЗОВАТЕЛЬ МАНИПУЛИРУЕТ

ПЕРЕМЕННЫМИ, КОТОРЫМ НЕ ПРИСВОЕНО НИКАКОГО ЗНАЧЕНИЯ.

$$A \text{ Г} = \frac{2 \cdot G + 3 \cdot G + H \cdot 3 / H}{2}$$

КОМАНДА

$$A \text{ I} = H + 5 \cdot G$$

ОТВЕТ СИСТЕМЫ

$$B \text{ I} = A \cdot (F + G)$$

$$B \text{ Г} = H \cdot G + H \cdot F + 5 \cdot G + 5 \cdot G \cdot F$$

$$B \text{ I} = A \cdot 2$$

$$B \text{ I} = H + 10 \cdot H \cdot G + 25 \cdot G$$

$$A + 1/A$$

$$(H + 10 \cdot H \cdot G + 25 \cdot G + 1) / (H + 5 \cdot G)$$

(2) СИСТЕМА МОЖЕТ ПРОИЗВОДИТЬ НЕ ТОЛЬКО АНАЛИТИЧЕСКИЕ, НО И ЧИСЛЕННЫЕ РАСЧЕТЫ:

$$V_1 = (987654321 / 15) \cdot 3$$

$$V_1 = 15626944334403 / 125$$

$$V_1 = -V / 25 + 3 \cdot (13 \cdot 5)$$

$$V_1 = (-15626944259403) / 3125$$

(3) В ЭТОМ ПРИМЕРЕ ЗАДАННОЕ АЛГЕБРАИЧЕСКОЕ ВЫРАЖЕНИЕ ВНАЧАЛЕ УПРОЩАЕТСЯ СИСТЕМОЙ, А ЗАТЕМ ДИФФЕРЕНЦИРУЕТСЯ ПОСРЕДСТВОМ ОПЕРАЦИИ DF. ИМЕНА ФУНКЦИЙ SIN, LOG, COS ИЗВЕСТНЫ СИСТЕМЕ, ИМЕНА ОСТАЛЬНЫХ ФУНКЦИЙ ВВОДЯТСЯ ОПИСАНИЕМ OPERATOR. ПРАВИЛО LET ЗАДАЕТ ПОДСТАНОВКУ.

OPERATOR TG, F, SC

$$EX_1 = \text{SIN}(-X+Y) + \text{LOG}(E) + (3 \cdot Y + 2 \cdot \text{COS}(-1)) \cdot (1/2) + \text{TG}(X \cdot Y) + \text{F}(3 \cdot X + 2)$$

$$EX_1 = \text{SQRT}(\text{COS}(1)) \cdot \text{SQRT}(3) \cdot Y = \text{SIN}(Y+X) + \text{TG}(Y+X) + \text{F}(3 \cdot X) + 1$$

DF(EX, Y)

$$\text{SQRT}(\text{COS}(1)) \cdot \text{SQRT}(3) + \text{DF}(\text{TG}(Y+X), Y) = \text{COS}(Y+X) \cdot X$$

FOR ALL Z LET DF(TG(Z), Z) = SC(Z) + 2

DF(EX, Y)

$$\text{SQRT}(\text{COS}(1)) \cdot \text{SQRT}(3) + \text{SC}(Y+X) + X = \text{COS}(Y+X) \cdot X$$

В ЯЗЫКЕ REDUCE ИМЕЮТСЯ СЛЕДУЮЩИЕ ОСНОВНЫЕ КЛАССЫ ДАННЫХ: ЧИСЛА, ПЕРЕМЕННЫЕ, МАССИВЫ, ЯДРА, СТРОКИ, ВНЕШНИЕ ФАЙЛЫ, МАТРИЦЫ И ВЕКТОРЫ. СТРОКИ ИСПОЛЗУЮТСЯ В ОПЕРАТОРЕ WRITE (П. 3.9), ВНЕШНИЕ ФАЙЛЫ - В ОПЕРАЦИЯХ ВВОДА-ВЫВОДА (П. 4.5), МАТРИЦЫ - В МАТРИЧНЫХ ВЫЧИСЛЕНИЯХ (ГЛАВА 5), ВЕКТОРЫ - В ВЫЧИСЛЕНИЯХ ДЛЯ ЗАДАЧ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ (ГЛАВА 6). ЧИСЛА, ПЕРЕМЕННЫЕ, МАССИВЫ И ЯДРА РАССМАТРИВАЮТСЯ В ЭТОЙ ГЛАВЕ.

2.1 НЕКОТОРЫЕ СИНТАКСИЧЕСКИЕ ПРАВИЛА.

1. СТАНДАРТНЫЙ НАБОР ОСНОВНЫХ ЗНАКОВ ЯЗЫКА REDUCE.

СИМВОЛЫ ЯЗЫКА REDUCE СОСТАВЛЯЮТСЯ ИЗ СЛЕДУЮЩИХ ОСНОВНЫХ ЗНАКОВ:

1) ЗАГЛАВНЫХ БУКВ ЛАТИНСКОГО АЛФАВИТА:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

2) ЦИФР:

0123456789 ;

3) СПЕЦИАЛЬНЫХ ЗНАКОВ:

! " # \$ % & ' () * + , - . / : ; < > =

(ОДИН ИЗ СИМВОЛОВ "!" ОБОЗНАЧАЕТ АПОСТРОФ, ДРУГОЙ - САМ СЕБЯ).

ПРОГРАММЫ, СОСТАВЛЕННЫЕ НА ОСНОВЕ ЭТОГО СТАНДАРТНОГО НАБОРА ЗНАКОВ, БУДУТ ВОСПРИНИМАТЬСЯ ЛЮБЫМ ИЗ ДОСТУПНЫХ ВАРИАНТОВ СИСТЕМЫ REDUCE, В НЕКОТОРЫХ ВАРИАНТАХ СИСТЕМЫ (НАПРИМЕР, В ВАРИАНТЕ, РЕАЛИЗОВАННОМ НА ЭВМ ПДП-10) ДЛЯ ПРЕДСТАВЛЕНИЯ ОТДЕЛЬНЫХ ОПЕРАЦИЙ ПРИМЕНЯЮТСЯ ДОПОЛНИТЕЛЬНЫЕ ЗНАКИ. СИМВОЛЫ ОПЕРАЦИЙ ДЛЯ ЭТИХ ЭВМ ДОЛЖНЫ БЫТЬ СОГЛАСОВАНЫ СО СМЫСЛОМ ТАКИХ СПЕЦИАЛЬНЫХ ЗНАКОВ. ОДНАКО, ДЛЯ БОЛЬШЕЙ ОБЩНОСТИ, В ДАННОМ РУКОВОДСТВЕ МЫ ОГРАНИЧИМСЯ СТАНДАРТНЫМ НАБОРОМ ЗНАКОВ.

2. ИДЕНТИФИКАТОРЫ.

ИДЕНТИФИКАТОРЫ В ЯЗЫКЕ REDUCE - ЭТО ПОСЛЕДОВАТЕЛЬНОСТИ БУКВЕННО-ЦИФРОВЫХ ЗНАКОВ (Т.Е. ЗАГЛАВНЫХ БУКВ ИЛИ ЦИФР) ДЛИНОЙ ОТ 1 ДО 24 ЗНАКОВ, НАЧИНАЮЩИЕСЯ С БУКВЫ.

НАПРИМЕР:

A

AZ

P1

Q23P

AVERYLONGVARIABLE

(В П.7.2. РАССМАТРИВАЕТСЯ ОБОБЩЕНИЕ ИДЕНТИФИКАТОРОВ).

ИДЕНТИФИКАТОРЫ ИСПОЛЬЗУЮТСЯ КАК ИМЕНА ПЕРЕМЕННЫХ, МАССИВОВ, ОПЕРАЦИЙ, ФАЙЛОВ, А ТАКЖЕ КАК МЕТКИ.

ОГРАНИЧЕНИЯ.

ЗАРЕЗЕРВИРОВАННЫЕ В ЯЗЫКЕ REDUCE СЛОВА (СМ. ПРИЛОЖЕНИЕ А) НЕЛЬЗЯ ИСПОЛЬЗОВАТЬ В КАЧЕСТВЕ ИДЕНТИФИКАТОРОВ. ЗАРЕЗЕРВИРОВАННЫЕ СЛОВА МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ ЛИШЬ ТАК, КАК ЭТО ПОДРАЗУМЕВАЕТСЯ В ДАННОМ РУКОВОДСТВЕ.

ВНУТРИ ИДЕНТИФИКАТОРОВ НЕ ДОЛЖНО БЫТЬ ПРОБЕЛОВ, КРОМЕ ТОГО, ВСЯКИЙ ИДЕНТИФИКАТОР ДОЛЖЕН РАСПОЛАГАТЬСЯ НА ОДНОЙ ВХОДНОЙ СТРОКЕ ТЕКСТА ПРОГРАММЫ (БЕЗ ПЕРЕНОСОВ).

3. КОММЕНТАРИИ.

КОММЕНТАРИИ ПОЛЕЗНЫ ДЛЯ ВКЛЮЧЕНИЯ ПОЯСНЕНИЯ В РАЗЛИЧНЫХ МЕСТАХ ПРОГРАММЫ. ОНИ МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ В СЛЕДУЮЩЕЙ ФОРМЕ:

COMMENT <ПРОИЗВОЛЬНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ, НЕ СОДЕРЖАШАЯ ОГРАНИЧИТЕЛЯ><ОГРАНИЧИТЕЛЬ>

ПРИМЕР:

COMMENT THIS IS A COMMENT)

КОММЕНТАРИИ НА ВВОДЕ ЭКВИВАЛЕНТНЫ ПРОБЕЛУ. КРОМЕ ТОГО, ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ

END <ЛЮБАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ, НЕ СОДЕРЖАШАЯ ОГРАНИЧИТЕЛЯ, ПРАВЫХ ГРУППОВЫХ СКОБОК ИЛИ ЗАРЕЗЕРВИРОВАННЫХ СЛОВ
END, ELSE, UNTIL>

ЭКВИВАЛЕНТНА ЗАРЕЗЕРВИРОВАННОМУ СЛОРУ END. (ЛЕВЫМИ И ПРАВЫМИ ГРУППОВЫМИ СКОБКАМИ В ДАННОМ РУКОВОДСТВЕ НАЗЫВАЮТСЯ, СООТВЕТСТВЕННО, ПАРЫ СИМВОЛОВ << И >>).

2.2 ЧИСЛА.

В ЯЗЫКЕ REDUCE УПОТРЕБЛЯЮТСЯ ЧИСЛА ДВУХ ТИПОВ: ЦЕЛЫЕ И ВЕЩЕСТВЕННЫЕ. ЦЕЛЫЕ ЧИСЛА ПРЕДСТАВЛЯЮТСЯ ПОСЛЕДОВАТЕЛЬНОСТЯМИ ДЕСЯТИЧНЫХ ЦИФР (БЕЗ ДЕСЯТИЧНОЙ ТОЧКИ), СО ЗНАКОМ ИЛИ БЕЗ НЕГО.

НАПРИМЕР:

НА КОЛИЧЕСТВО ЦИФР В ЧИСЛЕ ПРАКТИЧЕСКИ НЕ НАКЛАДЫВАЕТСЯ ОГРАНИЧЕНИЯ, Т.К. В СИСТЕМЕ ПРИМЕНЯЕТСЯ РАЦИОНАЛЬНАЯ АРИФМЕТИКА С ПРОИЗВОЛЬНОЙ ТОЧНОСТЬЮ, ЧТО ОТЛИЧАЕТ ЯЗЫК REDUCE ОТ ТРАДИЦИОННЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ.

ДЕЯТЕЛВТЕЛЪНЫЕ ЧИСЛА МОГУТ БЫТЬ ЗАПИСАНЫ В ОДНОЙ ИЗ ДВУХ ФОРМ:

- 1) В ВИДЕ ПОСЛЕДОВАТЕЛЬНОСТИ ЦИФР (ДЛИНОЙ ОТ ОДНОЙ ЦИФРЫ ДО ДЕВЯТИ), СО ЗНАКОМ ИЛИ БЕЗ НЕГО И С ДЕСЯТИЧНОЙ ТОЧКОЙ, РАСПОЛОЖЕННОЙ ВНУТРИ ПОСЛЕДОВАТЕЛЬНОСТИ ЦИФР ИЛИ НЕПОСРЕДСТВЕННО ПОСЛЕ НЕЕ;
- 2) В ВИДЕ ТАКОЙ ЖЕ, КАК В П.1) ПОСЛЕДОВАТЕЛЬНОСТИ ЦИФР С ДЕСЯТИЧНОЙ ТОЧКОЙ И СО ЗНАКОМ ИЛИ БЕЗ НЕГО, ЗА КОТОРОЙ СЛЕДУЕТ ДЕСЯТИЧНЫЙ ПОРЯДОК. ДЕСЯТИЧНЫЙ ПОРЯДОК ОБОЗНАЧАЕТСЯ БУКВОЙ E, НЕПОСРЕДСТВЕННО ЗА КОТОРОЙ СЛЕДУЕТ ЦЕЛОЕ ЧИСЛО СО ЗНАКОМ ИЛИ БЕЗ НЕГО.

НАПРИМЕР:

32.
+32.0
0,32E2
320,E=1

-ЭТО ВАРИАНТЫ ЗАПИСИ ОДНОГО И ТОГО ЖЕ ВЕЩЕСТВЕННОГО ЧИСЛА 32.

ОГРАНИЧЕНИЕ.

ПОСЛЕДОВАТЕЛЬНОСТЬ ЦИФР В ИЗОБРАЖЕНИИ ВЕЩЕСТВЕННОГО ЧИСЛА НЕ ДОЛЖНА НАЧИНАТЬСЯ С ДЕСЯТИЧНОЙ ТОЧКИ, Т.Е., НЕЛЬЗЯ ЗАПИСАТЬ:

.5, *.23, +.12

2.3 ПЕРЕМЕННЫЕ.

1. ТИПЫ ПЕРЕМЕННЫХ.

ПЕРЕМЕННЫЕ В ЯЗЫКЕ ОПРЕДЕЛЯЮТСЯ ИМЕНЕМ И ТИПОМ. ИМЕНА ПЕРЕМЕННЫХ ДОЛЖНЫ БЫТЬ СИНТАКСИЧЕСКИ ПРАВИЛЬНЫМИ ИДЕНТИФИКАТОРАМИ. ДОПУСКАЕТСЯ ТРИ ТИПА ПЕРЕМЕННЫХ - ЦЕЛЫЕ, ДЕЙСТВИТЕЛЬНЫЕ И СКАЛЯРНЫЕ. ИМЕНА ЭТИХ ПЕРЕМЕННЫХ ВВОДЯТСЯ С ПОМОЩЬЮ ОПИСАНИЯ ТИПА, СООТВЕТСТВЕННО, INTEGER, REAL, SCALAR.

ПРИМЕРЫ:

INTEGER M,N;

REAL M1;

SCALAR X,Y;

ПЕРЕМЕННЫЕ С ПОМОЩЬЮ ОПИСАНИЯ ТИПА МОЖНО ВВОДИТЬ НА ЛЮБОМ УРОВНЕ ПРОГРАММЫ. ОПИСАНИЯ ТИПА, ВКЛЮЧЕННЫЕ В БЛОК, ИМЕЮТ СИЛУ ТОЛЬКО ВНУТРИ ЭТОГО БЛОКА.

НЕОПИСАННЫЕ ПЕРЕМЕННЫЕ РАССМАТРИВАЮТСЯ СИСТЕМОЙ КАК СКАЛЯРНЫЕ. ЭТО ОСНОВНОЙ ТИП СИМВОЛЬНЫХ ПЕРЕМЕННЫХ.

ПЕРЕМЕННЫЕ ЯЗЫКА REDUCE МОГУТ БЫТЬ СВЯЗАННЫМИ И СВОБОДНЫМИ. СВЯЗАННОЙ ПЕРЕМЕННОЙ НАЗЫВАЕТСЯ ПЕРЕМЕННАЯ, КОТОРОЙ ПРИСВОЕНО КАКОЕ-ЛИБО ЗНАЧЕНИЕ. ПРИ ВЫЧИСЛЕНИЯХ СВЯЗАННАЯ ПЕРЕМЕННАЯ ЗАМЕНЯЕТСЯ ЕЕ ЗНАЧЕНИЕМ. СВОБОДНОЙ ПЕРЕМЕННОЙ НАЗЫВАЕТСЯ ПЕРЕМЕННАЯ, КОТОРОЙ НЕ ПРИСВОЕНО ЗНАЧЕНИЯ. СВОБОДНАЯ ПЕРЕМЕННАЯ ОБОЗНАЧАЕТ САМУ СЕБЯ И ПРИСУТСТВУЕТ В РЕЗУЛЬТИРУЮЩЕМ ВЫРАЖЕНИИ, ВЫВОДИМОМ СИСТЕМОЙ.

СВЯЗАННУЮ ПЕРЕМЕННУЮ МОЖНО СДЕЛАТЬ СВОБОДНОЙ, ВОСПОЛЬЗОВАВШИСЬ ОПИСАНИЕМ CLEAR (П. 4.4.2).

ЗАМЕЧАНИЕ.

МАТРИЧНЫЕ ПЕРЕМЕННЫЕ РАССМАТРИВАЮТСЯ В ГЛАВЕ 5, ВЕКТОРНЫЕ ПЕРЕМЕННЫЕ - В ГЛАВЕ 6.

2. ЗАРЕЗЕРВИРОВАННЫЕ ПЕРЕМЕННЫЕ.

СЛЕДУЮЩИЕ ПЕРЕМЕННЫЕ В ЯЗЫКЕ REDUCE ИМЕЮТ ПОСТОЯННЫЕ ЗНАЧЕНИЯ, КОТОРЫЕ ПРОГРАММИСТ НЕ МОЖЕТ ИЗМЕНИТЬ ПРОСТО СРЕДСТВАМИ ВХОДНОГО ЯЗЫКА:

I

-МНИМАЯ ЕДИНИЦА. ВСЯКАЯ СТЕПЕНЬ ПЕРЕМЕННОЙ I АВТОМАТИЧЕСКИ ЗАМЕНЯЕТСЯ НА СООТВЕТСТВУЮЩУЮ КОМБИНАЦИЮ I И -1;

E

-ОСНОВАНИЕ НАТУРАЛЬНЫХ ЛОГАРИФМОВ.

T

-ЗАРЕЗЕРВИРОВАННОЕ ИМЯ ФАЙЛА "ТЕРМИНАЛ" ПРИ РАБОТЕ В ИНТЕРАКТИВНОМ РЕЖИМЕ.

ЗАМЕЧАНИЯ.

(1) В ОПЕРАТОРАХ ЦИКЛА, ПРОИЗВЕДЕНИЯ, СУММИРОВАНИЯ ДОПУСКАЕТСЯ ИСПОЛЬЗОВАНИЕ ЗАРЕЗЕРВИРОВАННОЙ ПЕРЕМЕННОЙ I В КАЧЕСТВЕ ПЕРЕМЕННОЙ ЦИКЛА (СМ. 2.11.1).

(2) (ЭТО ЗАМЕЧАНИЕ ОТНОСИТСЯ К ВЕРСИИ СИСТЕМЫ ОТ 15 МАРТА 1981 ГОДА).

В СИСТЕМУ ВСТРОЕНЫ СЛЕДУЮЩИЕ СООТНОШЕНИЯ :

$$E^{**}(I+PI/2)=I$$

$$E^{**}(I+PI)=-1$$

$$E^{**}(3+I+PI/2)=-1$$

2.4 МАССИВЫ.

В ЯЗЫКЕ REDUCE РАЗРЕШЕНЫ МАССИВЫ ПРОИЗВОЛЬНОЙ РАЗМЕРНОСТИ. МАССИВЫ ДОЛЖНЫ БЫТЬ ОПИСАНЫ С ПОМОЩЬЮ ОПИСАНИЯ ARRAY, НАПРИ-

МЕР:

ARRAY A(10), B(2,3,4), C(J)

(ЗДЕСЬ J — ПЕРЕМЕННАЯ ЦЕЛОГО ТИПА).

ИНДЕКСЫ МАССИВОВ ПРОБЕГАЮТ ЗНАЧЕНИЯ ОТ 0 ДО УКАЗАННОГО ПРИ ОПИСАНИИ ЗНАЧЕНИЯ. ОБРАЩАЯСЬ К ЭЛЕМЕНТУ МАССИВА, УКАЗЫВАЮТ ИДЕНТИФИКАТОР МАССИВА И КОНКРЕТНЫЕ ЗНАЧЕНИЯ ИНДЕКСОВ, НАПРИМЕР:

A(2), B(0,0,0), C(1)

ВСЕМ ЭЛЕМЕНТАМ МАССИВА В МОМЕНТ ОПИСАНИЯ ПРИСВАИВАЕТСЯ ПЕР-
ВОНАЧАЛЬНОЕ ЗНАЧЕНИЕ 0,

2.5 ОПЕРАЦИИ.

ВСЕ ОПЕРАЦИИ В ЯЗЫКЕ REDUCE ОПРЕДЕЛЯЮТСЯ КАК ФУНКЦИИ, ПО-
ЭТОМУ В ДАННОМ РУКОВОДСТВЕ ОБА ЭТИ ТЕРМИНА РАВНОПРАВНЫ. ЗАМЕ-
ТИМ, ЧТО АНГЛИЙСКИЙ ТЕРМИН "OPERATOR" МЫ ПЕРЕВОДИЛИ КАК "СИМ-
ВОЛ ОПЕРАЦИИ", ЗАРЕЗЕРВИРОВАВ РУССКОЕ СЛОВО "ОПЕРАТОР" ДЛЯ ПЕ-
РЕВОДА ТЕРМИНА "STATEMENT".

ОПЕРАЦИИ — В ЯЗЫКЕ REDUCE ОПРЕДЕЛЯЮТСЯ ИМЕНЕМ И ТИПОМ, В
ЯЗЫКЕ СУЩЕСТВУЮТ ОПЕРАЦИИ ДВУХ ТИПОВ: ИНФИКСНЫЕ И ПРЕФИКСНЫЕ.

1. ИНФИКСНЫЕ ОПЕРАЦИИ.

СИМВОЛЫ ИНФИКСНЫХ ОПЕРАЦИЙ РАСПОЛОЖЕНЫ МЕЖДУ АРГУМЕНТА-
МИ, НАПРИМЕР:

A + B = C

X = Y AND W MEMBER Z

В СИСТЕМЕ ИМЕЮТСЯ СЛЕДУЮЩИЕ ВСТРОЕННЫЕ ОПЕРАЦИИ:

<ИНФИКСНАЯ ОПЕРАЦИЯ> ::= E OR E AND E NOT E MEMBER E = E NEQ E
>= E > E <= E < E + E = E * E / E **

ЭТИ ОПЕРАЦИИ МОЖНО РАЗБИТЬ НА СЛЕДУЮЩИЕ ПОДКЛАССЫ:

<ОПЕРАЦИЯ ПРИСВАИВАНИЯ> ::= E = E

<ЛОГИЧЕСКАЯ ОПЕРАЦИЯ> ::= E OR E AND E NOT E MEMBER E

<ОПЕРАЦИЯ ОТНОШЕНИЯ> ::= E EQ E NEQ E >= E > E <= E <

<АРИФМЕТИЧЕСКАЯ ОПЕРАЦИЯ> ::= + E = E * E / E **

<СИМВОЛЬНАЯ ОПЕРАЦИЯ> ::= .

ОПЕРАЦИИ В ЭТОМ СПИСКЕ РАСПОЛОЖЕНЫ В ПОРЯДКЕ ВОЗРАСТАНИЯ СТАР-

шинства).

ДЛЯ СОВМЕСТИМОСТИ С ПРОМЕЖУТОЧНЫМ ЯЗЫКОМ, ИСПОЛЬЗУЕМЫМ СИСТЕМОЙ REDUCE, КАЖДОМУ СПЕЦИАЛЬНОМУ ЗНАКУ ИНФИКСНОЙ ОПЕРАЦИИ СОПОСТАВЛЯЕТСЯ ВУКВЕННО-ЦИФРОВОЙ ИДЕНТИФИКАТОР, А ИМЕННО:

=	(=) SETO
=	EQUAL
>=	GEQ
>	GREATERP
<=	LEQ
<	LESSP
+	PLUS
-	DIFFERENCE
	(В СЛУЧАЕ ОДНОМЕСТНОЙ ОПЕРАЦИИ- MINUS)
*	TIMES
/	QUOTIENT
	(В СЛУЧАЕ ОДНОМЕСТНОЙ ОПЕРАЦИИ- RECIP)
**	EXPT
.	CONS

(В ЭТОМ СПИСКЕ ОПЕРАЦИИ РАСПОЛОЖЕНЫ В ПОРЯДКЕ ВОЗРАСТАНИЯ СТАР-
шинства).

В ТЕКСТЕ ПРОГРАММЫ, НАПИСАННОЙ НА ЯЗЫКЕ REDUCE, ЭТИ ИДЕНТИФИКАТОРЫ МОГУТ УПОТРЕБЛЯТЬСЯ ВМЕСТО СООТВЕТСТВУЮЩИХ СПЕЦИАЛЬНЫХ ЗНАКОВ И НАОБОРОТ.

РАССМАТРИВАЕМЫЕ ОПЕРАЦИИ ПРЕДПОЛАГАЮТСЯ ДВУХМЕСТНЫМИ /*/ , ЗА ИСКЛЮЧЕНИЕМ ОПЕРАЦИИ NOT, КОТОРАЯ ЯВЛЯЕТСЯ ОДНОМЕСТНОЙ, А ТАКЖЕ ОПЕРАЦИЯ + И * , ЯВЛЯЮЩИХСЯ МНОГОМЕСТНЫМИ. КРОМЕ ТОГО, ОПЕРАЦИИ - И / МОГУТ ИСПОЛЬЗОВАТЬСЯ КАК В ДВУХМЕСТНОМ, ТАК И В ОДНОМЕСТНОМ ВАРИАНТАХ. ЛЮБАЯ ДРУГАЯ ОПЕРАЦИЯ ПРИ РАЗБОРЕ ВЫРАЖЕНИЯ РАССМАТРИВАЕТСЯ КАК ДВУХМЕСТНАЯ, ПРИЧЕМ ПОРЯДОК ВЫПОЛНЕНИЯ НЕСКОЛЬКИХ ОДИНАКОВЫХ СЛЕДУЮЩИХ ПОДРЯД ДРУГ ЗА ДРУГОМ ОПЕРАЦИЯ-СПРАВА НАЛЕВО. ПРИБЛИЖИТЕЛЬНО ВЫРАЖЕНИЕ A/B/C ИНТЕРПРЕТИРУЕТСЯ КАК (A/B)/C. ИЗ ЭТОГО ПРАВИЛА ИМЕЮТСЯ ТРИ ИСКЛЮЧЕНИЯ, А ИМЕННО: ПОРЯДОК ДЕЙСТВИЯ ПОДРЯД РАСПОЛОЖЕННЫХ ОПЕРАЦИЙ **, := И -СПРАВА НАЛЕВО. ТАК, ВЫРАЖЕНИЕ A**B**C ИНТЕРПРЕТИРУЕТСЯ КАК A**(B**C), А ПРИСВАИВАНИЕ A:=B:=C - КАК A:=(B:=C).

ДЛЯ УКАЗАНИЯ ПОРЯДКА ВЫПОЛНЕНИЯ ОПЕРАЦИЙ МОГУТ ПРИМЕНЯТЬСЯ СКОБКИ. ПРИ ОТСУТСТВИИ СКОБОК ЭТОТ ПОРЯДОК ОПРЕДЕЛЯЕТСЯ СТАР-

* Т.Е. ОПЕРАЦИЯМИ НАД ДВУМЯ ОПЕРАНДАМИ

ШИНСТВОМ ОПЕРАЦИЙ В СООТВЕТСТВИИ С РАЧЕЕ ПРИВЕДЕННЫМИ СПИСКАМИ

$$x = \dots * (\dots)$$

2. ПРЕФИКСНЫЕ ОПЕРАЦИИ

СИМВОЛЫ ПРЕФИКСНЫХ ОПЕРАЦИЙ РАСПОЛАГАЮТСЯ ПЕРЕД СВОИМИ АРГУМЕНТАМИ. АРГУМЕНТЫ ПРЕФИКСНОЙ ОПЕРАЦИИ ЗАДАЮТСЯ В ВИДЕ ЗАКЛЮЧЕННОГО В СКОБКИ СПИСКА ЭЛЕМЕНТОВ, ОТДЕЛЯЕМЫХ ДРУГ ОТ ДРУГА ЗАПЯТОЙ, КАК ПРИ ОБЫЧНОЙ МАТЕМАТИЧЕСКОЙ ЗАПИСИ ФУНКЦИЙ.

НАПРИМЕР:

$$\text{COS}(U)$$

$$\text{DF}(X**2, X)$$

ЕСЛИ ОПЕРАЦИЯ ОДНОМЕСТНАЯ, ТО ЕЕ АРГУМЕНТ МОЖНО НЕ ЗАКЛЮЧАТЬ В СКОБКИ, НАПРИМЕР, ВЫРАЖЕНИЯ ВИДА

$$\text{COS } Y$$

И

$$\text{COS } (Y)$$

ЭКВИВАЛЕНТНЫ И ПРЕДСТАВЛЯЮТ ОДНУ И ТУ ЖЕ ОПЕРАЦИЮ ВЗЯТИЯ КОСИНУСА ОТ АРГУМЕНТА Y . ТАКИЕ ОДНОМЕСТНЫЕ ПРЕФИКСНЫЕ ОПЕРАЦИИ ИМЕЮТ БОЛЕЕ ВЫСОКОЕ СТАРШИНСТВО, ЧЕМ ЛЮБАЯ ИЗ ИНФИКСНЫХ ОПЕРАЦИЙ. В ИСХОДНОМ ТЕКСТЕ ПРОГРАММЫ ИНФИКСНЫЕ ОПЕРАЦИИ МОЖНО ЗАПИСЫВАТЬ ТАКЖЕ И В ПРЕФИКСНОЙ ФОРМЕ, ОДНАКО НА ПЕЧАТЬ ТАКИЕ ОПЕРАЦИИ ВСЕГДА ВЫДАЮТСЯ В ИНФИКСНОЙ ФОРМЕ.

КРОМЕ ПЕРЕЧИСЛЕННЫХ РАНЕЕ ИНФИКСНЫХ ОПЕРАЦИЙ, В СИСТЕМУ ВСТРОЕНЫ НЕКОТОРЫЕ ПРЕФИКСНЫЕ ОПЕРАЦИИ - **DF**, **SUB**, **NUM**, **DEN** И ФУНКЦИИ, Т.Е., СИСТЕМЕ ИЗВЕСТНЫ ИМЕНА ЭТИХ ФУНКЦИЙ И НЕКОТОРЫЕ ИХ СВОЙСТВА. ЭТИ ОПЕРАЦИИ И ФУНКЦИИ РАССМАТРИВАЮТСЯ В СЛЕДУЮЩИХ ПУНКТАХ ЭТОГО ПАРАГРАФА.

3. ВСТРОЕННЫЕ ФУНКЦИИ

В ЭТОМ ПУНКТЕ ПЕРЕЧИСЛЕННЫ ВСТРОЕННЫЕ ФУНКЦИИ, ИМЕЮЩИЕСЯ В ВЕРСИИ СИСТЕМЫ ОТ 15 МАРТА 1981 ГОДА. В БОЛЕЕ РАННИХ ВЕРСИЯХ ВСТРОЕНЫ ТОЛЬКО ФУНКЦИИ **COS**, **LOG**, **SIN**,

В СИСТЕМУ ВСТРОЕНЫ СЛЕДУЮЩИЕ ФУНКЦИИ :

LOG, **SQRT**, **ACOS**, **ACOSH**, **COS**, **COSH**, **ASIN**,
ATAN, **ASINH**, **ATANH**, **SIN**, **TAN**, **SINH**, **TANH**,

СИСТЕМЕ ИЗВЕСТНЫ ИМЕНА ЭТИХ ФУНКЦИЙ И СЛЕДУЮЩИЕ ИХ СВОЙСТВА

1) $\text{LOG}(E) = 1$

$\text{LOG}(1) = 0$

SORT(X)2=X**

2) СВОЙСТВО ЧЕТНОСТИ ($F(-X)=F(X)$) ДЛЯ ЧЕТНЫХ ТРИГОНОМЕТРИЧЕСКИХ ФУНКЦИЙ - ACOS, ACOSH, COS, COSH И СВОЙСТВО НЕЧЕТНОСТИ ДЛЯ НЕЧЕТНЫХ;

3) ЗНАЧЕНИЯ ТРИГОНОМЕТРИЧЕСКИХ ФУНКЦИЙ В НУЛЕ (ЕСЛИ ДАННАЯ ФУНКЦИЯ В НУЛЕ ОПРЕДЕЛЕНА);

4) НЕКОТОРЫЕ ЧАСТНЫЕ СВОЙСТВА ЭТИХ ФУНКЦИЙ :

$$\text{COS}(\pi/2) = 0$$

$$\text{SIN}(\pi/2) = 1$$

$$\text{SIN}(\pi) = 0$$

$$\text{COS}(\pi) = -1$$

$$\text{SIN}(N*\pi) = 0$$

$$\text{COS}(N*\pi) = (-1)**N$$

5) ПРОИЗВОДНЫЕ ЭТИХ ФУНКЦИЙ.

ЗАМЕТИМ ЕЩЕ РАЗ, ЧТО ЗНАЧЕНИЯ ВСТРОЕННЫХ ФУНКЦИЙ ОТ НЕТРИВИАЛЬНЫХ ЧИСЛОВЫХ АРГУМЕНТОВ СИСТЕМЕ НЕИЗВЕСТНЫ.

ПРОГРАММИСТ МОЖЕТ ДОБАВИТЬ ДАЛЬНЕЙШИЕ ПРАВИЛА ДЛЯ ПРЕОБРАЗОВАНИЯ ВЫРАЖЕНИЯ, ВКЛЮЧАЮЩИХ ЭТИ ФУНКЦИИ, ПОМОЩЬ ПОДСТАНОВОК (С. П. 4.1).

4. ОПЕРАЦИЯ DF.

DF - СИМВОЛ ОПЕРАЦИИ ДИФФЕРЕНЦИРОВАНИЯ В ЧАСТНЫХ ПРОИЗВОДНЫХ СКАЛЯРНОЙ ФУНКЦИИ ОДНОЙ ИЛИ НЕКОЛЬКИХ ПЕРЕМЕННЫХ. ПЕРВЫЙ АРГУМЕНТ ОПЕРАЦИИ - ДИФФЕРЕНЦИРУЕМОЕ СКАЛЯРНОЕ ВЫРАЖЕНИЕ, ОСТАЛЬНЫЕ АРГУМЕНТЫ УКАЗЫВАЮТ ПЕРЕМЕННЫЕ ДИФФЕРЕНЦИРОВАНИЯ И ПОРЯДОК СООТВЕТСТВУЮЩЕЙ ПРОИЗВОДНОЙ. СИНТАКСИС ОПЕРАЦИИ ДИФФЕРЕНЦИРОВАНИЯ:

DF (<ВЫРАЖЕНИЕ>, <ПЕРЕМЕННАЯ>, <ЧИСЛО ИЛИ ПЕРЕМЕННАЯ>, ..., <ПЕРЕМЕННАЯ>, <ЧИСЛО ИЛИ ПЕРЕМЕННАЯ>)

<ЧИСЛО ИЛИ ПЕРЕМЕННАЯ> ЗАДАЮТ ПОРЯДОК ПРОИЗВОДНОЙ. ПЕРЕМЕННАЯ, УКАЗЫВАЮЩАЯ ПОРЯДОК, ДОЛЖНО БЫТЬ ПРИСВОЕНО ЦЕЛОЧИСЛЕННОЕ ЗНАЧЕНИЕ.

ЕСЛИ ПОРЯДОК ДИФФЕРЕНЦИРОВАНИЯ ПО НЕКОТОРОЙ ПЕРЕМЕННОЙ РАВЕН 1, ТО ЕГО МОЖНО НЕ ПИСАТЬ ПОСЛЕ СООТВЕТСТВУЮЩЕЙ ПЕРЕМЕННОЙ.

ПРИМЕРЫ ОБОЗНАЧЕНИЙ:

$$\text{DF}(Y, X) = DY/DX$$

$$\text{DF}(Y, X, 2) = D^2 Y/DX^2$$

$$DF(Y, X_1, 2, X_2, X_3, 2) = D \frac{Y}{DX_1} \frac{DX_1}{DX_2} \frac{DX_2}{DX_3}$$

ПРИМЕР:

$$A_1 = X^{**4} + Y * SIN(3 * X * Y)$$

$$DF(A, X);$$

$$4 * X^{**3} + 3 * Y * COS(3 * X * Y)$$

$$DF(A, X, 2);$$

$$3 * (4 * X^{**2} + 3 * Y * SIN(3 * X * Y))$$

$$DF(A, X, 2, Y);$$

$$3 * (27 * Y^{**2} * (X * Y * COS(3 * X * Y) + SIN(3 * X * Y)))$$

ПОДРОБНО ДИФФЕРЕНЦИРОВАНИЕ ВЫРАЖЕНИЯ РАССМАТРИВАЕТСЯ В П. 4.2.

5. ОПЕРАЦИЯ SUB.

SUB * СИМВОЛ ОПЕРАЦИИ ЛОКАЛЬНОЙ ПОДСТАНОВКИ - ОДНОРАЗОВОЙ ЗАМЕНЫ ПЕРЕМЕННЫХ И ПОДВЫРАЖЕНИЙ В ДАННОМ ВЫРАЖЕНИИ. В ОБЩЕМ СЛУЧАЕ ОПЕРАЦИЯ ИМЕЕТ ВИД:

$$SUB \langle \langle \text{СПИСОК ПОДСТАНОВОК} \rangle, \langle \text{ВЫРАЖЕНИЕ} \rangle \rangle$$

НАПРИМЕР:

$$SUB(X=X+1, Y=1, X^{**2}+Y^{**2})$$

ЭТА ПОДСТАНОВКА ДЕЛАЕТ СНАЧАЛА УПРОЩЕНИЕ <ВЫРАЖЕНИЯ>, ЗАТЕМ ЗАМЕНЯЕТ КАЖДУЮ ПЕРЕМЕННУЮ, ВСТРЕЧАЮЩУЮСЯ В СПИСКЕ ПОДСТАНОВОК, И НАКОНЕЦ, УПРОЩАЕТ РЕЗУЛЬТАТ, Т.Е. В ИТОГЕ ПОЛУЧИМ:

$$X^{**2} + 2 * X + 2$$

БОЛЕЕ ПОДРОБНО ПОДСТАНОВКИ РАССМАТРИВАЮТСЯ В П. 4.1.

6. ОПЕРАЦИИ NUM И DEN.

ОПЕРАЦИИ NUM И DEN ВЫДЕЛЯЮТ, СООТВЕТСТВЕННО, ЧИСЛИТЕЛЬ И ЗНАМЕНАТЕЛЬ ВЫРАЖЕНИЯ. НАПРИМЕР, DEN(X/Y^{**2}) ИМЕЕТ ЗНАЧЕНИЕ Y^{**2}, А NUM(X/Y^{**2}) ИМЕЕТ ЗНАЧЕНИЕ X.

7. ВВЕДЕНИЕ НОВЫХ ПРЕФИКСНЫХ ОПЕРАЦИЙ (ФУНКЦИЙ).

ПРОГРАММИСТ МОЖЕТ РАБОТАТЬ НЕ ТОЛЬКО СО ВСТРОЕННЫМИ В СИСТЕМУ ПРЕФИКСНЫМИ ОПЕРАЦИЯМИ И ФУНКЦИЯМИ, НО И С ВВЕДЕННЫМИ ИМ САМИМ. ИМЕНА ЭТИХ НЕИЗВЕСТНЫХ СИСТЕМЕ ПРЕФИКСНЫХ ОПЕРАЦИЙ И ФУНКЦИЙ ДОЛЖНЫ БЫТЬ ОПИСАНЫ ПОСРЕДСТВОМ ОПИСАНИЯ OPERATOR. ТЕРМИН "ОПЕРАТОР" ЗДЕСЬ СЛЕДУЕТ ВОСПРИНИМАТЬ В ЗНАЧЕНИИ "СИМВОЛ, ОБОЗНАЧАЮЩИЙ ОПЕРАЦИЮ" НАПРИМЕР:

```
OPERATOR F, TG;
```

ИМЕНА ВСТРОЕННЫХ ПРЕФИКСНЫХ ОПЕРАЦИЙ СЧИТАЮТСЯ УЖЕ ОПРЕДЕЛЕННЫМИ КАК OPERATOR, НАПРИМЕР:

```
OPERATOR SIN;
***SIN ALREADY DEFINED AS OPERATOR
```

АВТОМАТИЧЕСКИ ОПРЕДЕЛЯЕТСЯ КАК OPERATOR ИМЯ КАЖДОЙ ПРОЦЕДУРЫ, А ТАКЖЕ ИМЕНА ФУНКЦИЙ В ЛЕВОЙ ЧАСТИ ПОДСТАНОВКИ В ПРАВИЛЕ LET.

ПОСЛЕ ОПРЕДЕЛЕНИЯ ИМЕНИ ФУНКЦИИ КАК OPERATOR ПРОГРАММИСТ МОЖЕТ ИСПОЛЬЗОВАТЬ В СВОЕЙ ПРОГРАММЕ ВЫРАЖЕНИЯ, ОБОЗНАЧАЮЩИЕ ДОСТАТОЧНО "ХИТРОЕ" ФУНКЦИОНАЛЬНЫЕ ЗАВИСИМОСТИ И МОЖЕТ ПРИСВАИВАТЬ ЭТИМ ВЫРАЖЕНИЯМ ЗНАЧЕНИЯ.

ОДНУ И ТУ ЖЕ ФУНКЦИЮ МОЖНО ИСПОЛЬЗОВАТЬ С РАЗЛИЧНЫМ ЧИСЛОМ АРГУМЕНТОВ, ВКЛЮЧАЯ 0.

ПРИМЕР:

```
OPERATOR F, TG, CTG;
```

```
G1 := F(F(CTG(X)), F());
G1 := F(F(CTG(X)), F());
```

```
TG(PI) := 0;
TG(PI) := 0
```

```
CTG(X) := 1/TG(X);
CTG(X) := 1/TG(X)
```

```
G1 := G1 + CTG(N+1);
G1 := CTG(N+1) + F(F(1/TG(X)), F());
```

2.6 ЯДРА.

В ПРОЦЕССЕ "ВЫЧИСЛЕНИЯ" (СМ. СХЕМУ В П. 1.1) ВЫРАЖЕНИЕ ПРЕОБРАЗУЕТСЯ СИСТЕМОЙ ВО ВНУТРЕНнюю КАНОНИЧЕСКУЮ ФОРМУ, КОТОРАЯ ИМЕЕТ МАЛО ОБЩЕГО С ПЕРВОНАЧАЛЬНОЙ ФОРМОЙ ВЫРАЖЕНИЯ. ОТНОСИТЕЛЬНО ЭТОГО ПРЕОБРАЗОВАНИЯ СИМВОЛЫ ВСЕХ ОПЕРАЦИЙ В СИСТЕМЕ ДЕЛЯТСЯ НА ДВЕ ГРУППЫ.

СИМВОЛЫ ОПЕРАЦИЙ ПЕРВОЙ ГРУППЫ ЯВНО В КАНОНИЧЕСКОЙ ФОРМЕ УЖЕ НЕ ПРИСУТСТВУЮТ, Т.К. ОНИ ЛИБО СВЕДЕНЫ ПО ИЗВЕСТНЫМ СИСТЕМЕ ПРАВИЛАМ СООТВЕТСТВИЯ К ДРУГИМ, "КАНОНИЧЕСКИМ" СИМВОЛАМ (ЭТО, НАПРИМЕР, "+", "-", "*"), ЛИБО ПРОСТО ВЫПОЛНЕНЫ СИСТЕМОЙ (НАП-

ПРИМЕР, В СЛУЧАЕ ОПЕРАЦИИ DET БУДЕТ ВЫЧИСЛЕН СООТВЕТСТВУЮЩИЙ ДЕТЕРМИНАНТ).

СИМВОЛЫ ОПЕРАЦИЯ ВТОРОЙ ГРУППЫ ПРИСУТСТВУЮТ В КАНОНИЧЕСКОЙ ФОРМЕ ЯВНО. НИКАКИХ ПРАВИЛ СООТВЕТСТВИЯ ДЛЯ ЭТИХ СИМВОЛОВ ОПЕРАЦИЯ СИСТЕМЕ НЕ ЗАДАНО, ВЫПОЛНИТЬ ИХ СИСТЕМА ТОЖЕ "НЕ МОЖЕТ". СЮДА ОТНОСИТСЯ, НАПРИМЕР, ОПЕРАЦИЯ ВЗЯТИЯ КОСИНУСА, ЕСЛИ ПРОГРАММИСТ НЕ ЗАДАЕТ НИКАКИХ ПРАВИЛ, ВЫРАЖАЮЩИХ КОСИНУС ЧЕРЕЗ ДРУГИЕ ФУНКЦИИ (НАПРИМЕР, ЧЕРЕЗ ЭКСПОНЕНТЫ). В КАНОНИЧЕСКОЙ ФОРМЕ СИМВОЛ COS БУДЕТ ПРИСУТСТВОВАТЬ ЯВНО В СОСТАВЕ "ОСТАТОЧНОГО" ВЫРАЖЕНИЯ ТИПА COS X, ТАКИЕ "ОСТАТОЧНЫЕ" ВЫРАЖЕНИЯ НАЗЫВАЮТСЯ "ЯДРАМИ", А ВЫРАЖЕНИЯ, КОТОРЫЕ ПРИВОДЯТ К ЯДРАМ (Т.Е., В ПРОЦЕССЕ УПРОЩЕНИЯ ПРЕОБРАЗУЮТСЯ В ЯДРА) - ЯДЕРНЫМИ ФОРМАМИ, ЯДРА В ПАМЯТИ ХРАНЯТСЯ КАК ПЕРЕМЕННЫЕ.

ПРИМЕРЫ:

A

OOS (X*Y)

LOG (SIN (X))

-ЯДЕРНЫЕ ФОРМЫ, НО:

A*B

(A+B)**4

-УЖЕ НЕГ.

ДЛЯ ТОГО, ЧТОБЫ ИЗМЕНИТЬ ВНУТРЕННИЙ (КАНОНИЧЕСКИЙ) ПОРЯДОК СЛЕДОВАНИЯ ЯДЕР, СЛЕДУЕТ ВОСПОЛЬЗОВАТЬСЯ КОМАНДОЙ KORDER,

KORDER <ЯДРО 1>, ..., <ЯДРО K>;

<ЯДРО 1> НЕ БУДЕТ ИМЕТЬ ВЫШШИЙ ПОРЯДОК В КАКОЙ-ЛИБО ВНУТРЕННЕЙ ФОРМЕ. ПОСЛЕДУЮЩИЕ ОБРАЩЕНИЯ К KORDER АННУЛИРУЮТ ЛЮБОЕ ПРЕДЫДУЩЕЕ.

S**2 * S - X + 1

((S, Y) R - (X**S) (Y* - X**S)

(S, Y) R (S, Y) * (S**S * S**S)

ЯЗЫК REDUCE ОПРЕДЕЛЯЕТСЯ ТАК, ЧТО ОПЕРАТОРЫ (STATEMENTS), КАК И ВЫРАЖЕНИЯ, ИМЕЮТ ЗНАЧЕНИЯ ("ВОЗВРАЩАЮТ ЗНАЧЕНИЯ"). ЕСЛИ КОМАНДА ПРЕДСТАВЛЯЕТ СОБОЙ ВЫРАЖЕНИЕ ИЛИ ОПЕРАТОР (КОНЕЧНО, ОКОНЧИВАЮЩИЕСЯ ОГРАНИЧИТЕЛЯМИ), ТО ОТВЕТОМ СИСТЕМЫ НА ЭТУ КОМАНДУ БУДЕТ ВЫЧИСЛЕННОЕ ЕЮ ЗНАЧЕНИЕ ЭТОГО ВЫРАЖЕНИЯ ИЛИ ОПЕРАТОРА (В ПОСЛЕДНЕМ СЛУЧАЕ, КОНЕЧНО, БУДУТ ВЫПОЛНЕНЫ И СООТВЕТСТВУЮЩИЕ ДЕЙСТВИЯ - НАПРИМЕР, ПРИСВАИВАНИЕ В СЛУЧАЕ ОПЕРАТОРА ПРИСВАИВАНИЯ).

3.1 ВЫРАЖЕНИЯ.

В ЯЗЫКЕ REDUCE УПОТРЕБЛЯЮТСЯ ВЫРАЖЕНИЯ НЕСКОЛЬКИХ ТИПОВ : ЧИСЛОВЫЕ, СКАЛЯРНЫЕ (АЛГЕБРАИЧЕСКИЕ), БУЛЕВСКИЕ, МАТРИЧНЫЕ, ВЕКТОРНЫЕ, ДИРАКОВСКИЕ. ЧИСЛОВЫЕ, СКАЛЯРНЫЕ И БУЛЕВСКИЕ ВЫРАЖЕНИЯ РАССМАТРИВАЮТСЯ В ЭТОЙ ГЛАВЕ, МАТРИЧНЫЕ - В ГЛАВЕ 5, ВЕКТОРНЫЕ И ДИРАКОВСКИЕ - В ГЛ. 5.

1. ЧИСЛОВЫЕ ВЫРАЖЕНИЯ.

ОНИ СОСТОЯТ ИЗ СИНТАКСИЧЕСКИ ДОПУСТИМЫХ КОМБИНАЦИЙ ЦЕЛЫХ ИЛИ ДЕЙСТВИТЕЛЬНЫХ ПЕРЕМЕННЫХ, ЗНАКОВ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ И СКОБОК И ПРИНИМАЮТ ЧИСЛОВЫЕ ЗНАЧЕНИЯ.

ПРИМЕР:

2

$$J * K + 2 * J ** 2$$

ЭТО ЧИСЛОВЫЕ ВЫРАЖЕНИЯ, ЕСЛИ J, K - ПЕРЕМЕННЫЕ ЦЕЛОГО ИЛИ ДЕЙСТВИТЕЛЬНОГО ТИПА.

2. СКАЛЯРНЫЕ (АЛГЕБРАИЧЕСКИЕ) ВЫРАЖЕНИЯ.

СКАЛЯРНЫЕ ВЫРАЖЕНИЯ СОСТОЯТ ИЗ СКАЛЯРНЫХ ПЕРЕМЕННЫХ, ИМЕН ФУНКЦИЙ И СИМВОЛОВ ПРЕФИКСНЫХ ОПЕРАЦИЙ, КРУГЛЫХ СКОБОК И СИМВОЛОВ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ И ПОДЧИНЯЮТСЯ ОБЫЧНЫМ ПРАВИЛАМ СКАЛЯРНОЙ АЛГЕБРЫ.

ПРИМЕРЫ:

X

$$X ** 3 + 2 * Y / (2 * Z ** 2 - DF(X, Z))$$

$$(P ** 2 + M ** 2) ** (1/2) * LOG (Y/M)$$

ВЕЗДЕ ДАЛЕЕ В НАСТОЯЩЕМ РУКОВОДСТВЕ ПОД «ВЫРАЖЕНИЕМ» ПОНИМАЕТСЯ ИМЕННО СКАЛЯРНОЕ ВЫРАЖЕНИЕ, ЕСЛИ КОНТЕКСТ НЕ ТРЕБУЕТ ЧИСЛОВОГО ИЛИ ЛОГИЧЕСКОГО ЗНАЧЕНИЯ.

3. БУЛЕВСКИЕ ВЫРАЖЕНИЯ.

БУЛЕВСКИЕ ВЫРАЖЕНИЯ — ЭТО ВЫРАЖЕНИЯ, ПРИНИМАЮЩИЕ ЗНАЧЕНИЯ ИСТИННОСТИ. В ЯЗЫКЕ REDUCE ЗАРЕЗЕРВИРОВАННОЕ СЛОВО NIL СООТВЕТСТВУЕТ ЗНАЧЕНИЮ "FALSE". ЛЮБОЕ ДРУГОЕ ВЫРАЖЕНИЕ СООТВЕТСТВУЕТ ЗНАЧЕНИЮ "TRUE". В НЕКОТОРОМ СМЫСЛЕ ВСЕ ВЫРАЖЕНИЯ ЯВЛЯЮТСЯ БУЛЕВСКИМИ, Т.К. КАЖДОЕ ВЫРАЖЕНИЕ ПРИНИМАЕТ КАКОЕ-ТО ЗНАЧЕНИЕ. ОДНАКО СОБСТВЕННО БУЛЕВСКИЕ ВЫРАЖЕНИЯ ИМЕЮТ СЛЕДУЮЩУЮ СИНТАКСИЧЕСКУЮ ФОРМУ:

<ВЫРАЖЕНИЕ><ОПЕРАЦИЯ ОТНОШЕНИЯ><ВЫРАЖЕНИЕ>

ИЛИ

<БУЛЕВСКОЕ ВЫРАЖЕНИЕ><ЛОГИЧЕСКАЯ ОПЕРАЦИЯ><БУЛЕВСКОЕ ВЫРАЖЕНИЕ>

БУЛЕВСКИЕ ВЫРАЖЕНИЯ ИСПОЛЗУЮТСЯ В ОСНОВНОМ В УСЛОВНЫХ ОПЕРАТОРАХ И ОПЕРАТОРАХ ЦИКЛА.

ПРИМЕРЫ:

$X < 1$

$X > 0$ OR $X = -2$

ЗАМЕЧАНИЯ.

(1) В ПОСЛЕДОВАТЕЛЬНОСТИ ВЫРАЖЕНИЙ, СОБДИНЕННЫХ AND (OR) ПРОВЕРКА ПРОИЗВОДИТСЯ СЛЕВА НАПРАВО. ЕСЛИ ПЕРВЫЙ ЧЛЕН — FALSE (TRUE), ТО ВЫЧИСЛЕНИЕ ВЫРАЖЕНИЯ ПРЕКРАЩАЕТСЯ.

(2) ЗНАЧЕНИЕ "NIL" СИСТЕМА НЕ ПЕЧАТАЕТ. НАПРИМЕР:

A := 15;

A := 15

B := 20;

B := 20

A < B;

T

A > B;

NOT A < B;

NOT A > B;

A < B OR A > B;

T

A < B AND A > B;

4. РАВЕНСТВА.

ДАЛЕЕ В НАСТОЯЩЕМ РУКОВОДСТВЕ МЫ БУДЕМ НАЗЫВАТЬ РАВЕНСТВАМИ ВЫРАЖЕНИЯ СЛЕДУЮЩЕГО ВИДА:

<ВЫРАЖЕНИЕ>=><ВЫРАЖЕНИЕ>

5. ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ВЫРАЖЕНИЯ В ЯЗЫКЕ REDUCE.

ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ВЫРАЖЕНИЯ В ЯЗЫКЕ REDUCE ПРОИЗВОДИТСЯ СЛЕДУЮЩИМ ОБРАЗОМ. ЕСЛИ ЭТО ЧИСЛОВОЕ ИЛИ БУЛЕВСКОЕ ВЫРАЖЕНИЕ, ТО СИСТЕМА ВЫПОЛНИТ АРИФМЕТИЧЕСКИЕ ИЛИ ЛОГИЧЕСКИЕ ОПЕРАЦИИ НАД СООТВЕТСТВУЮЩИМИ ОПЕРАНДАМИ. РЕЗУЛЬТАТОМ ЯВЛЯЕТСЯ, СООТВЕТСТВЕННО, ЧИСЛОВОЕ ИЛИ БУЛЕВСКОЕ ЗНАЧЕНИЕ.

ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ АЛГЕБРАИЧЕСКОГО (СКАЛЯРНОГО) ВЫРАЖЕНИЯ СОСТОИТ В УПРОЩЕНИИ, ВЫПОЛНЕНИИ ВСТРОЕННЫХ ПРЕФИКСНЫХ ОПЕРАЦИЙ И ВЫПОЛНЕНИИ ПОДСТАНОВОК И ПРИСВАИВАНИЙ.

АВТОМАТИЧЕСКОЕ, Т.Е. ПРОВОДИМОЕ САМОЙ СИСТЕМОЙ УПРОЩЕНИЕ ВЫРАЖЕНИЯ ВКЛЮЧАЕТ РАСКРЫТИЕ СКОБОК, ПРИВЕДЕНИЕ ПОДОБНЫХ, УПОРЯДОЧЕНИЕ ЧЛЕНОВ ВЫРАЖЕНИЯ, ПРИВЕДЕНИЕ К ОБЩЕМУ ЗНАМЕНАТЕЛЮ. ЕСЛИ ПРИ ПРОВЕДЕНИИ УПРОЩЕНИЯ СИСТЕМА ВСТРЕЧАЕТ ДЕЙСТВИТЕЛЬНОЕ ЧИСЛО, ОНА ПРЕДСТАВЛЯЕТ ЕГО В ВИДЕ ОТНОШЕНИЯ ДВУХ ЦЕЛЫХ ЧИСЕЛ, ПЕЧАТАЯ ПРИ ЭТОМ СООТВЕТСТВУЮЩЕЕ СООБЩЕНИЕ. ПРОГРАММИСТ МОЖЕТ ОСУЩЕСТВЛЯТЬ НЕКОТОРЫЙ КОНТРОЛЬ НАД ПРОЦЕССОМ УПРОЩЕНИЯ - СМ. П. 4.6.

ПОД ВЫПОЛНЕНИЕМ ВСТРОЕННЫХ ПРЕФИКСНЫХ ОПЕРАЦИЙ ПОНИМАЕТСЯ ВЫПОЛНЕНИЕ ОПЕРАЦИЙ **DP**, **NUM**, **DEN**, **SUB**, А ТАКЖЕ ПРИМЕНЕНИЕ ИЗВЕСТНЫХ СИСТЕМЕ СВОЙСТВ ОСТАЛЬНЫХ ВСТРОЕННЫХ ОПЕРАЦИЙ (**COS**, **LOG**, **SIN** И Т.Д.).

ПРИ ВЫПОЛНЕНИИ ПОДСТАНОВОК ВЫПОЛНЯЮТСЯ ЗАДАННЫЕ ПОДСТАНОВКИ ДЛЯ ПЕРЕМЕННЫХ И ПОДВЫРАЖЕНИЙ. И, НАКОНЕЦ, ПРИ ВЫПОЛНЕНИИ ПРИСВАИВАНИЯ ВСЕ СВЯЗАННЫЕ ПЕРЕМЕННЫЕ И ЯДЕРНЫЕ ФОРМЫ, КОТОРЫМ ПРИСВОЕНЫ ЗНАЧЕНИЯ, ЗАМЕНЯЮТСЯ ИХ ЗНАЧЕНИЯМИ.

ЗНАЧЕНИЕМ АЛГЕБРАИЧЕСКОГО ВЫРАЖЕНИЯ ЯВЛЯЕТСЯ АЛГЕБРАИЧЕСКОЕ ВЫРАЖЕНИЕ (В ЧАСТНОМ СЛУЧАЕ ЭТО ЗНАЧЕНИЕ МОЖЕТ БЫТЬ ЧИСЛОМ).

3.2 ОПЕРАТОРЫ.

ОПЕРАТОР - ЭТО НЕКОТОРАЯ ДОПУСТИМАЯ КОМБИНАЦИЯ ЗАРЕЗЕРВИРОВАННЫХ СЛОВ И ВЫРАЖЕНИЙ.

К ОСНОВНЫМ ОПЕРАТОРАМ ЯЗЫКА REDUCE ОТНОСЯТСЯ :

- ОПЕРАТОР ПРИСВАИВАНИЯ
- УСЛОВНЫЕ ОПЕРАТОРЫ
- ОПЕРАТОРЫ ЦИКЛА
- ОПЕРАТОР СУММИРОВАНИЯ
- ОПЕРАТОР ПРОИЗВЕДЕНИЯ
- БЛОК
- ГРУППОВЫЙ ОПЕРАТОР
- ОПЕРАТОР GO TO
- ОПЕРАТОР RETURN
- ОПЕРАТОР WRITE.

ВСЕ ЭТИ ОПЕРАТОРЫ РАССМАТРИВАЮТСЯ В НАСТОЯЩЕЙ ГЛАВЕ. КРОМЕ

ЭТИХ ОПЕРАТОРОВ ЕСТЬ ЕЩЕ НЕКОТОРЫЕ, ИСПОЛЬЗУЕМЫЕ В ОСНОВНОМ ВО ВНУТРЕННЕМ ЯЗЫКЕ.

3.3 ОПЕРАТОР ПРИСВАИВАНИЯ.

ОПЕРАТОР ПРИСВАИВАНИЯ ИМЕЕТ СЛЕДУЮЩУЮ СИНТАКСИЧЕСКУЮ ФОРМУ:

<ОПЕРАТОР ПРИСВАИВАНИЯ> ::= <ПЕРЕМЕННАЯ ИЛИ ЯДЕРНАЯ ФОРМА> := <ВЫРАЖЕНИЕ ИЛИ ОПЕРАТОР>

ПРИМЕРЫ:

```

A1 := B + C
N(X,Y) := X - 2*Y
G := TAYLOR(E**X, X, 0, 4)
NDIM := SUB (IND = PNT, N)
N := NUM (EX)

```

```

FOR I := 1 TO PRODUCT I

```

ОПЕРАТОР ПРИСВАИВАНИЯ ВЫПОЛНЯЕТСЯ СЛЕДУЮЩИМ ОБРАЗОМ : СИСТЕМА ВЫЧИСЛЯЕТ ЗНАЧЕНИЕ (СМ. П. 3.1,5) <ВЫРАЖЕНИЯ ИЛИ ОПЕРАТОРА> И ПРИСВАИВАЕТ ЕГО <ПЕРЕМЕННОЙ ИЛИ ЯДЕРНОЙ ФОРМЕ>.

ЗНАЧЕНИЕМ ОПЕРАТОРА ПРИСВАИВАНИЯ ЯВЛЯЕТСЯ ПРИСВАИВАЕМОЕ ЗНАЧЕНИЕ.

В ЯЗЫКЕ ДОПУСТИМЫ МНОГОКРАТНЫЕ ПРИСВАИВАНИЯ ВИДА

<ПЕРЕМЕННАЯ> := <ПЕРЕМЕННАЯ> := ... := <ПЕРЕМЕННАЯ> := <ОПЕРАТОР>

В ЭТОМ СЛУЧАЕ КАЖДОЙ <ПЕРЕМЕННОЙ> ПРИСВАИВАЕТСЯ ЗНАЧЕНИЕ <ОПЕРАТОРА>.

ПРИМЕР.

```

A := X**2;
A := X

```

```

B := Y**2;
B := Y

```

```

C := A := B;
C := A := Y

```

```

D := B := A;

```


$D := B := Y$

КРОМЕ ТОГО, ПРИСВАИВАНИЕ МОЖЕТ БЫТЬ ВСТАВЛЕНО ВНУТРИ ПРОСТОГО ВЫРАЖЕНИЯ, ТАКОГО, КАК, НАПРИМЕР,

$X*(Y:=5)$

ТАКИЕ ПРИСВАИВАНИЯ ОБЯЗАТЕЛЬНО ДОЛЖНЫ ЗАКЛЮЧАТЬСЯ В СКОБКИ. ЧРЕЗМЕРНОЕ УПОТРЕБЛЕНИЕ ТАКИХ КОНСТРУКЦИЙ ПРИВОДИТ К ОШИБКАМ.

ЗАМЕЧАНИЯ.

(1) ПУСТЬ НЕКОТОРАЯ ПЕРЕМЕННАЯ, СКАЖЕМ $A1$, ПРИСВОЕНО ВЫРАЖЕНИЕ, СОДЕРЖАЩЕЕ ПЕРЕМЕННУЮ $B1$. ИЗМЕНИТСЯ ЛИ АВТОМАТИЧЕСКИ ЗНАЧЕНИЕ $A1$ ПРИ ПОСЛЕДУЮЩЕМ ПРИСВАИВАНИИ ЗНАЧЕНИЯ ПЕРЕМЕННОЙ $B1$ - ЗАВИСИТ ОТ ТОГО, БЫЛА ИЛИ НЕТ $B1$ СВЯЗАННОЙ ПЕРЕМЕННОЙ ДО ФОРМИРОВАНИЯ ВЫРАЖЕНИЯ, ПРИСВОЕННОГО $A1$.

ЕСЛИ $B1$ БЫЛА СВЯЗАННОЙ ПЕРЕМЕННОЙ, ТО ИЗМЕНЕНИЕ ЕЕ ЗНАЧЕНИЯ НЕ ОТРАЗИТСЯ НА ЗНАЧЕНИИ $A1$, Т.К. ПРИСУТСТВИЕ $B1$ В $A1$ СТАНОВИТСЯ НЕЗАМЕТНЫМ ПОСЛЕ ТОГО, КАК $B1$ ВНЕСЕТ СВОЙ ВКЛАД В ФОРМИРОВАНИЕ ЗНАЧЕНИЯ $A1$.

ЕСЛИ $B1$ БЫЛА СВОБОДНОЙ ПЕРЕМЕННОЙ, ТОГДА ДЛЯ КАЖДОГО ПОСЛЕДУЮЩЕГО ВЫЧИСЛЕНИЯ $A1$ ИСПОЛЬЗУЕТСЯ ТЕКУЩЕЕ ЗНАЧЕНИЕ $B1$.

ПРИМЕР 1

$B2 := B1$

$B2 := B$

$A1 := B1 + B2$

$A1 := B1 + B$

$B2 := A1$

$B1 := B$

$B1 := A1$

$B1 := A$

$B1 := C1$

$B1 := C$

$A1$

$C + B$

$B1 := A1$

$B1 := A$

$A1$

$A+B$

ЕСЛИ ПРОГРАММИСТУ ТРЕБУЕТСЯ, ЧТОБЫ ПРИСВАИВАНИЕ ЗНАЧЕНИЯ СВОБОДНОЙ ПЕРЕМЕННОЙ ИЗ РАНЕЕ ПРЕДСТАВЛЕННОГО ВЫРАЖЕНИЯ ИМЕЛО "ПЕРМАНЕНТНЫЙ" ЭФФЕКТ, КАК ЕСЛИ БЫ ОНО БЫЛО ВЫПОЛНЕНО ДО ФОРМИРОВАНИЯ ЭТОГО ВЫРАЖЕНИЯ, ТО СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ФУНКЦИЮ SUB (П.5.5). СРАВН. С ПРЕДЫДУЩИМ ПРИМЕРОМ :

$B2 := B1$

$B2 := B$

$A1 := B1 + B2$

```

A1 := B1 + B
C1 := SUB(B1=A, A1);
C1 := B+A
C1)
B + A
A1)
B1 + B
B1 := C;
B1 := C
C1)
B + A
A1)
C + B
    
```

(2) СУЩЕСТВУЕТ ЕЩЕ ОДИН СПОСОБ СДЕЛАТЬ ТАК, ЧТОБЫ ПРИСВАИВАНИЯ ЗНАЧЕНИЯ СВОБОДНОЙ ПЕРЕМЕННОЙ ИЗ РАНЕЕ ОФОРМИРОВАННОГО ВЫРАЖЕНИЯ, А ТАКЖЕ ТЕКУЩИЕ ПОДСТАНОВКИ ПРОИЗВЕЛИ "ПЕРМАНИЕНТНЫЙ" ЭФФЕКТ. ДЛЯ ЭТОГО НУЖНО ИСПОЛЬЗОВАТЬ ОПЕРАТОР ПРИСВАИВАНИЯ В ВИДЕ $X := X$; , НЕПРИВЫЧНОМ ДЛЯ ПРОГРАММИСТА, РАБОТАВШЕГО НА АЛГОЛЕ ИЛИ ФОРТРАНЕ :

```

A1)
A + B
A1 := A1)
A1 := (A + B)
B1 := A;
B1 := A
A1)
A + B
    
```

Т.Е., ОПЕРАТОР ВИДА $X := X$ КАК БЫ ЗАНОВО ФОРМИРУЕТ ВЫРАЖЕНИЕ X И МОЖЕТ ВЫПОЛНИТЬ БОЛЬШУЮ РАБОТУ ПО "ПЕРМАНИЕНТНОМУ" УПРОЩЕНИЮ ВЫРАЖЕНИЯ X (СВЯЗАННОЙ ПЕРЕМЕННОЙ) В ТЕКУЩЕЙ СРЕДЕ, ЗАДАННОЙ ПРИСВАИВАНИЯМИ БЫВШИМ СВОБОДНЫМ ПЕРЕМЕННЫМ, ВХОДЯЩИМ В X , А ТАКЖЕ ПОДСТАНОВКАМИ; ХАРАКТЕРНЫЙ ПРИМЕР :

```

X := <ВЫРАЖЕНИЕ, СОДЕРЖАЩЕЕ НЕОПРЕДЕЛЕННЫЕ КОЭФФИЦИЕНТЫ>;
<ВЫЧИСЛЕНИЕ ЭТИХ КОЭФФИЦИЕНТОВ>;
X := X;
    
```

(3) КОГДА СВОБОДНАЯ ПЕРЕМЕННАЯ (В ПЕРВЫЕ ВОТРЕЧАЕТСЯ (В ЛЕВОЙ ЧАСТИ ОПЕРАТОРА ПРИСВАИВАНИЯ (Т.Е. КОГДА ОНА "СВЯЗЫВАЕТСЯ"), НУЖНО СЛЕДИТЬ ЗА ТЕМ, ЧТОБЫ ЭТА ПЕРЕМЕННАЯ НЕ ВХОДИЛА В ВЫРАЖЕНИЕ В ПРАВОЙ ЧАСТИ ОПЕРАТОРА. НАПРИМЕР, ~~ПУСТЬ ДАЖЕ~~ СВОБОДНАЯ ПЕРЕМЕННАЯ И ЗАПИСАН ОПЕРАТОР $X := X+1$. В ДАЛЬНЕЙШЕМ, ПРИ ВЫЧИСЛЕНИИ КАКОГО-ЛИБО ВЫРАЖЕНИЯ, ВКЛЮЧАЮЩЕГО X , ТО КАЖЕТАСЯ $A := X+B$ ЧИТА СИСТЕМА БУДЕТ ЗАМЕНЯТЬ ЭТУ СВЯЗАННУЮ ПЕРЕМЕННУЮ ЕЕ ЗНАЧЕНИЕМ $X+1$, В КОТОРОМ В СВОЮ ОЧЕРЕДЬ X БУДЕТ ЗАМЕНЕН НА $X+1$ И Т.Д., ЧТО ПРИВЕДЕТ К ПЕРЕПОЛНЕНИЮ СТЕКА.

3.4 УСЛОВНЫЕ ОПЕРАТОРЫ.

УСЛОВНЫЙ ОПЕРАТОР ИМЕЕТ СЛЕДУЮЩУЮ СИНТАКСИЧЕСКУЮ ФОРМУ:

<УСЛОВНЫЙ ОПЕРАТОР> ::= IF <БУЛЕВСКОЕ ВЫРАЖЕНИЕ> THEN <ОПЕРАТОР1> ELSE <ОПЕРАТОР2>;

КОНСТРУКЦИЯ ELSE <ОПЕРАТОР2> ЯВЛЯЕТСЯ НЕОБЯЗАТЕЛЬНОЙ.

УСЛОВНЫЙ ОПЕРАТОР ВЫПОЛНЯЕТСЯ СЛЕДУЮЩИМ ОБРАЗОМ. НАХОДИТСЯ ЗНАЧЕНИЕ <БУЛЕВСКОГО ВЫРАЖЕНИЯ>. ЕСЛИ ЭТО НЕ NIL, ТО ВЫПОЛНЯЕТСЯ <ОПЕРАТОР1>, И ЗНАЧЕНИЕМ УСЛОВНОГО ОПЕРАТОРА В ЭТОМ СЛУЧАЕ ЯВЛЯЕТСЯ ЗНАЧЕНИЕ <ОПЕРАТОРА1>. ЕСЛИ ЗНАЧЕНИЕ <БУЛЕВСКОГО ВЫРАЖЕНИЯ> ЕСТЬ NIL, ВЫПОЛНЯЕТСЯ ЗАТЯГ ELSE, И ЗНАЧЕНИЕМ УСЛОВНОГО ОПЕРАТОРА ЯВЛЯЕТСЯ ЗНАЧЕНИЕ <ОПЕРАТОРА2>.

ПРИМЕРЫ :

IF A=B THEN WRITE "OKAY";

IF SUB(J=N,A)-SUB(N=N,S)=0 AND S+SUB(J=N+1,A)-SUB(N=N+1,S)=0
THEN PROVED ELSE UNPROVED;

3.5 ОПЕРАТОРЫ ЦИКЛА.

ЯЗЫК REDUCE ПРЕДОСТАВЛЯЕТ ВОЗМОЖНОСТЬ ОРГАНИЗОВАТЬ РАЗНООБРАЗНЫЕ ПРОГРАММНЫЕ ЦИКЛЫ, УПРАВЛЯЕМЫЕ КАК ВЫЧИСЛЯЕМЫМ УСЛОВИЕМ, ТАК И ЯВНЫМ СЧЕТЧИКОМ ИЛИ СПИСОКОМ ЗНАЧЕНИЯ.

ЗАМЕЧАНИЯ.

(1) ОПЕРАТОРЫ ЦИКЛА НЕЯВНО ИМЕЮТ СТРУКТУРУ БЛОКА (П. 3.8), ПОЭТОМУ ОПЕРАТОР RETURN (П. 3.3.4) ВНУТРИ ТЕЛ ЭТИХ ОПЕРАТОРОВ МОЖЕТ ПЕРЕДАТЬ УПРАВЛЕНИЕ ТОЛЬКО В ТОЧКУ ПРОГРАММЫ, НЕПОСРЕДСТВЕННО СЛЕДУЮЩУЮ ЗА ЦИКЛОМ.

(2) В ОПЕРАТОРАХ ЦИКЛА, ПРОИЗВЕДЕНИЯ И СУММИРОВАНИЯ ДОПУСКАЕТСЯ ИСПОЛЬЗОВАНИЕ ЗАРЕЗЕРВИРОВАННОЙ ПЕРЕМЕННОЙ I В КАЧЕСТВЕ ПЕРЕМЕННОЙ ЦИКЛА.

1. ОПЕРАТОРЫ WHILE И REPEAT.

ОПЕРАТОРЫ WHILE И REPEAT ОПРЕДЕЛЯЮТ ЦИКЛЫ, УПРАВЛЯЕМЫЕ ВЫЧИСЛЯЕМЫМ УСЛОВИЕМ. ИХ ОБЩАЯ СИНТАКСИЧЕСКАЯ ФОРМА :

WHILE <УСЛОВИЕ> DO <ОПЕРАТОР>;

REPEAT <ОПЕРАТОР> UNTIL <УСЛОВИЕ>;

ГДЕ
<УСЛОВИЕ> ::= <БУЛЕВСКОЕ ВЫРАЖЕНИЕ>

В КАЧЕСТВЕ ПРИМЕРА ПРИВЕДЕМ КОНСТРУКЦИЮ, ЭКВИВАЛЕНТНУЮ ОПЕРАТОРУ FOR...UNTIL...DO....

ПРИМЕР:

```

BEGIN INTEGER INDEX ;
INDEX := INIT ;
IF INCR > 0 THEN WHILE INDEX <= FIN DO <<
<ОПЕРАТОР>,
INDEX := INDEX + INCR >>
ELSE WHILE INDEX > FIN DO <<
<ОПЕРАТОР>,
INDEX := INDEX + INCR >>;
RETURN 0
END
    
```

2. ОПЕРАТОР FOR.

ОПЕРАТОР FOR ОПРЕДЕЛЯЕТ ЦИКЛЫ, УПРАВЛЯЕМЫЕ 1) ЯВНЫМ СЧЕТЧИКОМ ЗНАЧЕНИЯ 2) СЧЕТЧИКОМ И ВЫЧИСЛЯЕМЫМ УСЛОВИЕМ ОДНОВРЕМЕННО.

ЯВНЫМ СЧЕТЧИКОМ ЗНАЧЕНИЯ УПРАВЛЯЕТСЯ ОПЕРАТОР

```

FOR <ПЕРЕМЕННАЯ> := <АРИФМЕТИЧЕСКОЕ ВЫРАЖЕНИЕ> STEP <АРИФМЕТИЧЕСКОЕ ВЫРАЖЕНИЕ> UNTIL <АРИФМЕТИЧЕСКОЕ ВЫРАЖЕНИЕ> DO <ОПЕРАТОР>;
    
```

ЭТОТ ОПЕРАТОР АНАЛОГИЧЕН ФОРТРАННОМУ ОПЕРАТОРУ DO, ЕГО ЗНАЧЕНИЕ РАВНО 0.

ОДНОВРЕМЕННО ЯВНЫМ СЧЕТЧИКОМ ЗНАЧЕНИЯ И ВЫЧИСЛЯЕМЫМ УСЛОВИЕМ УПРАВЛЯЕТСЯ ОПЕРАТОР

```

FOR <ПЕРЕМЕННАЯ> := <АРИФМЕТИЧЕСКОЕ ВЫРАЖЕНИЕ> STEP <АРИФМЕТИЧЕСКОЕ ВЫРАЖЕНИЕ> WHILE <БУЛЕВСКОЕ ВЫРАЖЕНИЕ> DO <ОПЕРАТОР>;
    
```

ЗНАЧЕНИЕ ЭТОГО ОПЕРАТОРА РАВНО 0.

ПОДРАЗУМЕВАЕТСЯ, ЧТО <ПЕРЕМЕННАЯ> В ОПЕРАТОРЕ FOR ИМЕЕТ ТИП "ЦЕЛАЯ". ЕЕ ЗНАЧЕНИЕ ВО ВРЕМЯ ВЫЧИСЛЕНИЯ ОПЕРАТОРА НЕ ЗАВИСИТ ОТ ЕЕ ЗНАЧЕНИЯ ВНЕ ЕГО, ТАК ЧТО БУКВА I МОЖЕТ БЫТЬ ИСПОЛЬЗОВАНА В ТАКОМ КОНТЕКСТЕ, ХОТЯ ОБЫЧНО ОНА ОБОЗНАЧАЕТ МНИМУЮ ЕДИНИЦУ. <АРИФМЕТИЧЕСКОЕ ВЫРАЖЕНИЕ> ТАКЖЕ ДОЛЖНО ПРИНИМАТЬ ЦЕЛОЧИСЛЕННОЕ ЗНАЧЕНИЕ.

ПРИМЕРЫ:

ЗДЕСЬ И В СЛЕДУЮЩИХ ДАЛЕЕ ПОДПУНКТАХ ПРЕДПОЛАГАЕТСЯ, ЧТО ДО ВЫПОЛНЕНИЯ ПРИМЕРОВ В ПРОГРАММЕ УЖЕ БЫЛО ВЫПОЛНЕНО ОПИСАНИЕ

```

ARRAY A(10);
    
```

(1) ЧТОБЫ ПРИСВОИТЬ КАЖДОМУ ЭЛЕМЕНТУ МАССИВА A(I) = ЗНАЧЕНИЕ X**I, МОЖНО ЗАПИСАТЬ:

```

FOR I:=0 STEP 1 UNTIL 10 DO A(I):=X**I
    
```

ОБЩАЯ КОНСТРУКЦИЯ **STEP 1 UNTIL** ДЛЯ УДОБСТВА МОЖЕТ БЫТЬ ЗАМЕНЕНА ДВОЙТОЧЬЕМ, Т.Е. ВМЕСТО ПРИВЕДЕННОГО ПРИМЕРА МОЖНО НАПИСАТЬ:

```
FOR I:=0:10 DO A(I)=X**I;
```

Т.К. ПРИСВАИВАНИЯ В ЭТОМ ВЫЧИСЛЕНИИ ДЕЛАЮТСЯ НЕ НА ВЕРХНЕМ УРОВНЕ ПРОГРАММЫ, РЕЗУЛЬТАТЫ ИХ НЕ ПЕЧАТАЮТСЯ. ЕСЛИ ПРОГРАММИСТ ЗАХОЧЕТ ИХ НАПЕЧАТАТЬ, ОН МОЖЕТ ВОСПОЛЬЗОВАТЬСЯ ОПЕРАТОРОМ **WRITE** СЛЕДУЮЩИМ ОБРАЗОМ:

```
FOR I:=0:10 DO WRITE A(I)=X**I;
```

БОЛЕЕ ПОДРОБНО ПРИМЕНЕНИЕ ОПЕРАТОРА **WRITE** ОПИСАНО В П. 3.9.

(2) ЧТОБЫ ПРИСВОИТЬ ЭЛЕМЕНТУ **A(I)** ЗНАЧЕНИЕ **I!** НУЖНО ВОСПОЛЬЗОВАТЬСЯ КОМАНДОЙ

```
A(0)=1; FOR I:=1:10 DO A(I)=I*A(I-1);
```

3.6 ОПЕРАТОР СУММИРОВАНИЯ.

СИНТАКСИЧЕСКАЯ ФОРМА ЭТОГО ОПЕРАТОРА :

```
FOR <ИНДЕКС> := <НАЧАЛЬНОЕ ЗНАЧЕНИЕ> STEP <ШАГ> UNTIL  
<КОНЕЧНОЕ ЗНАЧЕНИЕ> SUM <ВЫРАЖЕНИЕ>
```

ЗНАЧЕНИЕМ ОПЕРАТОРА СУММИРОВАНИЯ ЯВЛЯЕТСЯ ЗНАЧЕНИЕ "ВЫЧИСЛЕННОЙ" СУММЫ.

ПРИМЕР :

ДЛЯ ПОЛУЧЕНИЯ СУММЫ КВАДРАТОВ ВСЕХ ЧЕТНЫХ ПОЛОЖИТЕЛЬНЫХ ЦЕЛЫХ ЧИСЕЛ ДО 50, МОЖНО ЗАПИСАТЬ:

```
FOR I:=2 STEP 2 UNTIL 50 SUM I**2 ;
```

3.7 ОПЕРАТОР ПРОИЗВЕДЕНИЯ.

СИНТАКСИЧЕСКАЯ ФОРМА ЭТОГО ОПЕРАТОРА :

```
FOR <ИНДЕКС> := <НАЧАЛЬНОЕ ЗНАЧЕНИЕ> STEP <ШАГ> UNTIL  
<КОНЕЧНОЕ ЗНАЧЕНИЕ> PRODUCT <ВЫРАЖЕНИЕ>
```

ПРИМЕР :

ЧТОБЫ ПРИСВОИТЬ X ЗНАЧЕНИЕ 10! , МОЖНО ЗАПИСАТЬ:

```
X := FOR I:=1:10 PRODUCT I;
```

ЗНАЧЕНИЕМ ОПЕРАТОРА ПРОИЗВЕДЕНИЯ ЯВЛЯЕТСЯ ЗНАЧЕНИЕ "ВЫЧИСЛЕННОГО" ПРОИЗВЕДЕНИЯ.

3.8 БЛОК И ГРУППОВОЙ ОПЕРАТОР.

Язык REDUCE ПРЕДОСТАВЛЯЕТ ВОЗМОЖНОСТЬ ГРУППИРОВАТЬ НЕСКОЛЬКО ОПЕРАТОРОВ В БЛОК ИЛИ ГРУППОВОЙ ОПЕРАТОР, БЛОК И ГРУППОВОЙ ОПЕРАТОР МОЖНО ИСПОЛЬЗОВАТЬ В ЛЮБОМ КОНТЕКСТЕ, ГДЕ РАЗРЕШЕН ПРОСТОЙ ОПЕРАТОР, ЗА ИСКЛЮЧЕНИЕМ ПРАВОЙ ЧАСТИ ПРАВИЛА LET.

1. БЛОК.

БЛОК В ЯЗЫКЕ REDUCE ИМЕЕТ СЛЕДУЮЩУЮ СИНТАКСИЧЕСКУЮ ФОРМУ :

<БЛОК> ::= BEGIN <ОПИСАНИЯ ТИПА><ПОСЛЕДОВАТЕЛЬНОСТЬ ОПЕРАТОРОВ> END <ОГРАНИЧИТЕЛЬ>

<ПОСЛЕДОВАТЕЛЬНОСТЬ ОПЕРАТОРОВ> МОЖЕТ ВКЛЮЧАТЬ ПОМЕЧЕННЫЕ ОПЕРАТОР. ПОСЛЕД. МЕТКИ ПЕРЕД ОПЕРАТОРОМ СТАВИТСЯ ДВОЕТОЧИЕ.

ТАКИМ ОБРАЗОМ, ТОЧКОЙ ВХОДА В СОСТАВНОЙ ОПЕРАТОР ИЛИ БЛОК ЯВЛЯЕТСЯ ПЕРВЫЙ ОПЕРАТОР <ПОСЛЕДОВАТЕЛЬНОСТИ ОПЕРАТОРОВ>, ТОЧКОЙ ВЫХОДА - ОПЕРАТОР RETURN (П. 2.11.8) ИЛИ END, ВЫПОЛНЕНИЕ <ПОСЛЕДОВАТЕЛЬНОСТИ ОПЕРАТОРОВ> ЗАВЕРШАЕТСЯ ПО ПЕРВОМУ ДОСТИЖЕНИЮ ТОЧКИ ВЫХОДА. ЗНАЧЕНИЕМ СОСТАВНОГО ОПЕРАТОРА ИЛИ БЛОКА ЯВЛЯЕТСЯ ЗНАЧЕНИЕ СООТВЕТСТВУЮЩЕГО ОПЕРАТОРА RETURN,

ПРИМЕР

```
X := BEGIN INTEGER M;
      M1=10
      L1: IF N=0 THEN RETURN M1
          M1=M+N0
          N1=N-10
          GO TO L1
      END OF BLOCK;
```

ЗДЕСЬ X ПРИСВАИВАЕТСЯ ЗНАЧЕНИЕ M1, ГДЕ N - НАПЕРЕД ЗАДАННОЕ ЦЕЛОЕ ЧИСЛО.

ПЕРЕМЕННЫЕ ВНУТРИ БЛОКА (В ОТЛИЧИЕ ОТ СЛУЧАЯ ГРУППОВОГО ОПЕРАТОРА) ЛОКАЛИЗУЮТСЯ, Т.Е. ЭТИ ПЕРЕМЕННЫЕ ОТЛИЧАЮТСЯ ОТ ЛЮБЫХ ДРУГИХ ПЕРЕМЕННЫХ С ТЕМИ ЖЕ ИМЕНАМИ ВНЕ БЛОКА ИЛИ СОСТАВНОГО ОПЕРАТОРА. ПРЕИМУЩЕСТВО ЛОКАЛИЗАЦИИ ПЕРЕМЕННЫХ ВНУТРИ ТЕЛ ОПЕРАТОРОВ ЗАКЛЮЧАЕТСЯ В ТОМ, ЧТО ПРИ ВЫХОДЕ ИЗ ЭТОГО ОПЕРАТОРА ОСВОБОЖДАЕТСЯ БОЛЬШОЕ КОЛИЧЕСТВО ПАМЯТИ, ЗАНЯТОЕ ПРЕЖДЕ ЗНАЧЕНИЯМИ ЛОКАЛИЗОВАННЫХ ПЕРЕМЕННЫХ.

2. ГРУППОВОЙ ОПЕРАТОР.

ГРУППОВОЙ ОПЕРАТОР ИМЕЕТ СЛЕДУЮЩУЮ СИНТАКСИЧЕСКУЮ ФОРМУ:

<ГРУППОВОЙ ОПЕРАТОР> ::= << <ПОСЛЕДОВАТЕЛЬНОСТЬ ОПЕРАТОРОВ> >>

ГРУППОВЫЕ ОПЕРАТОРЫ ОБЛЕГЧАЮТ РЕАЛИЗАЦИЮ ЗАДАЧ, ВЫПОЛНЕНИЕ КОТОРЫХ ЕСТЕСТВЕННО СВОДИТСЯ К ВЫПОЛНЕНИЮ ПОСЛЕДОВАТЕЛЬНОСТИ ОПЕРАЦИЙ. ЗНАЧЕНИЕМ ГРУППОВОГО ОПЕРАТОРА ЯВЛЯЕТСЯ ЗНАЧЕНИЕ ПОСЛЕДНЕГО ОПЕРАТОРА ГРУППЫ.

ВНУТРИ ГРУППОВОГО ОПЕРАТОРА НЕ РЕКОМЕНДУЕТСЯ ИСПОЛЬЗОВАТЬ ВНЕШНИЕ ПО ОТНОШЕНИЮ К ЭТОМУ ОПЕРАТОРУ ПЕРЕМЕННЫЕ, ИСПОЛЬЗУЕМЫЕ В КАКИХ-ЛИБО ДРУГИХ ВЫРАЖЕНИЯХ, - ЭТО МОЖЕТ ПРИВЕСТИ К ОШИБКЕ.

ЗАМЕЧАНИЯ.

(1) ЕСЛИ ВНУТРИ БЛОКА ИЛИ ГРУППОВОГО ОПЕРАТОРА ДЕЛАЕТСЯ ПРИСВАИВАНИЕ, РЕЗУЛЬТАТ ПРИСВАИВАНИЯ АВТОМАТИЧЕСКИ НА ПЕЧАТЬ НЕ ВЫВОДИТСЯ. ДЛЯ ТОГО, ЧТОБЫ ВЫВЕСТИ ЭТОТ РЕЗУЛЬТАТ, СЛЕДУЕТ ВОСПОЛЬЗОВАТЬСЯ ОПЕРАТОРОМ WRITE.

(2) ВНУТРИ БЛОКОВ И ГРУППОВЫХ ОПЕРАТОРОВ МОЖНО ИСПОЛЬЗОВАТЬ НЕ ТОЛЬКО ОПЕРАТОРЫ, НО И СОБСТВЕННО КОМАНДЫ, НАПРИМЕР, LET. ЕСЛИ КОМАНДА СТОИТ ПОСЛЕДНЕЙ ПЕРЕД >> ИЛИ END, ЕЕ НЕОБХОДИМО ЗАКАНЧИВАТЬ ЗНАКОМ ";". ЕСЛИ ОНА СТОИТ ПОСЛЕДНЕЙ В ГРУППОВОМ ОПЕРАТОРЕ, ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ ЕСТЬ 0.

3. ОПЕРАТОР GO TO.

ОПЕРАТОР GO TO (GOTO) ИСПОЛЬЗУЕТСЯ В БЛОКЕ ДЛЯ БЕЗУСЛОВНОГО ПЕРЕХОДА К ОПЕРАТОРУ С УКАЗАННОЙ МЕТКОЙ. ОПЕРАТОР GO TO ИМЕЕТ СЛЕДУЮЩУЮ СИНТАКСИЧЕСКУЮ ФОРМУ: СЛЕДУЮЩУЮ СИНТАКСИЧЕСКУЮ ФОРМУ:

<ОПЕРАТОР ПЕРЕХОДА> ::= GO TO <МЕТКА>

<МЕТКА> ::= <ИДЕНТИФИКАТОР>

ПОСЛЕ МЕТКИ ПЕРЕД ОПЕРАТОРОМ СТАВИТСЯ ДВОЙТОЧИЕ.

ОГРАНИЧЕНИЕ:

ОПЕРАТОРЫ GOTO МОГУТ НАХОДИТЬСЯ ЛИШЬ ВНУТРИ БЛОКОВ. ИХ НЕ ДОЛЖНО БЫТЬ НА ВЕРХНЕМ УРОВНЕ ПРОГРАММЫ. ДАЛЕЕ, ОНИ МОГУТ ССЫЛАТЬСЯ ТОЛЬКО НА МЕТКУ ВНУТРИ "СВОЕГО" БЛОКА.

4. ОПЕРАТОР RETURN.

ЧАСТО НЕОБХОДИМО ВЫЙТИ ИЗ БЛОКА РАНЬШЕ, ЧЕМ ВСТРЕТИТСЯ СЛОВО END, ОБОЗНАЧАЮЩЕЕ КОНЕЦ БЛОКА, И ПЕРЕДАТЬ ИЗ БЛОКА НА СЛЕДУЮЩИЙ БОЛЕЕ ВЫСОКИЙ ПРОГРАММНЫЙ УРОВЕНЬ ЗНАЧЕНИЕ. ВЫХОД ИЗ БЛОКА, ПЕРЕДАЧУ УПРАВЛЕНИЯ И ЗНАЧЕНИЯ В ТОЧКУ ПРОГРАММЫ, СЛЕДУЮЩУЮ ЗА БЛОКОМ, ОСУЩЕСТВЛЯЕТ ОПЕРАТОР RETURN. ЭТОТ ОПЕРАТОР ИСПОЛЬЗУЕТСЯ В СЛЕДУЮЩЕЙ ФОРМЕ:

RETURN <ВЫРАЖЕНИЕ>

ЗНАЧЕНИЕМ ОПЕРАТОРА ЯВЛЯЕТСЯ ЗНАЧЕНИЕ <ВЫРАЖЕНИЯ>.

ПРИМЕРЫ:

RETURN X+Y

RETURN M

RETURN;

В ПОСЛЕДНЕМ СЛУЧАЕ ОПЕРАТОР ВОЗВРАЩАЕТ ЗНАЧЕНИЕ 0.

ОПЕРАТОР RETURN МОЖЕТ БЫТЬ ТАКЖЕ ИСПОЛЬЗОВАН ВНУТРИ ГРУППОВОГО ОПЕРАТОРА ИЛИ ОПЕРАТОРА ЦИКЛА.

ОПЕРАТОР RETURN МОЖЕТ НАХОДИТЬСЯ ЛИШЬ ВНУТРИ СОСТАВНОГО ОПЕРАТОРА, БЛОКА ИЛИ ОПЕРАТОРА ЦИКЛА. ЕСЛИ СИСТЕМА ВСТРЕЧАЕТ ОПЕРАТОР RETURN НА ВЕРХНЕМ УРОВНЕ ПРОГРАММЫ, ТО ПЕЧАТАЕТСЯ СООБЩЕНИЕ

***RETURN INVALID

А ВЫПОЛНЕНИЕ ПРОГРАММЫ ПРЕКРАЩАЕТСЯ.

3.9 ОПЕРАТОР WRITE.

ЧАСТО ПОЛЕЗНО ЗАПИСАТЬ ЗАГОЛОВОК ИЛИ КОММЕНТАРИЙ, ИЛИ КАК-ТО ПО-ОСОБОМУ НАЗВАТЬ ВЫРАЖЕНИЕ ПРИ ВЫВОДЕ. В ЯЗЫКЕ REDUCE ЭТО ВОЗМОЖНО ОСУЩЕСТВИТЬ С ПОМОЩЬЮ ОПЕРАТОРА WRITE, ЕГО СИНТАКСИЧЕСКАЯ ФОРМА:

WRITE <ВЫРАЖЕНИЕ>, ..., <ВЫРАЖЕНИЕ>

ГДЕ ПОД <ВЫРАЖЕНИЕМ> ПОНИМАЕТСЯ АЛГЕБРАИЧЕСКОЕ ВЫРАЖЕНИЕ ИЛИ СТРОКА (СТРОКА - ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ, ЗАКЛЮЧЕННАЯ В КАВЫЧКИ, НАПРИМЕР, "A STRING").

ПРИ ВЫПОЛНЕНИИ ОПЕРАТОРА ЗНАЧЕНИЯ <ВЫРАЖЕНИЯ> ВЫЧИСЛЯЮТСЯ (ИЛИ ПЕЧАТАЮТСЯ НА ВЫВОДЕ В ОДНУ СТРОЧКУ, БЕЗ ПРОБЕЛОВ, СТРОКИ ПЕЧАТАЮТСЯ В ТОЧНОСТИ ТАК, КАК БЫЛИ ВВЕДЕНЫ, ИСКЛЮЧАЮТСЯ ТОЛЬКО ТЕ СИМВОЛЫ, КОТОРЫЕ ПРОИГНОРИРОВАЛО ВВОДНОЕ УСТРОЙСТВО.

ЗНАЧЕНИЕ ОПЕРАТОРА WRITE - ЗНАЧЕНИЕ ЕГО ПОСЛЕДНЕГО АРГУМЕНТА.

ЗАМЕЧАНИЕ.

ОПЕРАТОР WRITE УДОБНО ИСПОЛЬЗОВАТЬ ВНУТРИ БЛОКОВ ИЛИ ГРУППОВЫХ ОПЕРАТОРОВ - МОЖНО ВЫВЕСТИ НА ПЕЧАТЬ РЕЗУЛЬТАТЫ ПРИСВАИВАНИЙ. НАПОМНИМ, ЧТО РЕЗУЛЬТАТЫ ЭТИХ "ВНУТРЕННИХ" ПРИСВАИВАНИЙ АВТОМАТИЧЕСКИ НА ПЕЧАТЬ НЕ ВЫВОДЯТСЯ).

ПРИМЕРЫ:

(1) (ПРИМЕР ИЗ [10])

```
PROCEDURE RR(X,Y,S)
  BEGIN REAL Z;
  FOR I := 1:Y DO Z := WRITE (S+X**I);
  RETURN Z;
END;
```

RR

RR (3,2,15)

18
24
0

(2) ПРИВЕДЕННАЯ ПРОГРАММА ВЫЧИСЛЯЕТ ИЗВЕСТНЫЕ F И G РЯДЫ ИЗ [11].

```

X1I = SIG*(MU+2*EPS)
X2I = EPS+2*SIG**2
X3I = 3*MU*SIG
F1 = 1
G1 = 0
FOR II=1 STEP 1 UNTIL 10 DO BEGIN
  F1I = -MU*G + X1*DF(F, EPS) + X2*DF(F, SIG) + X3*DF(F, MU)
  WRITE "F(", I, ") = ", F1I
  G1I = F + X1*DF(G, EPS) + X2*DF(G, SIG) + X3*DF(G, MU)
  WRITE "G(", I, ") = ", G1I
  F1 = F1I
  G1 = G1I
END)

```

ДЛЯ ИЛЛЮСТРАЦИИ ПЕЧАТИ ПО КОМАНДЕ WRITE РАССМОТРИМ ЧАСТЬ ВЫВОДА:

...<НАЧАЛО ВЫВОДА>...

$F(4) = MU*(3*EPS + 15*SIG^2 + MU)$

$G(4) = G*SIG*MU$

$F(5) = 15*SIG*MU*(3*EPS + 7*SIG^2 + MU)$

$G(5) = MU*(9*EPS + 45*SIG^2 + MU)$

...<ПОСЛЕДУЮЩИЙ ВЫВОД>...

3.10 ПРОЦЕДУРЫ.

РАССМОТРИМ БОЛЕЕ СЛОЖНЫЕ СТРУКТУРЫ, ЧЕМ ПРОСТЫЕ ОПЕРАТОРЫ И ВЫРАЖЕНИЯ - ПРОЦЕДУРЫ.

ЧАСТО БЫВАЕТ УДОБНО ДАТЬ ОПЕРАТОРУ ИМЯ ДЛЯ НЕОДНОКРАТНОГО ИСПОЛЬЗОВАНИЯ ЕГО В ВЫЧИСЛЕНИЯХ С РАЗЛИЧНЫМИ ПАРАМЕТРАМИ, ИЛИ ДЛЯ ОПРЕДЕЛЕНИЯ ПОЛНОГО ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА, РЕАЛИЗУЮЩЕГО НЕКОТОРЫЙ ОПЕРАТОР. ДЛЯ ЭТОЙ ЦЕЛИ В ЯЗЫКЕ REDUCE ИМЕЮТСЯ ОПИСАНИЯ ПРОЦЕДУР. ИХ ОБЩИЙ СИНТАКСИЧЕСКИЙ ВИД:

<ТИП ПРОЦЕДУРЫ> PROCEDURE <ИМЯ ПРОЦЕДУРЫ> <СПИСОК ПЕРЕМЕННЫХ>; <ОПЕРАТОР>;

ГДЕ

<СПИСОК ПЕРЕМЕННЫХ> ::= (<ПЕРЕМЕННАЯ>, ..., <ПЕРЕМЕННАЯ>)

ДОПУСКАЮТСЯ СЛЕДУЮЩИЕ ТИПЫ ПРОЦЕДУР:

REAL, INTEGER, ALGEBRAIC. ЕСЛИ ТИП ПРОЦЕДУРЫ НЕ УКАЗАН, ТО

СЧИТАЕТСЯ, ЧТО ОНА ИМЕЕТ ТИП **ALGEBRAIC** , В СУЩЕСТВУЮЩЕЙ ВЕРСИИ СИСТЕМЫ НЕ ДЕЛАЕТСЯ НИКАКИХ РАЗЛИЧИЙ ПРИ ОБРАБОТКЕ ЭТИХ ТРЕХ ТИПОВ, ХОТЯ В БОЛЕЕ ПОЗДНИХ ВЕРСИЯХ МОГУТ БЫТЬ РАЗЛИЧИЯ.

ОПИСАНИЕ ПРОЦЕДУРЫ АВТОМАТИЧЕСКИ ОПРЕДЕЛЯЕТ ИМЯ ПРОЦЕДУРЫ КАК **OPERATOR** .

ЗНАЧЕНИЕМ ОПИСАНИЯ ПРОЦЕДУРЫ ЯВЛЯЕТСЯ **<ИМЯ ПРОЦЕДУРЫ>** СИСТЕМА ПЕЧАТАЕТ ЕГО СРАЗУ ПОСЛЕ ОПИСАНИЯ ПРОЦЕДУРЫ (СМ. ПРИБИМЕР (1) К П. 3.9). ЗНАЧЕНИЕМ ПРОЦЕДУРЫ (Т.Е. ВЫЗОВА ПРОЦЕДУРЫ) ЯВЛЯЕТСЯ ЗНАЧЕНИЕ **<ОПЕРАТОРА>**.

ЗАМЕЧАНИЯ.

(1) **<ПЕРЕМЕННЫЕ>** В ОПИСАНИИ ПРОЦЕДУРЫ ЯВЛЯЮТСЯ ФОРМАЛЬНЫМИ ПАРАМЕТРАМИ, КОТОРЫЕ ОТЛИЧАЮТСЯ ОТ ПЕРЕМЕННЫХ С ТЕМИ ЖЕ ИМЕНАМИ ВНЕ ОПИСАНИЯ ПРОЦЕДУРЫ. СООТВЕТСТВУЮЩИЕ ФАКТИЧЕСКИЕ ПАРАМЕТРЫ МОГУТ БЫТЬ ПРОИЗВОЛЬНЫМИ ВЫРАЖЕНИЯМИ.

(2) ПРОЦЕДУРА МОЖЕТ БЫТЬ ВЫПОЛНЕНА В ЛЮБОЙ МОМЕНТ ПОСЛЕ ПОЯВЛЕНИЯ ОПИСАНИЯ ПРОЦЕДУРЫ, ДО ТЕХ ПОР, ПОКА ОНА НЕ ОЧИЩЕНА.

(3) ЯЗЫК ДОПУСКАЕТ ИСПОЛЬЗОВАНИЕ ПРОЦЕДУР БЕЗ ПАРАМЕТРОВ, В ОПИСАНИИ ТАКОЙ ПРОЦЕДУРЫ **<СПИСОК ПЕРЕМЕННЫХ>** МОЖНО ОПУСТИТЬ, ОДНАКО ВЫЗОВ ТАКОЙ ПРОЦЕДУРЫ ДОЛЖЕН ИМЕТЬ ВИД **<ИМЯ ПРОЦЕДУРЫ>()**;

ЕСЛИ НЕ БУДЕТ СКОБОК, ПРОЦЕДУРА ВЫПОЛНЕНА НЕ БУДЕТ БЕЗ ВСЯКОГО СООБЩЕНИЯ ОБ ЭТОМ.

(4) В ЯЗЫКЕ **REDUCE** ДОПУСКАЮТСЯ РЕКУРСИВНЫЕ ПРОЦЕДУРЫ, И РАБОТАЮТ ОНИ ВЕСЬМА ЭФФЕКТИВНО.

ПРИМЕРЫ:

(1) ПРИМЕР ИЗ П.2.11.5 МОЖЕТ БЫТЬ ВЫПОЛНЕН С ПОМОЩЬЮ ПРОЦЕДУРЫ ЦЕЛОГО ТИПА **FAC** :

```
INTEGER PROCEDURE FAC (N) ;
BEGIN INTEGER M;
  M:=1;
  L1: IF N=0 THEN RETURN M;
  M:=M*N;
  N:=N-1;
  GO TO L1
END;
```

ЕСЛИ ТЕПЕРЬ МЫ БУДЕМ ВЫЧИСЛЯТЬ ВЫРАЖЕНИЕ **FAC (3)** , ТО ПОЛУЧИМ В РЕЗУЛЬТАТЕ **6**.

(2) В КАЧЕСТВЕ ПРИМЕРА АЛГЕБРАИЧЕСКОЙ ПРОЦЕДУРЫ ОПРЕДЕЛИМ ОПЕРАЦИЮ **P** ОТ ДВУХ АРГУМЕНТОВ, ВЫЧИСЛЯЮЩУЮ ПОЛИНОМЫ ЛЕЖАНДРА, МЫ ОПРЕДЕЛЯЕМ ЭТУ ОПЕРАЦИЮ, ИСХОДЯ ИЗ ФОРМУЛЫ:

$$P(N) = \frac{1}{N!} \frac{d^N}{dy^N} \left((y^2 + x + y + 1)^{1/2} \right) \quad [\quad y=0]$$

В ЯЗЫКЕ REDUCE ЭТА ПРОЦЕДУРА ИМЕЕТ ВИД:

```
ALGEBRAIC PROCEDURE P(N,X);
SUB(Y=0,DF((Y**2-2*X*Y+1)**(-1/2),Y,N))/(FOR I:=1IN
PRODUCT I)W
```

БЛАГОДАря ЭТОМУ ОПРЕДЕЛЕНИЮ ВЫЧИСЛЕНИЕ ВЫРАЖЕНИЯ

$$2 * P(2, W)$$

ДАСТ В РЕЗУЛЬТАТЕ

$$\frac{2}{3 * W + 1}$$

МЫ МОЖЕМ, КОНЕЧНО, ОПУСТИТЬ СЛОВО ALGEBRAIC В ДАННОМ ОПИСАНИИ ПРОЦЕДУРЫ, Т.К. НЕОПИСАННЫЙ ТИП КАК РАЗ И ЕСТЬ ALGEBRAIC.

ДЛЯ ТОГО, ЧТОБЫ ДАТЬ ПРОГРАММИСТАМ ВОЗМОЖНОСТЬ СРАВНИТЕЛЬНО ЛЕГКО ВОЗДЕЙСТВОВАТЬ НА САМУ СИСТЕМУ, СИСТЕМНЫЕ ПРОЦЕДУРЫ НЕ ЗАЩИЩЕНЫ ОТ ПЕРЕОПРЕДЕЛЕНИЯ ИХ ПРОГРАММИСТОМ. ЕСЛИ НЕКОТОРАЯ ПРОЦЕДУРА ПЕРЕОПРЕДЕЛЯЕТСЯ, ТО ПЕЧАТАЕТСЯ СООБЩЕНИЕ

***СИМЯ ПРОЦЕДУРЫ> REDEFINED

В ПОДОБНЫХ СЛУЧАЯХ, ЕСЛИ ПЕРЕОПРЕДЕЛЯЕТСЯ СИСТЕМНАЯ ПРОЦЕДУРА, А НЕ ПРОЦЕДУРА, ОПРЕДЕЛЕННАЯ САМИМ ПРОГРАММИСТОМ В ЕГО ПРОГРАММЕ, ТО ЕМУ НАСТОЯТЕЛЬНО РЕКОМЕНДУЕТСЯ ПЕРЕИМЕНОВАТЬ ВНОВЬ ОПРЕДЕЛЕННУЮ ПРОЦЕДУРУ.

НАПОМИМ, ЧТО К НЕВЫЧИСЛЯЕМЫМ КОМАНДАМ ОТНОСЯТСЯ КОМАНДЫ, НЕ ТРЕБУЮЩИЕ ОТВЕТА СИСТЕМЫ.

4.1 КОМАНДЫ ПОДСТАНОВОК,

ВАЖНЫЙ КЛАСС КОМАНД В ЯЗЫКЕ REDUCE СОСТАВЛЯЮТ КОМАНДЫ, ОПРЕДЕЛЯЮЩИЕ ПОДСТАНОВКИ ДЛЯ ПЕРЕМЕННЫХ И ВЫРАЖЕНИЙ, КОТОРЫЕ МОГУТ БЫТЬ СДЕЛАНЫ В ПРОЦЕССЕ ПРЕОБРАЗОВАНИЯ ВЫРАЖЕНИЯ. ТАКИЕ ПОДСТАНОВКИ ГЛОБАЛЬНО ОБЪЯВЛЯЮТСЯ КОМАНДОЙ LET, А ЛОКАЛЬНО — ПОСРЕДСТВОМ ОПЕРАЦИИ SUB.

1. КОМАНДА LET

КОМАНДА LET ИСПОЛЬЗУЕТСЯ В ФОРМЕ:

LET <СПИСОК ПОДСТАНОВОК>;

ГДЕ <СПИСОК ПОДСТАНОВОК> — СПИСОК РАВЕНСТВ В ВИДЕ $(X) = (Y, X) \text{ ТРИ } Y, X \text{ ПЛА } Y$

<ПЕРЕМЕННАЯ> = <ВЫРАЖЕНИЕ> $S ** Y * S = (Y) \text{ ТРИ } O < Y \text{ ТАК } \text{НОУ} < Y \text{ ПЛА } Y$

ИЛИ

<ПРЕФИКСНАЯ ОПЕРАЦИЯ>(<АРГУМЕНТ>, ..., <АРГУМЕНТ>) = <ВЫРАЖЕНИЕ>

ИЛИ

<АРГУМЕНТ><ИНФИКСНАЯ ОПЕРАЦИЯ>, ..., <АРГУМЕНТ> = <ВЫРАЖЕНИЕ>

ПОДСТАНОВКИ В <СПИСКЕ ПОДСТАНОВОК> РАЗДЕЛЯЮТСЯ ЗАПЯТЫМИ,

ПРИМЕРЫ:

LET X = Y**2 + 2,

N(X, Y) = X - Y,

COS(60) = 1/2,

Y**3 = 2*2 = 3!

ЭТИ ПОДСТАНОВКИ БУДУТ ТЕПЕРЬ СДЕЛАНЫ ДЛЯ ВСЕХ ТАКИХ ПЕРЕМЕННЫХ И ВЫРАЖЕНИЙ, ПОЯВЛЯЮЩИХСЯ В ВЫЧИСЛЕНИЯХ. КАЖДЫЙ СИМВОЛ ОПЕРАЦИИ, ПОЯВЛЯЮЩИЙСЯ В ЭТИХ РАВЕНСТВАХ, АВТОМАТИЧЕСКИ ОПРЕДЕЛЯЕТСЯ СИСТЕМОЙ КАК OPERATOR.

В КАЖДОМ ИЗ ЭТИХ ПРИМЕРОВ ПОДСТАНОВКИ ДЕЛАЮТСЯ ТОЛЬКО ДЛЯ ЯВНО ЗАДАННЫХ ВЫРАЖЕНИЙ, Т.Е. НИ ОДНА ИЗ ПЕРЕМЕННЫХ НЕ МОЖЕТ РАССМАТРИВАТЬСЯ КАК ФОРМАЛЬНЫЙ ПАРАМЕТР. К ПРИМЕРУ, КОМАНДА

LET N(X, Y) = X - Y;

ЗАМЕНИТ N(X, Y) НА X - Y ОДНАКО, ТАКАЯ ПОДСТАНОВКА НЕ БУДЕТ ВЫПОЛНЕНА ДЛЯ N(X, Z) ИЛИ ФУНКЦИИ N ОТ КАКИХ-ЛИБО ДРУГИХ АРГУМЕНТОВ.

ЕСЛИ ТРЕБУЕТСЯ ПОДСТАНОВКА ДЛЯ ВСЕХ ДОПУСТИМЫХ ЗНАЧЕНИЙ ДАННОГО АРГУМЕНТА ОПЕРАЦИИ, ТО МОЖНО ИСПОЛЬЗОВАТЬ ОПИСАНИЕ FOR ALL (ИЛИ FORALL)

СИНТАКСИС ЭТОЙ КОМАНДЫ:

FOR ALL <СВОБОДНЫЕ ПЕРЕМЕННЫЕ> SUCH THAT <БУЛЕРСКОЕ ВЫРАЖЕНИЕ> LET <СПИСОК ПОДСТАНОВОК>;

<СВОБОДНЫЕ ПЕРЕМЕННЫЕ> ::= <ПЕРЕМЕННАЯ>, ..., <ПЕРЕМЕННАЯ>

КОНСТРУКЦИЯ SUCH THAT <БУЛЕРСКОЕ ВЫРАЖЕНИЕ> ЯВЛЯЕТСЯ НЕОБЯЗАТЕЛЬНОЙ.

ПРИМЕР:

FOR ALL X, Y LET H(X, Y) = X + Y;

FOR ALL X LET K(X, Y) = X * Y;

FOR ALL Y SUCH THAT Y > 0 LET F(Y) = 2 * Y ** 2;

ЗАМЕТИМ, ЧТО КАЖДАЯ ПЕРЕМЕННАЯ, УКАЗАННАЯ В FORALL, ЯВЛЯЕТСЯ ФОРМАЛЬНЫМ ПАРАМЕТРОМ, КОТОРЫЙ ОТЛИЧАЕТСЯ ОТ ДРУГИХ, РАНЕЕ ИЛИ ВПОСЛЕДСТВИИ ВВЕДЕННЫХ ПЕРЕМЕННЫХ ИЛИ ФОРМАЛЬНЫХ ПАРАМЕТРОВ, ИМЕЮЩИХ ТЕ ЖЕ САМЫЕ ИМЕНА ВНЕ ЭТОГО ПРАВИЛА.

ПРИ ПРИМЕНЕНИИ ЛЮБЫХ ПОДСТАНОВОК, ВВЕДЕННЫХ ПОСРЕДСТВОМ LET, СИСТЕМА ПРОСМАТРИВАЕТ ПОДСТАНОВОЧНЫЕ ВЫРАЖЕНИЯ В ПОИСКАХ КАКИХ-ЛИБО ВЫРАЖЕНИЙ, КОТОРЫЕ МОГУТ САМИ ИМЕТЬ ОБЪЯВЛЕННЫЕ ДЛЯ НИХ ПОДСТАНОВКИ. ТАКИМ ОБРАЗОМ, LET УСТАНОВЛИВАЕТ ЭКВИВАЛЕНТНОСТЬ МЕЖДУ ЛЕВОЙ И ПРАВОЙ ЧАСТЯМИ ПОДСТАНОВКИ В ОТЛИЧИЕ ОТ ОПЕРАТОРОВ ПРИСВАИВАНИЯ. ДРУГИМИ СЛОВАМИ, ПОДСТАНОВКА ТАКОГО ВИДА, КАК

LET X = X + 1;

НЕ РАЗРЕШАЕТСЯ, Т.К. В ПОДСТАНОВКЕ ДЛЯ X СИСТЕМА ОБНАРУЖИТ, ЧТО ПЕРЕМЕННАЯ X, НАХОДЯЩАЯСЯ В ПОДСТАНОВОЧНОМ ВЫРАЖЕНИИ, САМА ИМЕЕТ ПОДСТАНОВКУ, И РЕЗУЛЬТАТОМ БУДЕТ ОШИБКА ПО ПЕРЕПОЛНЕНИЮ СТЕКА. АНАЛОГИЧНО, ПАРА ПОДСТАНОВОК

LET L = M + N, N = L + R;

НЕ ДОПУСТИМА.

КРОМЕ ТОГО, ПРАВИЛО LET В ПРОСТЕЙШЕЙ ФОРМЕ "LET <ПЕРЕМЕННАЯ> = ..." ОТЛИЧАЕТСЯ ОТ СООТВЕТСТВУЮЩЕГО ПРИСВАИВАНИЯ "<ПЕРЕМЕННАЯ> := ..."

ЕЩЕ И ТЕМ, ЧТО ПРАВИЛО LET БУДУЧИ ЗАДАНО СИСТЕМЕ, НЕ ЗАМЕШАЕТ ПЕРЕМЕННУЮ ЕЕ ЗНАЧЕНИЕМ ДО ТЕХ ПОР, ПОКА ПОЛЬЗОВАТЕЛЬ НЕ ВВОДИТ ВЫРАЖЕНИЕ, СОДЕРЖАЩЕЕ ЭТУ ПЕРЕМЕННУЮ.

ПРИМЕР:

(СРАВН. С ПРИМЕРОМ К ЗАМЕЧАНИЮ (1) П.3.3).

E2 I = F1

E2 := F

LET E1 = E1 + E2

F1;

E1 + F

E2 := G;

E2 := G

F1;

E1 + G

$$; F = S*(X)M^2 + S*(X)200 \quad ; F = S*(X)M^2 + S*(X)200 \quad ; F = S*(X)M^2 + S*(X)200 \quad ; F = S*(X)M^2 + S*(X)200$$

2. ОПЕРАЦИЯ SUB.

ЕСЛИ ПРОГРАММИСТ ХОЧЕТ ПРОСТО ЗАМЕНИТЬ КАЖДУЮ ВХОДЯЩУЮ ПЕРЕМЕННУЮ НА ВЫРАЖЕНИЕ БЕЗ ПОСЛЕДУЮЩЕГО ПРОСМОТРА ЭТОГО ВЫРАЖЕНИЯ НА ДАЛЬНЕЙШИЕ ПОДСТАНОВКИ, ТО МОЖНО ВОСПОЛЬЗОВАТЬСЯ ОПЕРАЦИЕЙ SUB (СМ. П. 2.4.5).

ПОДЧЕРКНЕМ ОТЛИЧИЯ ОПЕРАЦИИ SUB ОТ LET - ПОДСТАНОВКИ

1) SUB ВЫПОЛНЯЕТ ПОДСТАНОВКУ ЛОКАЛЬНО, В ОДНОМ ВЫРАЖЕНИИ, LET "ВКЛЮЧАЕТ" ПОДСТАНОВКИ, КОТОРЫЕ БУДУТ ПРИМЕНЯТЬСЯ КО ВСЕМ ВЫРАЖЕНИЯМ ДО ТЕХ ПОР, ПОКА НЕ БУДУТ УНИЧТОЖЕНЫ КОМАНДОЙ CLEAR (П. 4.4.2).

2) SUB МОЖЕТ ВЫПОЛНЯТЬ ТОЛЬКО ЗАМЕНУ ПРОСТОЙ ПЕРЕМЕННОЙ ВЫРАЖЕНИЕМ, LET ДОПУСКАЕТ ШИРОКИЙ КЛАСС ЛЕВЫХ ЧАСТЕЙ ПОДСТАНОВОК, ВОПЛНЕ ДОПУСТИМА, НАПРИМЕР, ПОДСТАНОВКА SUB(X := X, Y); А ПРАВИЛО LET ПРИВОДИТ К ТОМУ, ЧТО ПОДСТАВЛЕННОЕ ВЫРАЖЕНИЕ СНОВА ПРОСМАТРИВАЕТСЯ В ПОИСКАХ СООТВЕТСТВИЯ ЛЕВЫМ ЧАСТЯМ ПОДСТАНОВОК ДО ТЕХ ПОР, ПОКА ПОДСТАНОВКИ НЕ СТАНУТ НЕВОЗМОЖНЫМИ, ТАК ЧТО ПОДСТАНОВКА ТИПА LET X := X; ПРИВЕДЕТ К НЕОГРАНИЧЕННОЙ РЕКУРСИИ.

ВЕЗДЕ, ГДЕ ЭТО ВОЗМОЖНО, СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ОПЕРАЦИЮ SUB ВМЕСТО ПРАВИЛА LET, Т.К. ЭТО ЗНАЧИТЕЛЬНО ЭФФЕКТИВНЕЕ.

3. ПОДСТАНОВКИ ДЛЯ ОБЩИХ ВЫРАЖЕНИЙ.

ВСЕ ПОДСТАНОВКИ, ОБСУЖДАВШИЕСЯ РАНЕЕ, ИМЕЮТ ОЧЕНЬ ОГРАНИЧЕННУЮ СФЕРУ ДЕЯТЕЛЬНОСТИ, Т.К. ПРИМЕНИМЫ ЛИШЬ К ПЕРЕМЕННЫМ И ЯВЛЯЮТСЯ ФОРМАМ REDUCE ДОПУСКАЕТ ИСПОЛЬЗОВАНИЕ LET В СЛЕДУЮЩЕЙ СИНАКСИЧЕСКОЙ ФОРМЕ:

<ПОДСТАНОВОЧНОЕ ВЫРАЖЕНИЕ> = <ВЫРАЖЕНИЕ>

ЗДЕСЬ <ПОДСТАНОВОЧНОЕ ВЫРАЖЕНИЕ> - ЭТО ЛЮБОЕ ВЫРАЖЕНИЕ, УДОВЛЕТ

$$; (Y)M^2 + (S)M^2 = (Y)200 + (S)200 = (Y+X)200 \quad ; (Y)M^2 + (S)M^2 = (Y)200 + (S)200 = (Y+X)200$$

$$; X = (X)M^2 + N, \quad X = (Y+X)M^2 + N$$

ВОРЯЮЩЕЕ СЛЕДУЮЩИМ ОГРАНИЧЕНИЯМ:

(1) СИМВОЛЫ ОПЕРАЦИЙ +, -, и / НЕ ДОЛЖНЫ ПОЯВЛЯТЬСЯ НА ВЕРХНЕМ УРОВНЕ В <ПОДСТАНОВОЧНОМ ВЫРАЖЕНИИ>. ОПЕРАЦИЯ "+" МОЖЕТ БЫТЬ ИСПОЛЬЗОВАНА НА ВЕРХНЕМ УРОВНЕ В <ПОДСТАНОВОЧНОМ ВЫРАЖЕНИИ>, НО ТОЛЬКО В БИНАРНОЙ ФОРМЕ И ТОЛЬКО, ЕСЛИ ОТСУТСТВУЮТ СВОБОДНЫЕ ПЕРЕМЕННЫЕ (Т.Е. В ПРАВИЛЕ LET, НО НЕ В ПРАВИЛЕ FOR ALL ... LET). ТАК,

```
LET COS(X)**2 + SIN(X)**2 = 1
```

-РАЗРЕШАЕТСЯ, А

```
FOR ALL X LET COS(X)**2 + SIN(X)**2 = 1
```

-НЕТ.

(2) ВНУТРИ <ПОДСТАНОВОЧНОГО ВЫРАЖЕНИЯ> (Т.Е. НЕ НА ВЕРХНЕМ УРОВНЕ) ОПЕРАЦИИ "+" И "*" МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ ТОЛЬКО В БИНАРНОЙ ФОРМЕ.

ПРИМЕР:

```
LET SIN (X*Y) = SIN(X)*COS(Y) + COS(X)*SIN(Y)
```

-ТАКАЯ ПОДСТАНОВКА ДОПУСТИМА, НО ПОДСТАНОВКА ДЛЯ

```
FN(X*Y + Z)
```

ЗАПРЕЩЕНА.

ОДНАКО, ПРАВИЛА ПОДСТАНОВОК ДЛЯ ВЫРАЖЕНИЙ, СОДЕРЖАЩИХ ТРИ И БОЛЕЕ СЛАГАЕМЫХ И НЕ НУЖНЫ, Т.К. СИСТЕМА БУДЕТ ДЕЛАТЬ ПОДСТАНОВКУ ЛЮБОГО ВЫРАЖЕНИЯ, СОДЕРЖАЩЕГО + ИЛИ * КАК МНОГОКРАТНЫХ ОПЕРАЦИЙ, ИСПОЛЬЗУЯ СООТВЕТСТВУЮЩИЕ РАСШИРЕНИЯ БИНАРНОГО ПРАВИЛА.

(3) ОПЕРАЦИЯ "-" ВНУТРИ <ПОДСТАНОВОЧНОГО ВЫРАЖЕНИЯ> МОЖЕТ БЫТЬ СПЕЦИФИЦИРОВАНА ТОЛЬКО КАК ОДНОМЕСТНАЯ.

ПРИМЕР:

```
LET COS(-X) = COS(X)
```

-РАЗРЕШАЕТСЯ, НО

```
LET COS(X-Y) = <ВЫРАЖЕНИЕ>
```

- ЗАПРЕЩЕНО. СЛЕДУЕТ ЗАМЕТИТЬ, ОДНАКО, ЧТО ПРАВИЛА ДЛЯ COS(X*Y) И ДЛЯ COS(-X) СПОСОБСТВУЮТ РАСКРЫТИЮ ВЫРАЖЕНИЯ ДЛЯ COS(X*Y), КОТОРОЕ РАССМАТРИВАЕТСЯ СИСТЕМОЙ КАК COS(X+(-Y)).

ЛЮБАЯ ПЕРЕМЕННАЯ, ПОЯВЛЯЮЩАЯСЯ В ПОДСТАНОВОЧНОМ ВЫРАЖЕНИИ, МОЖЕТ БЫТЬ ОПИСАНА КАК ПРОИЗВОЛЬНАЯ С ПОМОЩЬЮ ОПЕРАЦИИ FOR ALL.

ПРИМЕР:

```
FOR ALL X,Y LET COS(X*Y) = COS(X)*COS(Y) - SIN(X)*SIN(Y)
FOR ALL X LET LOG(E**X) = X , E**(LOG(X)) = X ,
```


COS(W*T + ТНЕТА(X)) = ТАУ(X);

КАЖДАЯ ПЕРЕМЕННАЯ, УКАЗАННАЯ В **FOR ALL**, ЯВЛЯЕТСЯ ФОРМАЛЬНЫМ ПАРАМЕТРОМ, КОТОРЫЙ ОТЛИЧАЕТСЯ ОТ ДРУГИХ, РАНЕЕ ИЛИ ВПОСЛЕДСТВИИ ВВЕДЕННЫХ ПЕРЕМЕННЫХ, НЕИЗВЕСТНЫХ ВЕЛИЧИН ИЛИ ФОРМАЛЬНЫХ ПАРАМЕТРОВ, ИМЕЮЩИХ ТЕ ЖЕ САМЫЕ ИМЕНА ВНЕ ЭТОГО ПРАВИЛА.

КАК И РАНЬШЕ, ПОСЛЕ ТИПОЛОЖКИ ПОДСТАВКА ОДЕЛИАНА, ВЫЧИСЛЕННОЕ ВЫРАЖЕНИЕ ПРОСМАТРИВАЕТСЯ НА СЛУЧАЙ НОВЫХ РАЗРЕШЕННЫХ ПОДСТАВОВОК, КОТОРЫЕ МОЖНО СДЕЛАТЬ:

ОДАКО, ИНОГДА ЭТО НЕ ЖЕЛАТЕЛЬНО, К ПРИМЕРУ, ЕСЛИ НУЖНО ПРОИНТЕГРИРОВАТЬ ПОЛИНОМ ПО X ПО ПРАВИЛУ ВИДА

FOR ALL N LET XN = X**(N+1)/(N+1);**

ЖЕЛАТЕЛЬНО, ЧТОБЫ ПОДСТАВКА БЫЛА СДЕЛАНА ОДИН РАЗ (ИНАЧЕ X**2 ПЕРЕЙДЕТ В X**3/3, КОТОРОЕ ПЕРЕЙДЕТ В X**4/12 И Т.Д.). ЭТА ПЕРЕПОДСТАВКА МОЖЕТ БЫТЬ ПРЕДОТВРАЩЕНА КОМАНДОЙ

OFF RESUBS

ФЛАГ **RESUBS** ОБЫЧНО УСТАНОВЛЕН.

КОГДА ПОДСТАВОВОЧНОЕ ВЫРАЖЕНИЕ ПОЯВЛЯЕТСЯ В ПРОИЗВЕДЕНИИ, ТО ПОДСТАВКА ДЕЛАЕТСЯ, ЕСЛИ ЭТО ПРОИЗВЕДЕНИЕ ДЕЛИТСЯ НА ПОДСТАВОВОЧНОЕ ВЫРАЖЕНИЯ.

ПРИМЕР:

LET A2*C = 3*Z;**

ЭТО ОЗНАЧАЕТ, ЧТО

A2*C*X**

ЗАМЕНЯЕТСЯ НА

3*Z*X

А

A2*C**2**

ЗАМЕНЯЕТСЯ НА

3*Z*C

ЕСЛИ ПОДСТАВКА ПЛАНИРУЕТСЯ ТОЛЬКО ТОГДА, КОГДА В ПРОИЗВЕДЕНИИ ПОЯВЛЯТСЯ ПОДСТАВОВОЧНЫЕ ВЫРАЖЕНИЯ В СТЕПЕНЯХ, ЯВНО УКАЗАННЫХ В ПРАВИЛЕ, ТО СЛЕДУЕТ ИСПОЛЬЗОВАТЬ КОМАНДУ **МАТЧ**.

ПРИМЕР:

МАТЧ A2*C = 3*Z;**

ПО ЭТОЙ КОМАНДЕ

A2*C*X**

БУДЕТ ЗАМЕНЯТЬСЯ НА

3*Z*X

НО К

A2*C**2**

ПОДСТАВКА ПРИМЕНЕНА НЕ БУДЕТ. **МАТЧ** ТАКЖЕ МОЖЕТ БЫТЬ ИСПОЛЬЗОВАНА В КОНСТРУКЦИИ **FOR ALL**,

ЧТОБЫ УНИЧТОЖИТЬ ПРАВИЛА ЗАДАННЫХ ПОДСТАВОВОК (ОБСУЖДАВШИХСЯ

В ЭТОМ РАЗДЕЛЕ, МОЖНО ВОСПОЛЬЗОВАТЬСЯ КОМАНДОЙ CLEAR, СКОМБИНИРОВАННОЙ, ЕСЛИ НАДО, С FOR ALL.

ПРИМЕР:

```
FOR ALL X CLEAR LOG(E**X), E**LOG(X), COS(W*T+THETA(X));
```

ЗАМЕТИМ, ОДНАКО, ЧТО ИМЕНА ФОРМАЛЬНЫХ ПАРАМЕТРОВ В ЭТОМ СЛУЧАЕ ДОЛЖНЫ БЫТЬ ТЕМИ ЖЕ САМЫМИ ИМЕНАМИ, КОТОРЫЕ ИСПОЛЬЗОВАЛИСЬ ПРИ ОПРЕДЕЛЕНИИ ПОДСТАНОВКИ.

ЗАМЕЧАНИЕ.

ОПЕРАЦИЯ СОПОСТАВЛЕНИЯ С ОБРАЗЦОМ В СИСТЕМЕ БЫЛА УСОВЕРШЕНСТВОВАНА ТАК, ЧТО ЗАМЕЩЕНИЕ В ФОРМЕ

```
FORALL X LET F(A+B*X) = Y;
```

БУДЕТ ПРОИЗВОДИТЬСЯ В ВЫРАЖЕНИИ ТИПА $F(A+B+C+D+E)$. ОДНАКО, ПО ДОСТАТОЧНО СЕРЬЕЗНЫМ ПРИЧИНАМ, В ПОДОБНОМ ВЫРАЖЕНИИ РАЗРЕШАЕТСЯ ТОЛЬКО 5 СЛАГАЕМЫХ.

4.2 ДИФФЕРЕНЦИРОВАНИЕ ВЫРАЖЕНИЯ.

ОПЕРАЦИЯ ДИФФЕРЕНЦИРОВАНИЯ DF ОТНОСИТСЯ К ВСТРОЕННЫМ ОПЕРАЦИЯМ СИСТЕМЫ (СМ. П. 2.4.4).

РАСШИРЕНИЕ СИНТАКСИСА АРГУМЕНТОВ КОМАНДЫ LET ПОЗВОЛЯЕТ ВВОДИТЬ ПРАВИЛА ДИФФЕРЕНЦИРОВАНИЯ ФУНКЦИЙ, ВВЕДЕННЫХ ПРОГРАММИСТОМ ПОСРЕДСТВОМ ОПИСАНИЯ ОПЕРАТОРА. ИХ ОБЩАЯ ФОРМА:

```
FOR ALL <ПЕРЕМЕННАЯ1>, ..., <ПЕРЕМЕННАЯМ> LET DF <ОПЕРАЦИЯ><СПИСОК ПЕРЕМЕННЫХ>, <ПЕРЕМЕННАЯК> = <ВЫРАЖЕНИЕ> ГДЕ
```

```
<СПИСОК ПЕРЕМЕННЫХ> ::= (<ПЕРЕМЕННАЯ1>, ..., <ПЕРЕМЕННАЯМ>)
```

```
<ПЕРЕМЕННАЯ1>, ..., <ПЕРЕМЕННАЯК>, ..., <ПЕРЕМЕННАЯМ>
```

— ЭТО ФОРМАЛЬНЫЕ ПЕРЕМЕННЫЕ — АРГУМЕНТЫ <ОПЕРАЦИИ>.

ДЛЯ ИНФИКСНЫХ ОПЕРАЦИЙ ИСПОЛЬЗУЕТСЯ АНАЛОГИЧНАЯ ФОРМА.

ПРИМЕРЫ:

```
FOR ALL X LET DF(TAN X, X) = SEC(X)**2;
```

```
FOR ALL X, Y LET DF(F(X, Y), X) = 2 * F(X, Y);
```

```
DF(F(X, Y), Y) = X * F(X, Y);
```

ЗАМЕТИМ, ЧТО ВСЕ ФОРМАЛЬНЫЕ ПАРАМЕТРЫ РАССМАТРИВАЕМОЙ ОПЕРАЦИИ ДОЛЖНЫ БЫТЬ ОПИСАНЫ КАК СВОБОДНЫЕ ПЕРЕМЕННЫЕ КОМАНДОЙ FOR ALL; И ЧТО ЭТИ ПРАВИЛА МОГУТ БЫТЬ ПРИМЕНЕНЫ К ОПЕРАЦИЯМ С ЛЮБЫМ ЧИСЛОМ АРГУМЕНТОВ. ЕСЛИ ДЛЯ НЕКОТОРОГО АРГУМЕНТА ОПЕРАЦИИ НИКАКИХ ПРАВИЛ ДИФФЕРЕНЦИРОВАНИЯ НЕ ЗАДАНО, ТО ПРОГРАММА ДИФФЕРЕНЦИРОВАНИЯ ВЕРНЕТ КАК РЕЗУЛЬТАТ ВЫРАЖЕНИЕ В ТЕРМИНАХ DF. К ПРИМЕРУ, ЕСЛИ ПРАВИЛО ДИФФЕРЕНЦИРОВАНИЯ F ПО ВТОРОМУ АРГУМЕН-

ТУ НЕ ЗАДАНО, ТО ВЫЧИСЛЕНИЕ
 $DF(F(X, Z), Z)$
 ОСТАВИТ ЭТО ВЫРАЖЕНИЕ БЕЗ ИЗМЕНЕНИЯ.

4.3 РАБОЧЕЕ ПОЛЕ ВЫРАЖЕНИЯ,

КОГДА ВЫПОЛНЯЕТСЯ ПРИСВАИВАНИЕ АЛГЕБРАИЧЕСКОГО ВЫРАЖЕНИЯ ИЛИ
 ВЫЧИСЛЕНИЕ ВЫРАЖЕНИЯ НА ВЕРХНЕМ УРОВНЕ (Т.Е., НЕ ВНУТРИ СОСТАВ-
 НОГО ОПЕРАТОРА ИЛИ ПРОЦЕДУРЫ), РЕЗУЛЬТАТЫ ВЫЧИСЛЕНИЯ АВТОМАТИ-
 ЧЕСКИ СОХРАНЯЮТСЯ В ОБЛАСТИ, НА КОТОРУЮ МЫ БУДЕМ ССЫЛАТЬСЯ КАК
 НА РАБОЧЕЕ ПОЛЕ. В ДЕЙСТВИТЕЛЬНОСТИ ВЫРАЖЕНИЕ ПРИСВАИВАЕТСЯ ПЕ-
 РЕМЕННОЙ $*ANS$, И ИМЕННО ЭТУ ПЕРЕМЕННУЮ МОЖНО ИСПОЛЬЗОВАТЬ В
 ДАЛЬНЕЙШИХ МАНИПУЛЯЦИЯХ. ДЛЯ ТОГО, ОДНАКО, ЧТОБЫ ВВЕСТИ СВЕД-
 ДОЧКУ КАК ЧАСТЬ ИМЕНИ ПЕРЕМЕННОЙ, НУЖНО ПОСТАВИТЬ ПЕРЕД НЕЕ
 ВОСКЛИЦАТЕЛЬНЫЙ ЗНАК (СМ. П.7.2, СТР.).

ПРИМЕР:

ЕСЛИ МЫ ВЫЧИСЛИЛИ ВЫРАЖЕНИЕ $(X+Y)**2$ НА ВЕРХНЕМ УРОВНЕ
 ПРОГРАММЫ И ЗАТЕМ ХОТИМ ПРОДИФФЕРЕНЦИРОВАТЬ ЕГО ПО Y , ТО
 МОЖНО ВОСПОЛЬЗОВАТЬСЯ КОМАНДОЙ

```
DF(!*ANS, Y);
```

И ПОЛУЧИТЬ ЖЕЛАЕМЫЙ РЕЗУЛЬТАТ.

ЕСЛИ ПРОГРАММИСТ НЕ ЖЕЛАЕТ КАЖДЫЙ РАЗ НАБИРАТЬ $*ANS$,
 ОН МОЖЕТ ВОСПОЛЬЗОВАТЬСЯ КОМАНДОЙ **DEFINE**, ЧТОБЫ ВВЕСТИ ДРУ-
 ГОЕ ИМЯ ДЛЯ РАБОЧЕГО ПОЛЯ. К ПРИМЕРУ,

```
DEFINE WS=!*ANS;
```

БУДЕТ ОЗНАЧАТЬ, ЧТО С ЭТОГО МОМЕНТА **WS** БУДЕТ РАСПОЗНАВАТЬСЯ НА
 ВВОДЕ КАК $*ANS$.

ЕСЛИ ПРОГРАММИСТ ХОЧЕТ ПРИСВОИТЬ ЗНАЧЕНИЕ РАБОЧЕГО ПОЛЯ ПЕ-
 РЕМЕННОЙ ИЛИ ВЫРАЖЕНИЮ ДЛЯ ДАЛЬНЕЙШЕГО ИСПОЛЬЗОВАНИЯ, ТО МОЖНО
 ВОСПОЛЬЗОВАТЬСЯ КОМАНДОЙ **SAVEAS**. ЕЕ СИНТАКСИС:

```
SAVEAS <ВЫРАЖЕНИЕ><ОГРАНИЧИТЕЛЬ>
```

ПРИМЕР:

В ПРЕДЫДУЩЕМ ПРИМЕРЕ, ПОСЛЕ ДИФФЕРЕНЦИРОВАНИЯ, НА РАБОЧЕМ
 ПОЛЕ НАХОДИТСЯ ВЫРАЖЕНИЕ $2*X+2*Y$. ЕСЛИ МЫ ХОТИМ ПРИСВОИ-
 ТЬ ЭТО ЗНАЧЕНИЕ ПЕРЕМЕННОЙ Z , ТО НУЖНО ИСПОЛЬЗОВАТЬ КОМАНДУ

```
SAVEAS Z;
```

ЕСЛИ ПРОГРАММИСТ ХОЧЕТ СОХРАНИТЬ ВЫРАЖЕНИЕ В ФОРМЕ, ПОЗВОЛЯ-
 ЮЩЕЙ ИСПОЛЬЗОВАТЬ НЕКОТОРЫЕ ПЕРЕМЕННЫЕ КАК ФОРМАЛЬНЫЕ ПАРАМЕТ-
 РЫ, ТО МОЖНО ВОСПОЛЬЗОВАТЬСЯ КОМАНДОЙ **FOR ALL**.

ПРИМЕР:

FOR ALL X SAVES N(X);

ТАКАЯ ЗАПИСЬ ДЛЯ ПРЕДЫДУЩЕГО ВЫРАЖЕНИЯ БУДЕТ ОЗНАЧАТЬ, ЧТО
N(Z) РАВНО
 $2 * Y + 2 * Z$

4.4 ОПИСАНИЯ.

К ОПИСАНИЯМ В ЯЗЫКЕ **REDUCE** ОТНОСЯТСЯ ОПИСАНИЯ ТИПА, ОПИСАНИЯ **ARRAY**, **OPERATOR**, **CLEAR**, ОПИСАНИЯ **ON** И **OFF**, ОПИСАНИЯ ВЫВОДА.

ОПИСАНИЯ ТИПА ВВОДЯТ ИМЕНА ПЕРЕМЕННЫХ СООТВЕТСТВУЮЩЕГО ТИПА. ТИПА. ОПИСАНИЕ **ARRAY** ВВОДИТ ИМЕНА МАССИВОВ И ЗАДАЕТ ИХ РАЗМЕРНОСТЬ. ОПИСАНИЕ **OPERATOR** ВВОДИТ ИМЕНА ПРЕФИКСНЫХ ОПЕРАЦИЙ И ФУНКЦИЙ.

ОПИСАНИЯ ТИПА, ОПИСАНИЯ **ARRAY**, **OPERATOR** УЖЕ РАССМАТРИВАЛИСЬ В СООТВЕТСТВУЮЩЕМ КОНТЕКСТЕ. ОСТАЛЬНЫЕ ОПИСАНИЯ РАССМАТРИВАЮТСЯ В ЭТОМ ПАРАГРАФЕ.

1. ОПИСАНИЯ **ON** И **OFF**.

ПРОГРАММИСТУ ПРЕДЛАГАЮТСЯ ДВА ОПИСАНИЯ ДЛЯ УСТАНОВКИ И СБРОСА РАЗЛИЧНЫХ ФЛАГОВ В СИСТЕМЕ. АРГУМЕНТАМИ ОПИСАНИЯ **ON** И **OFF** ЯВЛЯЮТСЯ СПИСКИ ИМЕН ФЛАГОВ, КОТОРЫЕ УСТАНАВЛИВАЮТСЯ ИЛИ СБРАСЫВАЮТСЯ, СООТВЕТСТВЕННО.

ПРИМЕРЫ:

ON FLOAT, GCD;

OFF LIST;

ИСПОЛЬЗОВАНИЕ ЭТИХ И ДРУГИХ ДОСТУПНЫХ ПОЛЬЗОВАТЕЛЮ ФЛАГОВ ПОЯСНЯЕТСЯ В НАСТОЯЩЕМ РУКОВОДСТВЕ ДАЛЕЕ.

2. ОПИСАНИЕ **CLEAR**.

ПРОГРАММИСТ МОЖЕТ УБРАТЬ ПРИСВАИВАНИЯ И ПРАВИЛА ПОДСТАНОВОК ИЗ ЛЮБОГО ВЫРАЖЕНИЯ КОМАНДОЙ **CLEAR**, ЗАПИСАННОЙ В СЛЕДУЮЩЕЙ ФОРМЕ:

CLEAR <ВЫРАЖЕНИЕ>, ..., <ВЫРАЖЕНИЕ><ОГРАНИЧИТЕЛЬ>

НАПРИМЕР:

CLEAR X, N(X, Y);

ЦЕЛЫЙ МАССИВ, ТАКОМ, КАК МАССИВ **A** В П. 2.12.2 (), МОЖЕТ

БЫТЬ ОЧИЩЕН ПО КОМАНДЕ

CLEAR A;

***** ПЕРЕМЕНАЕТ ПОЛОЖЕНИЕ *****

ОТДЕЛЬНЫЙ ЭЛЕМЕНТ ИЗ МАССИВА A МОЖЕТ БЫТЬ ОЧИЩЕН КОМАНДОЙ ТИ-
ПА

CLEAR A(3);

МАССИВ B ОСТАЕТСЯ

ЗАМЕЧАНИЕ.

ОПЫТ РАБОТЫ С СИСТЕМОЙ ПОКАЗЫВАЕТ, ЧТО ЛЮБОЕ ВЫРАЖЕНИЕ,
СТАВШЕЕ НЕБЖНЫМ, СЛЕДУЕТ СРАЗУ ЖЕ УНИЧТОЖАТЬ КОМАНДОЙ CLEAR,
ПРИ ЭТОМ СООТВЕТСТВУЮЩАЯ СВЯЗАННАЯ ПЕРЕМЕННАЯ СТАНОВИТСЯ СВО-
БОДНОЙ И ПАМЯТЬ, ЗАНИМАЕМАЯ ЕЕ ЗНАЧЕНИЕМ, ОСВОБОЖДАЕТСЯ, ТОЧНО
ТАКЖЕ СЛЕДУЕТ СРАЗУ УНИЧТОЖАТЬ СТАВШИЕ НЕБЖНЫМИ LET = ПОДСТА-
НОВКИ, ПОТОМУ ЧТО ОНИ ЗАМЕДЛЯЮТ РАБОТУ СИСТЕМЫ - ВРЕМЯ ТРАТИТСЯ НА
БЕЗУСПЕШНЫЕ ПОИСКИ СООТВЕТСТВУЮЩИХ ИХ ЛЕВЫМ ЧАСТЯМ ПОДВЫРАЖЕ-
НИЙ.

ПРИМЕРЫ.

(1)

A := X**2;

A := X

LET X=C+1;

A) 2
C + 2*C + 1

B := X;
B := C + 1

CLEAR X;

A) 2
X
B) C + 1

(2) (ПРИМЕР ИЗ [10]).

BEGIN
A := X**2;
LET X=C+1; WRITE A; B := X**2;
IF A=B THEN WRITE"OKAY"; END;

2
C + 2*C + 1
OKAY

LET X=A*B;
X := X**2;

***G2 - PUSHDOWN STACK OVERFLOW

*****ERROR TERMINATION*****

4.5 РАБОТА С ФАЙЛАМИ.

ЕСЛИ ПРОГРАММИСТ ЖЕЛАЕТ ЗАГРУЗИТЬ В СИСТЕМУ ПРЕДВАРИТЕЛЬНО ПОДГОТОВЛЕННЫЕ ФАЙЛЫ, СОДЕРЖАЩИЕ ПРОГРАММЫ НА ЯЗЫКЕ **REDUCE**, ИЛИ СОХРАНИТЬ РЕЗУЛЬТАТЫ ВО ВНЕШНЕЙ ПАМЯТИ (ДЛЯ ПОСЛЕДУЮЩЕГО ВВОДА ИЛИ) ВРЕМЕННО, КОГДА НЕ ХВАТАЕТ ОПЕРАТИВНОЙ ПАМЯТИ, ОН МОЖЕТ ВОСПОЛЬЗОВАТЬСЯ КОМАНДАМИ **IN, OUT, SHUT**.

ИМЕНА ВНЕШНИХ ФАЙЛОВ ДОЛЖНЫ УДОВЛЕТВОРЯТЬ НЕ ТОЛЬКО СИНТАКСИСУ ЯЗЫКА **REDUCE**, НО И СОГЛАШЕНИЯМ ОБ ИМЕНАХ ЛОКАЛЬНЫХ ФАЙЛОВ В СООТВЕТСТВУЮЩЕЙ ОПЕРАЦИОННОЙ СИСТЕМЕ. В БОЛЬШИНСТВЕ СИСТЕМ ИМЕНА ФАЙЛОВ - ПРОСТО ИДЕНТИФИКАТОРЫ. НАПОМНИМ, ЧТО ИДЕНТИФИКАТОР **T** ЗАРЕЗЕРВИРОВАН И ОБОЗНАЧАЕТ ТЕРМИНАЛ.

1. КОМАНДА **IN** . ФЛАГ **ESNO** .

АРГУМЕНТОМ ЭТОЙ КОМАНДЫ ЯВЛЯЕТСЯ СПИСОК ИМЕН ФАЙЛОВ. КОМАНДА ДАЕТ УКАЗАНИЕ СИСТЕМЕ ВВЕСТИ КАЖДЫЙ ФАЙЛ (КОТОРЫЙ ДОЛЖЕН СОДЕРЖАТЬ КОМАНДЫ ЯЗЫКА **REDUCE**) В СИСТЕМУ.

ПРИМЕР:

IN F1,GGG

ПО ЭТОЙ КОМАНДЕ БУДУТ ЗАГРУЖЕНЫ ФАЙЛЫ **F1** И **GGG**.

ПРИ ВВОДЕ С ВНЕШНЕГО ФАЙЛА ОПЕРАТОРЫ ПО МЕРЕ СЧИТЫВАНИЯ ПЕЧАТАЮТСЯ НА ЛИСТИНГЕ ИЛИ ЭКРАНЕ ТЕРМИНАЛА. ЕСЛИ ПЕЧАТЬ ВВОДИМОЙ ПРОГРАММЫ НЕ ТРЕБУЕТСЯ, ЕЕ МОЖНО ВЫКЛЮЧИТЬ, СБРОСИВ ФЛАГ **ESNO**.

2. КОМАНДА **END**.

КОМАНДА

END

ДОЛЖНА ЗАКАНЧИВАТЬ ВНЕШНИЙ ФАЙЛ, С КОТОРОГО ПРОИСХОДИТ ВВОД. ОДНА ИЗ ФУНКЦИЙ ЭТОЙ КОМАНДЫ - СБРОСИТЬ ФЛАГ **ESNO**, КОТОРЫЙ МЕШАЕТ ПРОГРАММИСТУ, ВЫВОДЯ ИСХОДНЫЕ КОМАНДЫ НА ТЕРМИНАЛ. ОДНАКО, КОМАНДА ТАКЖЕ ИСПОЛЬЗУЕТСЯ ДЛЯ НЕКОТОРОГО УПРАВЛЕНИЯ БУХГАЛТЕРИЕЙ ФАЙЛА И, СЛЕДОВАТЕЛЬНО, НЕ ДОЛЖНА ОПУСКАТЬСЯ.

ЕСЛИ КОМАНДА **END** ИСПОЛЬЗУЕТСЯ В ОСНОВНОЙ ПРОГРАММЕ, УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ТОГДА НА **LISP**.

3. КОМАНДА OUT

ЭТА КОМАНДА ИМЕЕТ СВОИМ АРГУМЕНТОМ ИМЯ ОДНОГО ФАЙЛА, ОНА НАПРАВЛЯЕТ НА НЕГО ВСЬ ПОСЛЕДУЮЩИЙ ВЫВОД ДО ТЕХ ПОР, ПОКА ЭТОТ ФАЙЛ НЕ БУДЕТ ЗАКРЫТ (ПОСРЕДСТВОМ КОМАНДЫ SHUT) ИЛИ ПОЖАЛЕНЕНО ПОЯВИТСЯ КОМАНДА OUT С ДРУГИМ АРГУМЕНТОМ. ЕСЛИ ФАЙЛ БЫЛ УЖЕ ИСПОЛЬЗОВАН ДЛЯ ЗАПИСИ ВЫВОДА В ПРОЦЕССЕ ВЫПОЛНЕНИЯ ТЕКУЩЕЙ ПРОГРАММЫ, ТО ВНОВЬ ВЫВОДИМЫЕ С ПОМОЩЬЮ КОМАНДЫ OUT РЕЗУЛЬТАТЫ ПРИСОЕДИНЯЮТСЯ К КОНЦУ ЭТОГО ФАЙЛА (ЕСЛИ, КОНЕЧНО, ФАЙЛ УЖЕ НЕ БЫЛ ЗАКРЫТ С ПОМОЩЬЮ КОМАНДЫ SHUT). ЕСЛИ В КАЧЕСТВЕ АРГУМЕНТА КОМАНДЫ OUT ИСПОЛЬЗУЕТСЯ УЖЕ СУЩЕСТВУЮЩИЙ ФАЙЛ, ТО ОН СТИРАЕТСЯ ПЕРЕД ПЕРВЫМ ИСПОЛЬЗОВАНИЕМ ДЛЯ ЗАПИСИ ВЫВОДА В ПРОЦЕССЕ ВЫПОЛНЕНИЯ ПРОГРАММЫ.

ЧТОБЫ ВЫВЕСТИ РЕЗУЛЬТАТЫ НА ТЕРМИНАЛ, НЕ ЗАКРЫВАЯ ВЫХОДНОГО ФАЙЛА, МОЖНО ИСПОЛЬЗОВАТЬ ФАЙЛ С ЗАРЕЗЕРВИРОВАННЫМ ИМЕНЕМ T

ПРИМЕР:

OUT OFILE;

ЭТА КОМАНДА НАПРАВЛЯЕТ ВЫВОД НА ФАЙЛ OFILE, А КОМАНДА

OUT T;

НАПРАВЛЯЕТ ВЫВОД НА ТЕРМИНАЛ.

4. КОМАНДА SHUT.

КОМАНДА ИСПОЛЬЗУЕТСЯ В СЛЕДУЮЩЕЙ ФОРМЕ:

SHUT <СПИСОК ИМЕН ФАЙЛОВ>;

<СПИСОК ИМЕН ФАЙЛОВ> - СПИСОК ИМЕН ФАЙЛОВ, КОТОРЫЕ ДОЛЖНЫ БЫТЬ ЗАКРЫТЫ;

ЭТА КОМАНДА ИСПОЛЬЗУЕТСЯ ДЛЯ ТОГО, ЧТОБЫ ЗАКРЫТЬ ВЫХОДНОЙ ФАЙЛ ПОСЛЕ ОКОНЧАНИЯ ВЫВОДА НА НЕГО. В БОЛЬШИНСТВЕ СИСТЕМ ЭТО ДЕЙСТВИЕ ПРОГРАММИСТА ЯВЛЯЕТСЯ ОБЯЗАТЕЛЬНЫМ, ИНАЧЕ ВЫВОДИМЫЙ РЕЗУЛЬТАТ МОЖЕТ БЫТЬ УТЕРЯН.

ЕСЛИ КОМАНДА OUT ПРИМЕНЯЕТСЯ К ФАЙЛУ, КОТОРЫЙ УЖЕ ЗАКРЫТ, ТО СОДЕРЖИМОЕ ЕГО СТИРАЕТСЯ ПРЕЖДЕ, ЧЕМ БУДЕТ ЗАПИСАН НОВЫЙ ВЫВОД.

ЕСЛИ ВЫВОДИМЫЕ НА ВНЕШНИЙ ФАЙЛ ВЫРАЖЕНИЯ В ДАЛЬНЕЙШЕМ ПРЕДПОЛАГАЕТСЯ ИСПОЛЬЗОВАТЬ КАК ВВОДНУЮ ИНФОРМАЦИЮ, ПЕРЕД ВЫВОДОМ ЭТИХ ВЫРАЖЕНИЙ СЛЕДУЕТ СБРОСИТЬ ФЛАГ INAT, (ФАКТИВ ТЕМ САМЫМ "ЕСТЕСТВЕННЫЙ" МЕТОД ПЕЧАТИ ВЫРАЖЕНИЙ (С "ПОДНЯТЫМИ" ПОКАЗАТЕЛЯМИ СТЕПЕНЕЙ), КОТОРЫЙ НЕСОВМЕСТИМ С СИНТАКСИСОМ ВВОДА. ЕСЛИ ФЛАГ ПРИ ВЫВОДЕ СБРОШЕН, ТО ПРИ ВВОДЕ ЭТИ ВЫРАЖЕНИЯ БУДУТ ПЕЧАТАТЬСЯ С ОГРАНИТЕЛЕМ.

ПРИМЕР.

```

OFF NAT)
OUT OUT)
X 1= (Y+Z)**2)
WRITE "END")
SHUT OUT)
ON NAT)

```

ЭТА ПРОГРАММА ОРГАНИЗУЕТ ФАЙЛ OUT, КОТОРЫЙ СОДЕРЖИТ СЛЕДУЮЩУЮ ИНФОРМАЦИЮ:

```

X 1= Y**2 + 2*Y*Z + Z**2)
END)

```

ЗАМЕЧАНИЯ.

(1) АЛГЕБРАИЧЕСКОЕ ВЫРАЖЕНИЕ, ПРЕЖДЕ, ЧЕМ ОНО ВВОДИТСЯ НА ВНЕШНИЙ ФАЙЛ, УПРОЩАЕТСЯ СИСТЕМОЙ.

(2) ПРОИЗВОДИТЬ ВВВОД ОДНОВРЕМЕННО НА 2 ИЛИ БОЛЕЕ ВНЕШНИХ ФАЙЛА НЕЛЬЗЯ.

(3) КОНСТРУКЦИИ ЯЗЫКА, НЕ ИМЕЮЩИЕ ЗНАЧЕНИЯ (ТАКИЕ, КАК ОПИСАНИЯ, ПРАВИЛА LET, ОПИСАНИЯ ПРОЦЕДУР), НЕ БУДУТ ЗАПИСАНЫ НА ВНЕШНЕМ ФАЙЛЕ. ДЛЯ ТОГО, ЧТОБЫ ОНИ ЗАПИСАЛИСЬ, МОЖНО ВОСПОЛЬЗОВАТЬСЯ КОМАНДОЙ WRITE, (СМ. П. 3.2.5) В ТАКОМ, НАПРИМЕР, ВИДЕ:

```

WRITE"
ALGEBRAIC PROCEDURE...
..."

```

4.6 КОНТРОЛЬ ЗА ВЫЧИСЛИТЕЛЬНЫМ ПРОЦЕССОМ.

КРОМЕ РАЗЛИЧНЫХ ОПИСАНИЙ И ФЛАГОВ ВЫВОДА ПРОГРАММИСТ МОЖЕТ ИСПОЛЬЗОВАТЬ ДВА ДОПОЛНИТЕЛЬНЫХ ФЛАГА, КОТОРЫЕ УПРАВЛЯЮТ ВНУТРЕННИМ ПРЕОБРАЗОВАНИЕМ ВЫРАЖЕНИЯ. ФЛАГ EXP (EXP ANSTON) УПРАВЛЯЕТ РАСКРЫТИЕМ СКОБОК В ВЫРАЖЕНИЯХ. ЕСЛИ ОН СБРОШЕН, ТО РАСКРЫТИЕ СТЕПЕНЕЙ МНОГОЧЛЕНОВ И ИХ ПРОИЗВЕДЕНИЙ НЕ ПРОИЗВОДИТСЯ.

КОГДА СУММИРУЮТСЯ ДВЕ РАЦИОНАЛЬНЫЕ ФУНКЦИИ, ТО REDUCE ОБЫЧНО ПОЛУЧАЕТ РЕЗУЛЬТАТ С ОБЩИМ ЗНАМЕНАТЕЛЕМ. ОДНАКО, ЕСЛИ ПРОГРАММИСТ НЕ ХОЧЕТ, ЧТОБЫ ЗНАМЕНАТЕЛИ КОМБИНИРОВАЛИСЬ, ОН МОЖЕТ СБРОСИТЬ ФЛАГ MSD (MAKING COMMON DENOMINATORS), КОТОРЫЙ УПРАВЛЯЕТ ЭТИМ ПРОЦЕССОМ. ОСОБЕННО ЭТО БЫВАЕТ ПОЛЕЗНО ТОГДА, КОГДА НИКАКИХ ВЫЧИСЛЕНИЯ НАИБОЛЬШЕГО ОБЩЕГО ДЕЛИТЕЛЯ НЕ ПЛАНИРУЕТСЯ, ИЛИ В ДАЛЬНЕЙШЕМ ТРЕБУЕТСЯ ДИФФЕРЕНЦИРОВАНИЕ РАЦИОНАЛЬНОЙ ФУНКЦИИ. EXP И MSD ОБЫЧНО УСТАНОВЛЕНЫ.

4.7 НАХОЖДЕНИЕ ОБЩЕГО ДЕЛИТЕЛЯ ДВУХ ПОЛИНОМОВ.

В ЯЗЫКЕ REDUCE СУЩЕСТВУЮТ СПОСОБЫ (ПО ВЫБОРУ ПРОГРАММИСТА) ДЛЯ СОКРАЩЕНИЯ ОБЩИХ МНОЖИТЕЛЕЙ В ЧИСЛИТЕЛЕ И ЗНАМЕНАТЕЛЕ ВЫРАЖЕНИЯ. СИСТЕМА ВЫПОЛНИТ ПРОВЕРКУ НА НАИБОЛЬШИЙ ОБЩИЙ ДЕЛИТЕЛЬ.

ЕСЛИ УСТАНОВЛЕН ФЛАГ GCD,

ОДНАКО, ЕСЛИ ОБЩИМИ МНОЖИТЕЛЯМИ В ЧИСЛИТЕЛЕ И ЗНАМЕНАТЕЛЕ ЯВЛЯЮТСЯ ПЕРЕМЕННЫЕ ИЛИ ЧИСЛОВЫЕ ВЫРАЖЕНИЯ, ПРОВЕРКА И СОКРАЩЕНИЯ ВЫПОЛНЯЮТСЯ АВТОМАТИЧЕСКИ.

ФЛАГ GCD ОБЫЧНО СБРОШЕН, ПО СЛЕДУЮЩИМ ПРИЧИНАМ:

- (1) ОН РАСХОДУЕТ БОЛЬШОЕ КОЛИЧЕСТВО МАШИННОГО ВРЕМЕНИ;
- (2) ФЛАГИ EXP, MSD, КОТОРЫЕ ОБЫЧНО УСТАНОВЛЕНЫ, ГАРАНТИРУЮТ, ЧТО ЛЮБОЕ РАЦИОНАЛЬНОЕ ВЫРАЖЕНИЕ, ЭКВИВАЛЕНТНОЕ 0, УПРОЩЕНИЕМ БУДЕТ СВЕДЕНО К 0;
- (3) ЕСЛИ ЗНАМЕНАТЕЛЬ ЯВЛЯЕТСЯ НАИБОЛЬШИМ ОБЩИМ ДЕЛИТЕЛЕМ, КАК В ВЫРАЖЕНИИ $(X^2 - 2X + 1)/(X - 1)$, СИСТЕМА ВЫПОЛНИТ СОКРАЩЕНИЕ, ДАЖЕ ЕСЛИ СБРОШЕН ФЛАГ GCD.

4.8 ВЫВОД ВЫРАЖЕНИЯ.

ПРИ ПЕЧАТАНИИ ВЫРАЖЕНИЙ, ПОЛУЧЕННЫХ ВО ВРЕМЯ ВЫЧИСЛЕНИЯ, СИСТЕМА REDUCE ПРОЯВЛЯЕТ ЗАМЕЧАТЕЛЬНУЮ ГИБКОСТЬ. КАК УПОМИНАЛОСЬ РАНЬШЕ, НИКАКОГО ЯВНОГО ОПЕРАТОРА ФОРМАТА НЕ ПРИМЕНЯЕТСЯ, Т.К. ЭТО ПРИНЕСЕТ МАЛО ПОЛЬЗЫ ПРИ АЛГЕБРАИЧЕСКИХ ВЫЧИСЛЕНИЯХ, ГДЕ РАЗМЕРЫ ВЫВОДИМЫХ РЕЗУЛЬТАТОВ ИЛИ ИХ КОМПОЗИЦИЯ, В ОБЩЕМ, НЕИЗВЕСТНЫ ЗАРАНЕЕ. ВМЕСТО ЭТОГО, REDUCE ПРЕДЛАГАЕТ НА ВЫБОР НЕКОЛЬКО СПОСОБОВ, ПОЗВОЛЯЮЩИХ ПРОУВЕСТИ ВЫВОД В ПОНЯТНОЙ И ДОПУСТИМО ПРЯТНОЙ ФОРМЕ.

КАК УЖЕ УПОМИНАЛОСЬ, АЛГЕБРАИЧЕСКОЕ ВЫРАЖЕНИЕ ПЕЧАТАЕТСЯ В РАЗВЕРНУТОМ ВИДЕ, ЗАПОЛНЯЯ ВСЮ СТРОКУ НА ВЫВОДЕ СВОИМИ ЧЛЕНАМИ.

ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ВЫВОДА МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ ОПИСАНИЯ ВЫВОДА. СЛЕДУЕТ, ОДНАКО, ЗАМЕТИТЬ, ЧТО ТРАНСФОРМАЦИЯ БОЛЬШОГО ВЫРАЖЕНИЯ, ПРОВОДИМАЯ С ЦЕЛЬЮ ОБЕСПЕЧЕНИЯ РАЗНООБРАЗИЯ ФОРМАТОВ ВЫВОДА, МОЖЕТ ПОТРЕБОВАТЬ МНОГО МАШИННОГО ВРЕМЕНИ И ПАМЯТИ. В ПОДОБНЫХ СЛУЧАЯХ, ЕСЛИ ПРОГРАММИСТ ХОЧЕТ УСКОРИТЬ ПЕЧАТЬ ВЫВОДИМЫХ РЕЗУЛЬТАТОВ, ОН МОЖЕТ СБРОСИТЬ ФЛАГ PRI, ИСПОЛЬЗУЯ КОМАНДУ OFF. ЕСЛИ ФЛАГ СБРОШЕН, ТО НЕ ИСПОЛЬЗУЕТСЯ НИ ОДИН ИЗ ПЕРЕЧИСЛЕННЫХ ДАЛЕЕ СПОСОБОВ ВЫВОДА - ВЫВОД ПРОИСХОДИТ В ОДНОМ ФИКСИРОВАННОМ ФОРМАТЕ, КОТОРЫЙ, В ОСНОВНОМ, ОТРАЖАЕТ ВНУТРЕННЮЮ ФОРМУ ВЫРАЖЕНИЯ. ФЛАГ PRI ОБЫЧНО УСТАНОВЛЕН,

ПРИ УСТАНОВЛЕННОМ ФЛАГЕ PRI СУЩЕСТВУЮТ СЛЕДУЮЩИЕ СПОСОБЫ ОПИСАНИЯ ВЫВОДА:

1. ОПИСАНИЕ ORDER.

ЭТО ОПИСАНИЕ МОЖЕТ БЫТЬ ИСПОЛЬЗОВАНО ДЛЯ УПОРЯДОЧЕНИЯ ПЕРЕМЕННЫХ НА ВЫВОДЕ.

ТАК, ПРИ ИСПОЛЬЗОВАНИИ

ORDER X, Y, Z

X СТАВИТСЯ ПЕРЕД Y, Y ПЕРЕД Z. X, Y, Z ПРЕДШЕСТВУЮТ ДРУГИМ ПЕРЕМЕННЫМ В СООТВЕТСТВУЮЩЕМ ВЫРАЖЕНИИ И ВСЕМ ПЕРЕМЕННЫМ, ПОЯВЛЯЮЩИМСЯ В ДАЛЬНЕЙШИХ ОПИСАНИЯХ ORDER, ЕСЛИ НЕ

УПОРЯДОЧЕННЫМ ТАКИМ СПОСОБОМ ПЕРЕМЕННЫМ, ПОРЯДОК ПЕРЕМЕННЫХ МОЖЕТ БЫТЬ ИЗМЕНЕН ДАЛЬНЕЙШИМ ВЫЗОВОМ **ORDER**, НО ТОГДА ПЕРЕУПОРЯДОЧЕННЫЕ ПЕРЕМЕННЫЕ БУДУТ ИМЕТЬ ПОРЯДОК НИЖЕ, ЧЕМ В ПРЕДЫДУЩИХ ВЫЗОВАХ **ORDER**.

ТАК, КОМАНДА

ORDER X,Y,Z

ORDER Y,X

ПОСТАВИТ Z ПЕРЕД Y И X

2. ОПИСАНИЕ FACTOR.

АРГУМЕНТОМ ЭТОГО ОПИСАНИЯ ЯВЛЯЕТСЯ СПИСОК ПЕРЕМЕННЫХ И/ЛИ ЯДЕРНЫХ ФОРМ.

ПРИ ВЫВОДЕ ВЫРАЖЕНИЯ НА ПЕЧАТЬ ЧЛЕНЫ, СОДЕРЖАЩИЕ ОДНУ И ТУ ЖЕ СТЕПЕНЬ ПЕРЕМЕННОЙ ИЛИ ЯДЕРНОЙ ФОРМЫ ИЗ СПИСКА АРГУМЕНТОВ ОПИСАНИЯ **FACTOR** ГРУППИРУЮТСЯ, А ОБЩИЙ МНОЖИТЕЛЬ ВЫНОСИТСЯ ЗА СКОБКУ.

ВСЕ ВЫРАЖЕНИЯ, ВКЛЮЧАЮЩИЕ ПРЕФИКСНУЮ ОПЕРАЦИЮ, ТАКЖЕ МОГУТ БЫТЬ ФАКТОРИЗОВАНЫ ВКЛЮЧЕНИЕМ ИМЕНИ ОПЕРАЦИИ В СПИСОК АРГУМЕНТОВ ОПИСАНИЯ.

ПРИМЕР:

FACTOR X,COS,SIN(X)

-ВСЕ СТЕПЕНИ X И SIN(X) И ВСЕ ФУНКЦИИ КОСИНУСА БУДУТ ФАКТОРИЗОВАНЫ.

ОПИСАНИЕ

REMFAC V1,...,VN

ЗАПРЕТИТ ФАКТОРИЗАЦИЮ ДЛЯ ВСЕХ V1,...,VN.

ЗАМЕЧАНИЯ.

(1) ЕСЛИ КАКОЯ-ТО ЧЛЕН ВЫРАЖЕНИЯ СОДЕРЖИТ ОДНОВРЕМЕННО ДВЕ ИЛИ БОЛЕЕ ПЕРЕМЕННЫХ И/ЛИ ЯДЕРНЫХ ФОРМ ИЗ СПИСКА АРГУМЕНТОВ ОПИСАНИЯ **FACTOR**, ТО ЭТОТ ЧЛЕН В ПРОЦЕССЕ ФАКТОРИЗАЦИИ НЕ УЧАСТВУЕТ.

ПРИМЕР 1

ЭТОТ ПРИМЕР ИЛЛЮСТРИРУЕТ НЕ ТОЛЬКО ДАННОЕ ЗАМЕЧАНИЕ, НО И ДЕЙСТВИЕ ФЛАГА **ALLFAC** (П. 4.8.4).

$A = F + 2 * (G + 2 * G) + F * (G + 2 * H) / (2 * B)$

$A = (F * (2 * B + G) + F * (4 * B * G + F + H + G)) / (2 * B)$

OFF ALLFAC

A)

2 2

2

2

```

(2*B*G *F + 4*B*G*F + H*F + G *F)/(2*B)
OFF ALLFAC;
FACTOR F)
A)
      2      2      2
(F *(2*B*G + 4*B*G) + F*(H + G ))/(2*B)
FACTOR G)
A)
      2      2      2
(2*F *G *B + 4*F *G*B + F*G + F*H)/(2*B)

```

(2) СРАВНИВАЯ РЕЗУЛЬТАТЫ ДЕЙСТВИЯ ОПИСАНИЯ **FACTOR** С РЕЗУЛЬТАТАМИ ДЕЙСТВИЯ ФЛАГА **ALLFAC** ОТМЕТИМ, ЧТО В ПОСЛЕДНЕМ СЛУЧАЕ "ОДНОЧЛЕННАЯ" ФАКТОРИЗАЦИЯ ОПРЕДЕЛЯЕТСЯ ПОСЛЕДОВАТЕЛЬНЫМ РАСПОЛОЖЕНИЕМ ЧЛЕНОВ В ВЫРАЖЕНИИ, А ОПИСАНИЕ **FACTOR** ОСУЩЕСТВЛЯЕТ БОЛЕЕ ИЗБИРАТЕЛЬНУЮ "ОДНОЧЛЕННУЮ" ФАКТОРИЗАЦИЮ В РАЗЛИЧНЫХ ВИДАХ.

3. ОПИСАНИЯ UP И DOWN,

АРГУМЕНТОМ КАЖДОГО ИЗ ЭТИХ ОПИСАНИЙ ЯВЛЯЕТСЯ СПИСОК ПЕРЕМЕННЫХ И/ИЛИ ЯДЕРНЫХ ФОРМ.

ПЕРЕМЕННАЯ (ЯДЕРНАЯ ФОРМА) ИЗ СПИСКА АРГУМЕНТОВ ОПИСАНИЯ UP ВСЕГДА БУДЕТ ПОМЕЩАТЬСЯ В ЧИСЛИТЕЛЬ (В ОТРИЦАТЕЛЬНОЙ СТЕПЕНИ, ЕСЛИ ЭТО ПОТРЕБУЕТСЯ). АНАЛОГИЧНО, С ПОМОЩЬЮ ОПИСАНИЯ DOWN ПЕРЕМЕННУЮ (ИЛИ ЯДЕРНУЮ ФОРМУ) МОЖНО ПОМЕСТИТЬ В ЗНАМЕНАТЕЛЬ.

ПРИМЕР:

```

X := A/B)
X := A/B
UP B; X;
(-1)
A*B
DOWN A; X)
(-1) (-1)
B /A

```

4. ФЛАГИ ВЫВОДА.

ФОРМА ВЫВОДА МОЖЕТ БЫТЬ МОДИФИЦИРОВАНА НЕ ТОЛЬКО ПОСРЕДСТВОМ ОПИСАНИЯ **ORDER**, **FACTOR**, **UP** И **DOWN**, НО И ПЕРЕКЛЮЧЕНИЕМ С ПОМОЩЬЮ ОПИСАНИЯ **ON** И **OFF** РАЗЛИЧНЫХ ФЛАГОВ УПРАВЛЕНИЯ ВЫВОДОМ. МЫ ПРОИЛЛЮСТРИРУЕМ ИСПОЛЬЗОВАНИЕ ЭТИХ ФЛАГОВ НА ПРИМЕРЕ, А ИМЕННО : ПЕЧАТАЯ ВЫРАЖЕНИЕ

```

S := X**2*(Y**2+2*Y)+X*(Y**2+Z)/(2*A)

```

СООТВЕТСТВУЮЩИЕ ФЛАГИ:

A*Y**2+Y**2
A*Y**2+Y**2

(1) ALLFAC.

ЭТОТ ФЛАГ УКАЗЫВАЕТ СИСТЕМЕ, ЧТО НАДО ПРОСМОТРЕТЬ ВСЕ ВЫРАЖЕНИЕ, ИЛИ КАЖДОЕ ПОДВЫРАЖЕНИЕ, ЗАКЛЮЧЕННОЕ В СКОБКИ В ПОИСКАХ ПРОСТОГО МНОЖИТЕЛЯ И НАПЕЧАТАТЬ ЕГО ЗА СКОБКАМИ.

ЕСЛИ ФЛАГ УСТАНОВЛЕН, ТО НА ВЫВОДЕ ПОЛУЧАЕТСЯ КОМПАКТНОЕ ВЫРАЖЕНИЕ С ВЫЯВЛЕННОЙ СТРУКТУРОЙ. ЧАСТО, ОДНАКО, ТАКИЕ ВЫРАЖЕНИЯ ТРУДНО ПОДАЮТСЯ ИНТЕРПРЕТАЦИИ И ПОЧЛЕННОМУ СРАВНЕНИЮ, Т.К. СКОБКИ В НИХ РАСПОЛАГАЮТСЯ В НЕСКОЛЬКО УРОВНЕЙ. ЧТОБЫ ЗАПРЕТИТЬ ТАКУЮ "ОДНОЧЛЕННУЮ" ФАКТОРИЗАЦИЮ, ФЛАГ НУЖНО СБРОСИТЬ. ТАКИМ ОБРАЗОМ, НАШЕ ВЫРАЖЕНИЕ СО СБРОШЕННЫМ ФЛАГОМ ALLFAC БУДЕТ ПЕЧАТАТЬСЯ В ВИДЕ:

OFF ALLFAC)

```
SI
      2 2      2      2
X*(2*X*Y + A + 4*X*Y + A + X*Y + X*Z)/(2*A)
```

A С УСТАНОВЛЕННЫМ ФЛАГОМ ALLFAC

ON ALLFAC)

```
SI
      2      2
X*(2*X*Y + A + 4*X*Y + A + Y + Z)/(2*A)
```

ALLFAC ОБЫЧНО УСТАНОВЛЕН, УСТАНОВЛЕН ОН И В ПОСЛЕДУЮЩИХ ПРИМЕРАХ, ЕСЛИ НЕ ОГОВОРЕНО ПРОТИВНОЕ.

(2) DIV.

ЭТОТ ФЛАГ ПОЗВОЛЯЕТ РАСПРЕДЕЛЯТЬ ПРОСТЫЕ МНОЖИТЕЛИ, ВХОДЯЩИЕ В СОСТАВ ЗНАМЕНАТЕЛЯ, - ЧИСЛА И ОДНОЧЛЕННЫ - ПОД КАЖДЫМ ТЕРМОМ ЧИСЛИТЕЛЯ, ТАК, ЧТО НА ВЫВОДЕ ПОЛУЧАЮТСЯ ЧИСЛЕННО-РАЦИОНАЛЬНЫЕ КОЭФФИЦИЕНТЫ И ОТРИЦАТЕЛЬНЫЕ СТЕПЕНИ ОДНОЧЛЕНОВ. (МНЕМОНИЧЕСКОЕ ПРАВИЛО: DIV DIVIDES BY MONOMIALS). С УСТАНОВЛЕННЫМ ФЛАГОМ DIV НАШЕ ВЫРАЖЕНИЕ ПЕЧАТАЕТСЯ В ВИДЕ:

ON ALLFAC) ON DIV)

```
SI
      2      2      (-1)      (-1)
X*(X*Y + 2*X*Y + 1/2*Y + A + 1/2*A + Z)
OFF DIV)
```

ФЛАГ DIV ОБЫЧНО СБРОШЕН.

(3) LIST.

ЭТОТ ФЛАГ ЗАСТАВЛЯЕТ СИСТЕМУ ПЕЧАТАТЬ КАЖДЫЙ ЧЛЕН СУММЫ НА НОВОЙ СТРОКЕ. С УСТАНОВЛЕННЫМ ФЛАГОМ LIST НАШЕ ВЫРАЖЕНИЕ НАПЕЧАТАЕТСЯ СЛЕДУЮЩИМ ОБРАЗОМ:

ON ALLFAC) ON LIST)

```
SI
      2
X*(2*X*Y + A
+ 4*X*Y + A
```

```

      2
    + Y
    + Z)/(2*A)
OFF LIST)

```

ФЛАГ LIST ОБЫЧНО СБРОШЕН.

(4) RAT.

ЭТОТ ФЛАГ ПОЛЕЗЕН ТОЛЬКО ПРИ ВЫВОДЕ ВЫРАЖЕНИЙ, В КОТОРЫХ ПЕРЕМЕННЫЕ ФАКТОРИЗОВАНЫ С ПОМОЩЬЮ ФАКТОР. В ЭТОМ РЕЖИМЕ ОБЫЧНО ЗАМЕНАТЕЛЬ ВЫРАЖЕНИЯ ПЕЧАТАЕТСЯ С КАЖДЫМ ФАКТОРИЗОВАННЫМ ПОДВЫРАЖЕНИЕМ. РЕЗУЛЬТАТ МОЖНО РАССМАТРИВАТЬ КАК ПОЛИНОМ ИЛИ КАК СТЕПЕННОЙ РЯД ПО ФАКТОРИЗОВАННОЙ ПЕРЕМЕННОЙ (ЯДЕРНОЙ ФОРМЕ) С КОЭФФИЦИЕНТАМИ, КОТОРЫЕ ЯВЛЯЮТСЯ РАЦИОНАЛЬНЫМИ ФУНКЦИЯМИ КАКИХ-ТО ДРУГИХ ПЕРЕМЕННЫХ (ЯДЕРНЫХ ФОРМ). (СНЕМОНИКА: RAT FOR RATIONAL-FUNCTION COEFFICIENTS).

ПРИМЕР:

```

FACTOR X)
ON ALLFAC; OFF RAT;
S)

```

```

      2      2
(2*X *Y*A*(Y + 2) + X*(Y + Z))/(2*A)

```

УСТАНАВЛИВАЕМ ФЛАГ RAT:

```

ON RAT;
S)
      2      2
X *Y*(Y + 2) + X*(Y + Z))/(2*A)

```

```
OFF RAT;
```

ОБЫЧНО ФЛАГ RAT СБРОШЕН.

ДАЛЕЕ, ЕСЛИ МЫ ОСТАВЛЯЕМ X ФАКТОРИЗОВАННЫМ И УСТАНАВЛИВАЕМ ОДНОВРЕМЕННО DIV И RAT, ТО ПОЛУЧАЕМ:

```

ON DIV; ON RAT;
S)
      2      2      (1)      2
X *Y*(Y + 2) + 1/2*X*A *(Y + Z)
OFF DIV;

```

НАКОНЕЦ, ПРИ ФАКТОРИЗОВАННОМ X, УСТААНОВЛЕННОМ RAT И СБРОШЕНОМ ALLFAC, ПОЛУЧИМ:

```

OFF ALLFAC; ON RAT;
S)
      2      2      2
X *(Y + 2*Y) + X*(Y + Z)/(2*A)
REMFAC X)
ON ALLFAC; OFF RAT;

```

ЗАМЕЧАНИЯ.

(1) ОТМЕТИМ, ЧТО РЯДЫ И ПОЛИНОМЫ ЧАСТО ПРИНИМАЮТ БОЛЕЕ ПРИВЛЕКАТЕЛЬНЫЙ ВИД, ЕСЛИ УСТААНОВЛЕННЫ ОДНОВРЕМЕННО И DIV И RAT

(2) ЕСЛИ УСТАНОВЛЕННЫ ОДНОВРЕМЕННО ФЛАГИ **FACTOR**, **ALLFAC**, ТО ФЛАГ **ALLFAC** БУДЕТ ОКАЗЫВАТЬ ВЛИЯНИЕ НА КОЭФФИЦИЕНТЫ ОДНОЧЛЕНОВ В ФАКТОРИЗОВАННОМ ВЫРАЖЕНИИ.

ПРИМЕР 1

A I = F2*(G**2 + 2*G) + F*(G**2 + H)/(2*B)**

A I = (F*(2*B*G2 + F + 4*B*G*F + H + G))/(2*B)**

OFF ALLFAC)

FACTOR G)

A I = (G2*(2*B*F + F) + 4*G*B*F + H*F)/(2*B)**

ON ALLFAC)

A I = (GF*(2*B*F + 1) + 4*G*B*F + H*F)/(2*B)**

4.9 ЧИСЛЕННЫЕ РАСЧЕТЫ.

В ЯЗЫКЕ **REDUCE**, ЕСТЕСТВЕННО, МОЖНО ПРОИЗВОДИТЬ И ЧИСЛЕННЫЕ РАСЧЕТЫ, ЗАДАВАЯ ПЕРЕМЕННЫЕ И ПОДВЫРАЖЕНИЯ ЧИСЛЕННО.

КАК УЖЕ УПОМИНАЛОСЬ (П. 3.2.5), ПРИ УПРОШЕНИИ ВЫРАЖЕНИЯ ДЕЙСТВИТЕЛЬНОЕ ЧИСЛО ПРЕДСТАВЛЯЕТСЯ СИСТЕМОЙ В ВИДЕ ОТНОШЕНИЯ ДВУХ ЦЕЛЫХ. НО ЕСЛИ ЭТИ ЦЕЛЫЕ ЧИСЛА ВЕЛИКИ ПО АБСОЛЮТНОЙ ВЕЛИЧИНЕ, ТО ПРИМЕНЯЕМАЯ В СИСТЕМЕ ТОЧНАЯ РАЦИОНАЛЬНАЯ АРИФМЕТИКА МОЖЕТ ПОТРЕБОВАТЬ ЗНАЧИТЕЛЬНОГО КОЛИЧЕСТВА МАШИННОГО ВРЕМЕНИ, А РЕЗУЛЬТАТ ВЫЧИСЛЕНИЯ ТРУДНО БУДЕТ ИНТЕРПРЕТИРОВАТЬ КАЧЕСТВЕННО. В ТАКИХ СЛУЧАЯХ, ЕСЛИ ПРИЕМЛЕМЫ ЧИСЛЕННЫЕ КОЭФФИЦИЕНТЫ НЕВЫСОКОГО ТОЧНОСТИ, ПРОГРАММИСТ МОЖЕТ ПЕРЕЙТИ К АРИФМЕТИКЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ТОЧКОЙ. ДЛЯ ЭТОГО НУЖНО УСТАНОВИТЬ ФЛАГ **!FLOAT**. КОГДА ЭТОТ ФЛАГ УСТАНОВЛЕН, ЛЮБОЕ НЕЦЕЛОЕ РАЦИОНАЛЬНОЕ ЧИСЛО АППРОКСИМИРУЕТСЯ ЧИСЛОМ С ПЛАВАЮЩЕЙ ТОЧКОЙ, И РЕЗУЛЬТАТОМ ЛЮБОЙ АРИФМЕТИЧЕСКОЙ ОПЕРАЦИИ (ЕСЛИ ЛЮБОЙ ИЗ ЕЕ ОПЕРАНДОВ - ЧИСЛО С ПЛАВАЮЩЕЙ ТОЧКОЙ) ЯВЛЯЕТСЯ ЧИСЛО С ПЛАВАЮЩЕЙ ТОЧКОЙ.

ПРИМЕР 1

ON FLOAT, EXP)

A I = (12.3456789E3*F + 3*G)2 + 1/2)**

A I = 9*G + 74074.07340*G*F + 1.52416E+8*F + 5.00000E-1

КОГДА ФЛАГ **FLOAT** СБРОШЕН, ЛЮБОЕ ЧИСЛО В ФОРМЕ С ПЛАВАЮЩЕЙ ТОЧКОЙ АВТОМАТИЧЕСКИ АППРОКСИМИРУЕТСЯ ОТНОШЕНИЕМ ЦЕЛЫХ ЧИСЕЛ.

OFF FLOAT)

A I = 12.35*G)

*****12.350000 REPRESENTED BY 12349999/1000000**

A I = (12349999*G)/1000000

В ЯЗЫКЕ **REDUCE** НЕТ ПРОГРАММ ВЫЧИСЛЕНИЯ ЭЛЕМЕНТАРНЫХ ФУНКЦИЙ **COS**, **SIN** И Т.Д. ДЛЯ ПРОМЕЖУТОЧНЫХ ЧИСЛОВЫХ АРГУМЕНТОВ. И, НАКОНЕЦ, АРИФМЕТИКА В СИСТЕМЕ **REDUCE** НЕДОСТАТОЧНО БЫСТРАЯ. ПРОГРАММИСТУ, КОТОРОМУ НУЖНЫ БОЛЬШИЕ ЧИСЛЕННЫЕ РАСЧЕТЫ ПОСЛЕ

ВСЕХ НЕОБХОДИМЫХ АЛГЕБРАИЧЕСКИХ ПРЕОБРАЗОВАНИЙ, РЕКОМЕНДУЕТСЯ ВЫПОЛНЯТЬ ИХ НА ФОРТРАНЕ ИЛИ ДРУГОМ АНАЛОГИЧНОМ ЯЗЫКЕ. ДЛЯ ЭТОЙ ЦЕЛИ REDUCE ПРЕДОСТАВЛЯЕТ ВОЗМОЖНОСТЬ СОЗДАТЬ ТЕКСТЫ НА ФОРТРАНЕ ДЛЯ ЧИСЛЕННЫХ РАСЧЕТОВ.

W*P** *M**R*E*+2**U*P**V*U*+55**+3**U*0**V** *0*15**+2**

ВО-ПЕРВЫХ, КОГДА ОПЕРАТОР PRINT УПОТРЕБЛЕН В СИСТЕМЕ, БУДЕТ ПЕЧАТАТЬ ВЫРАЖЕНИЯ В ФОРТРАННОМ ПРЕДОСТАВЛЕНИИ: ВЫРАЖЕНИЕ НАЧИНАЕТСЯ В 7 КОЛОНКЕ. ЕСЛИ ВЫРАЖЕНИЕ ПРЕВЫШАЕТ ДЛИНУ СТРОКИ, ТО НА ПОСЛЕДУЮЩИХ СТРОКАХ ПОВЛЯЕТСЯ СИМВОЛ ПРОДОЛЖЕНИЯ (*), ДОПУСКАЕТСЯ 19 СТРОК ПРОДОЛЖЕНИЯ, ПОСЛЕ ЧЕГО НАЧИНАЕТСЯ НОВОЕ ВЫРАЖЕНИЕ, ЕСЛИ ПЕЧАТАЕТСЯ ВЫРАЖЕНИЕ ПРИСВОЕНИЯ ПЕРЕМЕННОЙ, ТО ПЕРВЕННАЯ ПЕЧАТАЕТСЯ КАК ИМЯ ВЫРАЖЕНИЯ, ИНАЧЕ ВЫРАЖЕНИЕ ИМЕНУЕТСЯ TANS;

ВО-ВТОРЫХ, ДЛЯ СОЗДАНИЯ ДРУГИХ ПРОГРАММ МОЖЕТ БЫТЬ ИСПОЛЬЗОВАНА КОМАНДА WRITE.
ПРИМЕР:

```
ON FORT;
OUT FORFIL;
WRITE *C THIS IS A FORTRAN PROGRAM*
WRITE * 1 FORMAT(E,13.5)*
WRITE * U=1.23*
WRITE * V=2.17*
WRITE * W=5.2*
X=(U+V+W)**11;
WRITE *C OF COURSE IT WAS FOOLISH
TO EXPAND THIS EXPRESSION*
WRITE * PRINT 1,X*
WRITE * END*
SHUT FORFIL;
OFF FORT;
```

ЭТИ КОМАНДЫ ЯЗЫКА REDUCE СОЗДАЮТ ФАЙЛ FORFIL, КОТОРЫЙ СОДЕРЖИТ:

```
C THIS IS A FORTRAN PROGRAM
1 FORMAT (E13.5)
U=1.23
V=2.17
W=5.2
X=U**11+11*U**10*V+11*U**10*W+55*U**9*V**2+110*U**9*
X*V*W+55*U**9*W**2+165*U**8*V**3+495*U**8*V**2*W+495
X*U**8*V*W**2+165*U**8*W**3+330*U**7*V**4+1320*U**7*
X*V**3*W+1980*U**7*V**2*W**2+1320*U**7*V*W**3+330*U**
X7*V**4*462*U**6*V**5+2310*U**6*V**4*W+4620*U**6*V**
X3*W**2+4620*U**6*V**2*W**3+2310*U**6*V*W**4+462*U**
X6*W**5+462*U**5*V**6+2772*U**5*V**5*W+6930*U**5*V**
X4*W**2+9240*U**5*V**3*W**3+6930*U**5*V**2*W**6+2772
X*U**5*V*W**5+462*U**5*W**6+330*U**4*V**7+2310*U**4*
X*V**6*W+6930*U**4*V**5*W**2+11550*U**4*V**4*W**3+
X11550*U**4*V**3*W**4+6930*U**4*V**2*W**5+2310*U**4*
X*V**6+330*U**4*W**7+165*U**3*V**8+1320*U**3*V**7*W
X+4620*U**3*V**6*W**2+9240*U**3*V**5*W**3+11550*U**3*
X*V**4*W**4+9240*U**3*V**3*W**5+4620*U**3*V**2*W**6+
X1320*U**3*V*W**7+165*U**5*W**8+55*U**2*V**9+495*U**
X2*V**8*W+1980*U**2*V**7*W**2+4620*U**2*V**6*W**3+
X6930*U**2*V**5*W**4+6930*U**2*V**4*W**5+4620*U**2*V
X**3*W**6+1980*U**2*V**2*W**7+495*U**2*V*W**8+55*U**
X2*W**9+11*U*V**10+110*U*V**9*W+495*U*V**8*W**2+1320
```

```

X*U*V**7*W**3
X=X+2310*U*V**6*W**4+2772*U*V**5*W**5+2310*U*V**4*W
X**6+1320*U*V**3*W**7+405*U*V**2*W**8+110*U*V*W**9+
X11*U*W**10*V**11+11*V**10*W**55*V**9*W**2+165*V**8*W
X**3+330*V**7*W**4+462*V**6*W**5+462*V**5*W**6+330*V
X**4*W**7+165*V**3*W**8+55*V**2*W**9+11*V*W**10+W**
X11.

```

```

C OF COURSE IT WAS FOOLISH TO EXPAND THIS EXPRESSION
PRINT 1,X
END

```

ЧИСЛО КАРТ ПРОДОЛЖЕНИЯ МОЖЕТ БЫТЬ ИЗМЕНЕНО ПРИСВОЕНИЕМ

```

I=CARDNO I= <ЧИСЛО>;

```

ГДЕ <ЧИСЛО> ЕСТЬ ОБЩЕЕ ЧИСЛО КАРТ, ДОПУСТИМЫХ В ОПЕРАТОРЕ.
*CARDNO ПЕРВОНАЧАЛЬНО ПОДАЕТСЯ РАВНЫМ 20.

4.10 РАЗБИЕНИЕ ВЫРАЖЕНИЯ.

В ПРОЦЕССЕ ВЫЧИСЛЕНИЯ ЧАСТО НЕОБХОДИМО ПОЛУЧИТЬ ОТДЕЛЬНЫЕ
ЧАСТИ ВЫРАЖЕНИЯ. REDUCE ПРЕДЛАГАЕТ ТРИ ОПЕРАЦИИ ДЛЯ ЭТОЙ ЦЕ-
ЛИ. ОНИ ПРИГОДНЫ ДЛЯ ШИРОКОГО КРУГА РАСЧЕТОВ.

(1) COEFF

ЭТО ОПЕРАЦИЯ, КОТОРАЯ ВЫДЕЛЯЕТ ЗНАЧЕНИЯ КОЭФФИЦИЕНТОВ ПРИ
РАЗЛИЧНЫХ СТЕПЕНЯХ ЯДРА (СМ. П.5.1). ОНА ИМЕЕТ ТРИ АРГУМЕНТА.
ЕЕ СИНТАКСИС:

```

COEFF (<ВЫРАЖЕНИЕ>, <ЯДРО>, <ИДЕНТИФИКАТОР>)

```

ЕСЛИ <ИДЕНТИФИКАТОР> БЫЛ ПРЕДВАРИТЕЛЬНО ОПИСАН КАК ОДНОМЕР-
НЫЙ МАССИВ, ТО К-МУ ЭЛЕМЕНТУ МАССИВА ПРИСВАИВАЕТСЯ КОЭФФИЦИЕНТ
С 0 ИЛИ ЧЕЛОВЕКА ПРИ КАЖДОЙ СТЕПЕНИ ЯДРА В <ВЫРАЖЕНИИ>. И ТАК - ДО
ЭЛЕМЕНТА МАССИВА С МАКСИМАЛЬНЫМ ИНДЕКСОМ.

ЕСЛИ <ИДЕНТИФИКАТОР> НЕ ЕСТЬ ИМЯ МАССИВА, ТО ПЕРЕМЕННАЯ С
ИМЕНЕМ <ИДЕНТИФИКАТОР> <СТЕПЕНЬ> ПРИСВАИВАЕТСЯ КОЭФФИЦИЕНТ ПРИ
ЭТОЙ СТЕПЕНИ. ТАКИМ СПОСОБОМ ОБРАБАТЫВАЮТСЯ ТОЛЬКО НЕУДЕЛЬНЫЕ
СТЕПЕНИ, ПРИ ЭТОМ ПЕЧАТАЕТСЯ СООБЩЕНИЕ С СООТВЕТСТВУЮЩЕЙ ИНФОР-
МАЦИЕЙ О ПЕРЕМЕННЫХ, КОТОРЫМ ПРИСВОЕНО ЗНАЧЕНИЕ <КОЭФФИЦИЕНТ>.

ПРИМЕР:

```

ARRAY X(7);
COEFF (X**2+2*X**3) W(3), Y(3);

```

ЭТИ КОМАНДЫ ПРИКАЗЫВАЮТ СИСТЕМЕ ЭЛЕМЕНТУ X(7) ПРИСВОИТЬ ЗНА-
ЧЕНИЕ 0, X(6) - ЗНАЧЕНИЕ 1, X(5) - ЗНАЧЕНИЕ 0, X(4) - ЗНА-
ЧЕНИЕ 3*2 И ТАК ДАЛЕЕ ДО X(0), КОТОРОМУ ПРИСВОИТСЯ ЗНАЧЕНИЕ
2**3.

С ДРУГОЙ СТОРОНЫ, ЕСЛИ ЗАПИСАТЬ


```
COEFF((Y**2+Z)**3,Y,W);
```

ТО ПОЛУЧИМ СООБЩЕНИЕ

```
W6 W4 W2 W0
```

- НЕ НУЛИ, И W6 БУДЕТ ПРИСВАИВАТЬСЯ ЗНАЧЕНИЕ 1, W4 - ЗНАЧЕНИЕ 3*2 И Т.Д.

РАЗЛИЧНЫЕ КОЭФФИЦИЕНТЫ, ПОЛУЧЕННЫЕ ЧЕРЕЗ COEFF, МОЖНО РАЗМЕСТИТЬ В ОСОБОЙ КОЛОНКЕ МНОГОМЕРНОГО МАССИВА. ЧТОБЫ СДЕЛАТЬ ЭТО, «ИДЕНТИФИКАТОР» В ОСНОВНОЙ СИНТАКСИЧЕСКОЙ ФОРМЕ НУЖНО ЗАМЕНИТЬ НА ИМЯ МАССИВА, В КОТОРОМ СООТВЕТСТВУЮЩАЯ КОЛОНКА ВЫДЕЛЯЕТСЯ ЗВЕЗДОЧКОЙ.

ПРИМЕР:

```
ARRAY XX(7,7,7);
COEFF ((Y**2+Z)**3,Y,XX(5,*,7));
```

В РЕЗУЛЬТАТЕ XX(5,7,7) БУДЕТ ПРИСВОЕНО ЗНАЧЕНИЕ 0, XX(5,6,7) - ЗНАЧЕНИЕ 1, XX(5,5,7) - ЗНАЧЕНИЕ 0 И Т.Д.

ЗНАЧЕНИЕ COEFF РАВНО САМОЙ ВЫСОКОЙ НЕНУЛЕВОЙ СТЕПЕНИ, ВХОДЯЩЕЙ В ВЫРАЖЕНИЕ. ТАК, В ПРЕДУДУЩЕМ ПРИМЕРЕ ОНО БУДЕТ РАВНО 6.

4.11 КОМАНДЫ, ИСПОЛЬЗУЕМЫЕ В ИНТЕРАКТИВНЫХ СИСТЕМАХ,

СИСТЕМА ПРОГРАММИРОВАНИЯ REDUCE БЫЛА КОНСТРУИРОВАНА ПЕРВОНАЧАЛЬНО ДЛЯ ИНТЕРАКТИВНОГО РЕЖИМА В СИСТЕМЕ С РАЗДЕЛЕНИЕМ ВРЕМЕНИ, НО, ЕСТЕСТВЕННО, ОНА МОЖЕТ ТАКЖЕ РАБОТАТЬ И В ПАКЕТНОМ РЕЖИМЕ. ОДНАКО, ЕСТЬ СУЩЕСТВЕННАЯ РАЗНИЦА МЕЖДУ ИНТЕРАКТИВНЫМ И ПАКЕТНЫМ ИСПОЛЬЗОВАНИЯМИ СИСТЕМЫ. В ПЕРВОМ СЛУЧАЕ, КОГДА СИСТЕМА ОБНАРУЖИВАЕТ НЕОДНОЗНАЧНОСТЬ В КАКОМ-ТО МЕСТЕ ВЫЧИСЛЕНИЯ (НАПРИМЕР, НЕ ПРИСВОЕН ТИП), ТО ОНА ПОТРЕБУЕТ ОТ ПРОГРАММИСТА КОРРЕКТИРОВКИ. В ПАКЕТНОМ РЕЖИМЕ ПРАКТИЧЕСКИ НЕВОЗМОЖНО ОСТАНОВИТЬ ВЫЧИСЛЕНИЕ В ТАКОМ МЕСТЕ И ПОТРЕБОВАТЬ ПРОВЕРКИ ПРОГРАММЫ, ПОЭТОМУ СИСТЕМА ДЕЛАЕТ НАИБОЛЕЕ ОЧЕВИДНОЕ ПРЕДПОЛОЖЕНИЕ О НАМЕРЕНИИ ПРОГРАММИСТА И ПРОДОЛЖАЕТ ВЫЧИСЛЕНИЯ.

ЕСЛИ ВВОД ПРОИСХОДИТ С ВНЕШНЕГО ФАЙЛА, ТО СИСТЕМА ВЫПОЛНЯЕТ ИСХОДНУЮ ПРОГРАММУ В ПАКЕТНОМ РЕЖИМЕ. ДЛЯ ТОГО, ЧТОБЫ ПЕРЕЙТИ К ИНТЕРАКТИВНОМУ РЕЖИМУ, СЛЕДУЕТ УСТАНОВИТЬ ФЛАГ INT ВО ВХОДНОМ ФАЙЛЕ, А ДЛЯ ТОГО, ЧТОБЫ ВЕРНУТЬСЯ К ПАКЕТНОМУ РЕЖИМУ, ФЛАГ INT НУЖНО СБРОСИТЬ. REDUCE ПРЕДЛАГАЕТ ДВЕ КОМАНДЫ ДЛЯ РАБОТЫ В ИНТЕРАКТИВНОМ РЕЖИМЕ. КОМАНДА PAUSE) МОЖЕТ БЫТЬ ПОМЕЩЕНА В ЛЮБОМ МЕСТЕ ФАЙЛА ВВОДА, КОГДА НА ВВОДЕ ВСТРЕЧАЕТСЯ ЭТА КОМАНДА, СИСТЕМА ПОДАЕТ НА ТЕРМИНАЛ СООБЩЕНИЕ CONT (С ВОПРОСИТЕЛЬНЫМ ЗНАКОМ) И ПРИОСТАНАВЛИВАЕТ ВЫПОЛНЕНИЕ ПРОГРАММЫ. ЕСЛИ ПРОГРАММИСТ ОТВЕЧАЕТ Y (ДА), ТО ВЫЧИСЛЕНИЯ ПРОДОЛЖАЮТСЯ С ТОГО ЖЕ САМОГО МЕСТА В ФАЙЛЕ А ЕСЛИ ПРОГРАММИСТ ОТВЕЧАЕТ N (НЕТ), ТО УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ НА ТЕРМИНАЛ, И ПРОГРАММИСТ МОЖЕТ ВВОДИТЬ ДРУГИЕ КОМАНДЫ, ОДНАКО, ПОЗЖЕ ОН МОЖЕТ ВОСПОЛЬЗОВАТЬСЯ КОМАНДОЙ CONT, - УПРАВЛЕНИЕ ТОГДА БУДЕТ ПЕРЕДАНО ОПЯТЬ В ТУ ЖЕ ТОЧКУ ВХОДНОГО ФАЙЛА, ГДЕ ПОСЛЕДНИЙ РАЗ БЫЛО СДЕЛАНО ПРЕРЫВАНИЕ (ПО КОМАНДЕ PAUSE);).

4.12 КОМАНДА DEFINE.

КОМАНДА DEFINE ПОЗВОЛЯЕТ ПРОГРАММИСТУ ПЕРМАНЕНТНО ПЕРЕИМЕНОВЫВАТЬ ЛЮБОЙ ИДЕНТИФИКАТОР В СИСТЕМЕ. ЕЕ АРГУМЕНТОМ ЯВЛЯЕТСЯ СПИСОК ВЫРАЖЕНИЯ В ВИДЕ:

<ИДЕНТИФИКАТОР> = <ЧИСЛО> [<ИДЕНТИФИКАТОР> [<ОПЕРАЦИЯ> [<СЛУЖЕБНОЕ СЛОВО> [<ВЫРАЖЕНИЕ>

НАПРИМЕР:

DEFINE BE = (X=Y+Z)

ОЗНАЧАЕТ, ЧТО С МОМЕНТА ПРОЧТЕНИЯ ДАННОЙ КОМАНДЫ "BE" (БУДЕТ) ИНТЕРПРЕТИРОВАТЬСЯ КАК ЗНАК РАВЕНСТВА, А X — КАК ВЫРАЖЕНИЕ

Y + Z.

ЭТО ПЕРЕИМЕНОВАНИЕ ДЕЛАЕТСЯ НА УРОВНЕ ВВОДА И, СЛЕДОВАТЕЛЬНО, ИМЕЕТ ПРИОРИТЕТ НАД ЛЮБОЙ ДРУГОЙ ЗАМЕНОЙ, ОПИСАННОЙ ДЛЯ ТОГО ЖЕ САМОГО ИДЕНТИФИКАТОРА.

4.13 УНИЧТОЖЕНИЕ НУЛЕЙ НА ВЫВОДЕ. ФЛАГ ZERO.

ИНОГДА, ОСОБЕННО ПРИ ПЕЧАТАНИИ БОЛЬШИХ МАССИВОВ, НЕ ЖЕЛАТЕЛЬНО, ЧТОБЫ ПРИСВАИВАНИЯ НУЛЕЙ (Т.Е. ПРИСВАИВАНИЯ ВИДА <ВЫРАЖЕНИЕ> X=0) ВЫВОДИЛИСЬ НА ПЕЧАТЬ. ВЫВОД ТАКИХ ПРИСВАИВАНИЙ МОЖНО ОТМЕНИТЬ ВКЛЮЧЕНИЕМ ФЛАГА ZERO.

4.14 АСИМПТОТИЧЕСКИЕ КОМАНДЫ.

ЕСЛИ РАСКРЫТИЕ ПОЛИНОМА СВЯЗАНО С ПЕРЕМЕННЫМИ, ПРО КОТОРЫЕ ИЗВЕСТНО, ЧТО ОНИ МАЛЫ, ТО ЧАСТО ОКАЗЫВАЕТСЯ РАЗУМНЫМ УДАЛИТЬ ВСЕ СТЕПЕНИ ЭТИХ ПЕРЕМЕННЫХ, БОЛЬШЕ НЕКОТОРЫХ, ЧТОБЫ ИЗБАВИТЬСЯ ОТ НЕНУЖНЫХ ВЫЧИСЛЕНИЙ. ДЛЯ ЭТОГО МОЖНО ВОСПОЛЬЗОВАТЬСЯ КОМАНДОЙ LET.

ПРИМЕР:

ЕСЛИ НУЖНЫ СТЕПЕНИ X ДО X**7, ТО КОМАНДА

LET X**8=0

ЗАСТАВИТ СИСТЕМУ ВЫЧЕРКНУТЬ ВСЕ СТЕПЕНИ X, БОЛЬШЕ 7.

ЭТОТ МЕТОД, ОДНАКО, НЕОДНОЗНАЧЕН, КОГДА ВЫРАЖЕНИЕ ВКЛЮЧАЕТ НЕСКОЛЬКО ПЕРЕМЕННЫХ РАЗЛИЧНЫХ СТЕПЕНЕЙ МАЛОСТИ. В ЭТОМ СЛУЧАЕ НЕОБХОДИМО СНАБДИТЬ КАЖДУЮ ПЕРЕМЕННУЮ АСИМПТОТИЧЕСКИМ ВЕСОМ И ПОДСЧИТАТЬ ОБЩИЙ ВЕС КАЖДОГО ПРОИЗВЕДЕНИЯ В РАСКРЫТОМ ВЫРАЖЕНИИ ПРЖДЕ, ЧЕМ РЕШАТЬ, ОСТАВИТЬ ЧЛЕН ИЛИ НЕТ. ДЛЯ ВЫПОЛНЕНИЯ ЭТОГО ТИПА АСИМПТОТИЧЕСКИХ ПОСТРОЕНИЙ В СИСТЕМЕ СУЩЕСТВУЕТ ДВЕ КОМАНДЫ. КОМАНДА WEIGHT БЕРЕТ СПИСОК РАВЕНСТВ В ФОРМЕ

<ЯДЕРНАЯ ФОРМА>=<ЧИСЛО>

ЗДЕСЬ <ЧИСЛО> ДОЛЖНО БЫТЬ ПОЛОЖИТЕЛЬНЫМ И ЦЕЛЫМ. ЭТА КОМАНДА ПРИСВАИВАЕТ ВЕС, РАВНЫЙ ЧИСЛУ, СООТВЕТСТВУЮЩЕЙ ЯДЕРНОЙ ФОРМЕ. ЗАТЕМ ДЕЛАЕТСЯ ПРОВЕРКА ~~В~~ АЛГЕБРАИЧЕСКИХ ВЫРАЖЕНИИ С ЦЕЛЮ ВЫЯСНИТЬ, БОЛЬШЕ ЛИ ОБЩИЙ ВЕС ЧЛЕНА, ЧЕМ ВЕСОВОЙ УРОВЕНЬ, ПРИСВОЕННЫЙ ВЫЧИСЛЕНИЮ. ЕСЛИ ЭТО ТАК, ТО ЧЛЕН ~~ВЫЧЕРКИВАЕТСЯ~~, ЧТОБЫ ВЫЧИСЛИТЬ ОБЩИЙ ВЕС ПРОИЗВЕДЕНИЯ, НУЖНО УМНОЖИТЬ ИНДИВИДУАЛЬНЫЕ ВЕСА КАЖДОЙ ЯДЕРНОЙ ФОРМЫ НА СООТВЕТСТВУЮЩИЕ ИМ СТЕПЕНИ И ЗАТЕМ СЛОЖИТЬ.

ВЕСОВОЙ УРОВЕНЬ СИСТЕМЫ ПЕРВОНАЧАЛЬНО ПРЕДПОЛАГАЕТСЯ РАВНЫМ 2. ПРОГРАММИСТ МОЖЕТ ИЗМЕНИТЬ ЭТО ЗНАЧЕНИЕ КОМАНДОЙ

WTLEVEL <ЧИСЛО>;

КОТОРАЯ ПРЕДПИСЫВАЕТ СИСТЕМЕ <ЧИСЛО> В КАЧЕСТВЕ НОВОГО ВЕСОВОГО УРОВНЯ. <ЧИСЛО> ДОЛЖНО БЫТЬ ЦЕЛЫМ И ПОЛОЖИТЕЛЬНЫМ.

(0, 0) (0, 0)

((0, 0), (0, 0))

... ..

... ..

... ..

... ..

МОЩНЫМ ИНСТРУМЕНТОМ СИСТЕМЫ REDUCE ЯВЛЯЕТСЯ ВОЗМОЖНОСТЬ ВЫПОЛНЯТЬ МАТРИЧНЫЕ ПРЕОБРАЗОВАНИЯ. ДЛЯ ЭТИХ ЦЕЛЕЙ В СИСТЕМУ ВСТРОЕНА ОПЕРАЦИЯ МАТ И ВВЕДЕН ЕЩЕ ОДИН ТИП ПЕРЕМЕННЫХ И ВЫРАЖЕНИЙ.

5.1 ОПЕРАЦИЯ МАТ:

ЭТА ПРЕФИКСНАЯ ОПЕРАЦИЯ ИСПОЛЬЗУЕТСЯ ДЛЯ ПРЕДСТАВЛЕНИЯ МАТРИЦ РАЗМЕРНОСТИ $K \times M$. МАТ ИМЕЕТ К АРГУМЕНТОВ, ИНТЕРПРЕТИРУЕМЫХ КАК СТРОКИ МАТРИЦЫ. КАЖДАЯ ИЗ АРГУМЕНТОВ ЯВЛЯЕТСЯ СПИСКОМ ИЗ M ВЫРАЖЕНИЙ, ПРЕДСТАВЛЯЮЩИХ СОБОЙ ЭЛЕМЕНТЫ ЭТИХ СТРОК.

ПРИМЕР:
МАТРИЦА

(A B C)
(D E F)

В ЯЗЫКЕ REDUCE ПРЕДСТАВЛЯЕТСЯ В ВИДЕ

МАТ ((A,B,C),(D,E,F))

ЗАМЕЧАНИЕ.

АРГУМЕНТЫ ОПЕРАЦИИ МАТ ОБЯЗАТЕЛЬНО ДОЛЖНЫ ЗАКЛЮЧАТЬСЯ В СКОБКИ, ДАЖЕ ЕСЛИ ПРЕДСТАВЛЯЕТСЯ МАТРИЦА, СОДЕРЖАЩАЯ 1 СТОЛБЕЦ ИЛИ 1 ЭЛЕМЕНТ.

5.2 МАТРИЧНЫЕ ПЕРЕМЕННЫЕ:

МАТРИЧНЫЕ ПЕРЕМЕННЫЕ ОБОЗНАЧАЮТ НЕ ОТДЕЛЬНЫЕ МАТРИЧНЫЕ ЭЛЕМЕНТЫ, А МАТРИЦЫ ЦЕЛИКОМ.

ИМЕНА МАТРИЧНЫХ ПЕРЕМЕННЫХ ВВОДЯТСЯ С ПОМОЩЬЮ ОПИСАНИЯ MATRIX. РЯДОМ С ИМЕНЕМ МАТРИЧНОЙ ПЕРЕМЕННОЙ МОЖНО УКАЗАТЬ РАЗМЕРНОСТЬ СООТВЕТСТВУЮЩЕЙ МАТРИЦЫ. ЕСЛИ РАЗМЕРНОСТЬ НЕИЗВЕСТНА ЗАРАНЕЕ, ЕЕ МОЖНО ЗАДАТЬ НЕЯВНО, ПРИСВАИВАЯ МАТРИЧНОЙ ПЕРЕМЕННОЙ НЕКОТОРОЕ МАТРИЧНОЕ ВЫРАЖЕНИЕ. (СМ. П.П. 4.4 И 4.6).

ПРИМЕР:

MATRIX X(2,1),Y(3,4),Z

ЭТО ОПИСАНИЕ ОБЪЯВЛЯЕТ X КАК МАТРИЦУ РАЗМЕРОМ 2×1 (МАТРИЦУ-СТОЛБЕЦ), Y - КАК МАТРИЦУ РАЗМЕРОМ 3×4 , Z - КАК МАТРИЦУ НЕОПРЕДЕЛЕННОГО РАЗМЕРА. РАЗМЕРНОСТЬ Z БУДЕТ ОПИСАНА ПОЗЖЕ, НЕЯВНО, ВЫПОЛНЕНИЕМ МАТРИЧНОГО ПРИСВАИВАНИЯ. (СМ. ПРИМЕР К ЭТОМУ ПАРАГРАФУ).

ЕСЛИ РАЗМЕРНОСТЬ МАТРИЦЫ ОПИСАНА С ПОМОЩЬЮ ОПИСАНИЯ MATRIX, ТО ВСЕМ ЕЕ ЭЛЕМЕНТАМ ПЕРВОНАЧАЛЬНО ПРИСВАИВАЕТСЯ ЗНАЧЕНИЕ 0.

ЗАМЕЧАНИЕ.

ЕСЛИ МАТРИЦА ВВЕДЕНА ПОСРЕДСТВОМ ОПИСАНИЯ MATRIX, ТО ССЫЛАТЬСЯ НА ОТДЕЛЬНЫЙ ЭЛЕМЕНТ ЭТОЙ МАТРИЦЫ ИЛИ НА МАТРИЦУ В ЦЕЛОМ МОЖНО ТОЛЬКО ПОСЛЕ ПРЕДСТАВЛЕНИЯ РАЗМЕРНОСТИ МАТРИЦЫ И ПРЕДСТАВЛЕНИЯ ЗНАЧЕНИЙ ЕЕ ЭЛЕМЕНТОВ ВЫПОЛНЕНИЕМ МАТРИЧНОГО

ПРИСВАИВАНИЯ.

ПРИМЕР :

MATRIX E(4,1),F,H;

H := MAT((LOG(G), (G+3), (G/5/7)))

H(1,1) := LOG(G)

H(1,2) := G+3

H(2,1) := G

H(2,2) := 5/7

ЕСЛИ С ПОМОЩЬЮ ОПЕРАЦИИ MAT. ПРЕДСТАВЛЯЕТСЯ МАТРИЦА БОЛЬШОЙ РАЗМЕРНОСТИ, РАСПЕЧАТКА ЗНАЧЕНИЯ ЕЕ ЭЛЕМЕНТОВ МОЖЕТ ЗАНЯТЬ МНОГО МЕСТА, ПОЭТОМУ ПЕЧАТЬ В ЭТОМ СЛУЧАЕ ЛУЧШЕ ВЫКЛЮЧИТЬ С ПОМОЩЬЮ ЗНАКА ДОЛЛАРА.

5.3 МАТРИЧНЫЕ ВЫРАЖЕНИЯ.

ОНИ СЛЕДУЮТ ОБЫЧНЫМ ПРАВИЛАМ МАТРИЧНОЙ АЛГЕБРЫ, ЧТО ОПРЕДЕЛЯЕТСЯ СЛЕДУЮЩИМ СИНТАКСИСОМ :

<МАТРИЧНОЕ ВЫРАЖЕНИЕ> ::= MAT <АРГУМЕНТЫ ОПЕРАЦИИ> [<МАТРИЧНАЯ ПЕРЕМЕННАЯ> [<СКАЛЯРНОЕ ВЫРАЖЕНИЕ>* <МАТРИЧНОЕ ВЫРАЖЕНИЕ> [<МАТРИЧНОЕ ВЫРАЖЕНИЕ>* <МАТРИЧНОЕ ВЫРАЖЕНИЕ> [<МАТРИЧНОЕ ВЫРАЖЕНИЕ>* <МАТРИЧНОЕ ВЫРАЖЕНИЕ> [<МАТРИЧНОЕ ВЫРАЖЕНИЕ> ** <ЦЕЛЫЙ> [<МАТРИЧНОЕ ВЫРАЖЕНИЕ> / <МАТРИЧНОЕ ВЫРАЖЕНИЕ>

В СУММАХ И ПРОИЗВЕДЕНИЯХ МАТРИЧНЫХ ВЫРАЖЕНИЯ РАЗМЕРНОСТИ ДОЛЖНЫ БЫТЬ ПРИВЕДЕНЫ В СООТВЕТСТВИЕ, ИНАЧЕ РЕЗУЛЬТАТОМ ВЫЧИСЛЕНИЯ БУДЕТ ОШИБКА. АНАЛОГИЧНО, ТОЛЬКО КВАДРАТНЫЕ МАТРИЦЫ МОГУТ БЫТЬ ВОЗВЕДЕНЫ В СТЕПЕНЬ. ОТРИЦАТЕЛЬНЫЕ СТЕПЕНИ ВЫЧИСЛЯЮТСЯ КАК ОБРАТНЫЕ МАТРИЦЫ, ВОЗВЕДЕННЫЕ В СООТВЕТСТВУЮЩИЕ СТЕПЕНИ. A/B ИНТЕРПРЕТИРУЕТСЯ КАК $A*B^{(-1)}$.

ПРИМЕРЫ:

ПРЕДПОЛОЖИМ, ЧТО X, Y БЫЛИ ОПИСАНЫ КАК МАТРИЦЫ, ТОГДА СЛЕДУЮЩИЕ ВЫРАЖЕНИЯ ЯВЛЯЮТСЯ МАТРИЧНЫМИ:

Y
Y**2*X-3*Y**(-2)*X
Y+ MAT((1,A), (B,C))/2

5.4 ОПЕРАЦИИ С МАТРИЧНЫМИ АРГУМЕНТАМИ.

В МАТРИЧНЫХ ВЫЧИСЛЕНИЯХ ИСПОЛЬЗУЮТСЯ ТРИ ОПЕРАЦИИ: **DET**, **TR**, **TRACE**.

1. ОПЕРАЦИЯ **DET**.

ОПЕРАЦИЯ **DET** ИСПОЛЬЗУЕТСЯ ДЛЯ ПОЛУЧЕНИЯ ДЕТЕРМИНАНТА МАТРИЧНОГО ВЫРАЖЕНИЯ. ЕЕ СИНТАКСИС:

DET <МАТРИЧНОЕ ВЫРАЖЕНИЕ>;

ПРИМЕРЫ:

DET(Y2)**

-ЭТО СКАЛЯРНОЕ ВЫРАЖЕНИЕ. ЕГО ЗНАЧЕНИЕ ЕСТЬ ДЕТЕРМИНАНТ КВАДРАТА МАТРИЦЫ **Y**.

DET MAT((A,B,C),(D,E,F),(G,H,I));

-ЭТО СКАЛЯРНОЕ ВЫРАЖЕНИЕ, ПРЕДСТАВЛЯЮЩЕЕ СОБОЙ ДЕТЕРМИНАНТ МАТРИЦЫ

$$\begin{pmatrix} A & B & C \\ D & E & F \\ G & H & I \end{pmatrix}$$
2. ОПЕРАЦИЯ **TR**.

ЭТА ОПЕРАЦИЯ ТРАНСПОНИРУЕТ ОТДЕЛЬНУЮ МАТРИЦУ. ЕЕ ЗНАЧЕНИЕМ ЯВЛЯЕТСЯ МАТРИЦА, ТРАНСПОНИРОВАННАЯ ПО ОТНОШЕНИЮ К ИСХОДНОЙ, ЕЕ СИНТАКСИС:

TR <МАТРИЧНОЕ ВЫРАЖЕНИЕ>;

ИСПОЛЬЗОВАНИЕ ОПЕРАЦИИ ОЧЕВИДНО.

ЭТА ОПЕРАЦИЯ ИСПОЛЬЗУЕТСЯ ДЛЯ ПОЛУЧЕНИЯ ШПУРА КВАДРАТНОЙ МАТРИЦЫ. ЕЕ СИНТАКСИС:

TRACE <МАТРИЧНОЕ ВЫРАЖЕНИЕ>;

ИСПОЛЬЗОВАНИЕ ОПЕРАЦИИ ОЧЕВИДНО.

3. НАХОЖДЕНИЕ ОБРАТНОЙ МАТРИЦЫ.

МАТРИЦУ, ОБРАТНУЮ К ДАННОЙ МАТРИЦЕ **A**, МОЖНО НАЙТИ ВЫПОЛНЕНИЕМ ОПЕРАЦИИ **A**(-1)** ИЛИ **1/A**.

НАПРИМЕР, ДЛЯ ТОГО, ЧТОБЫ НАЙТИ РЕШЕНИЕ УРАВНЕНИЯ

$$A * B = \text{MAT}((C), (2)),$$

ПРАВУЮ ЧАСТЬ ЭТОГО УРАВНЕНИЯ НУЖНО УМНОЖИТЬ СЛЕВА НА МАТРИЦУ, ОБРАТНУЮ К А:

$$B := 1/A * \text{MAT}((C), (2));$$

ЗАМЕТИМ, ЧТО $\text{MAT}((C), (2))/A$ БУДЕТ ОЗНАЧАТЬ УМНОЖЕНИЕ СПРАВА, А ЭТО НЕ ТО, ЧТО ТРЕБУЕТСЯ.

ЗАМЕЧАНИЕ.

"ВЫЧИСЛЕНИЕ" ВЫРАЖЕНИЯ ВИДА $1/A * \dots$ ИЛИ \dots / A , БОЛЕЕ ЭФФЕКТИВНО, ЧЕМ ОТДЕЛЬНОЕ ВЫЧИСЛЕНИЕ ОБРАТНОЙ МАТРИЦЫ И ПОСЛЕДУЮЩЕЕ ЕЕ ПЕРЕМНОЖЕНИЕ С ДРУГИМИ СМНОЖИТЕЛЯМИ. ОТДЕЛЬНОЕ ВЫЧИСЛЕНИЕ И ЗАПОМИНАНИЕ ОБРАТНОЙ МАТРИЦЫ ОПРАВДАНО, ТОЛЬКО ЕСЛИ ОНА БУДЕТ МНОГОКРАТНО ИСПОЛЬЗОВАТЬСЯ.

5,5 МАТРИЧНЫЕ ПРИСВАИВАНИЯ,

МАТРИЧНЫЕ ВЫРАЖЕНИЯ МОГУТ ПОЯВЛЯТЬСЯ В ПРАВОЙ ЧАСТИ ОПЕРАТОРА ПРИСВАИВАНИЯ. ЕСЛИ ЛЕВАЯ ЧАСТЬ В ОПЕРАТОРЕ ПРИСВАИВАНИЯ, КОТОРАЯ ДОЛЖНА БЫТЬ ПЕРЕМЕННОЙ, НЕ ОПИСАНА УЖЕ КАК МАТРИЦА, ОНА ОБЪЯВЛЯЕТСЯ ТАКОВОЙ НЕЯВНО УКАЗАНИЕМ РАЗМЕРНОСТИ В ПРАВОЙ ЧАСТИ.

ТАКОЕ ПРИСВАИВАНИЕ МОЖЕТ БЫТЬ ОЧЕНЬ УДОБНЫМ ПРИ НАХОЖДЕНИИ РЕШЕНИЯ СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ.

ПРИМЕР:

ЧТОБЫ НАЙТИ РЕШЕНИЕ СЛЕДУЮЩЕЙ СИСТЕМЫ УРАВНЕНИЙ:

$$\begin{aligned} (0, 0, 0, 0) & \dots (0, 0, 0) \\ A_{11} * X(1) + A_{12} * X(2) &= Y_1 \\ A_{21} * X(1) + A_{22} * X(2) &= Y_2 \end{aligned}$$

ПРОСТО ЗАПИСЫВАЕМ:

$$X := 1/\text{MAT}((A_{11}, A_{12}), (A_{21}, A_{22})) * \text{MAT}((Y_1), (Y_2));$$

ДЛЯ СОВМЕСТНОГО РЕШЕНИЯ НЕСКОЛЬКИХ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ С РАЗНЫМИ ПРАВЫМИ ЧАСТЯМИ ДОСТАТОЧНО ОДНОГО МАТРИЧНОГО ПРИСВАИВАНИЯ:

$$X := 1/\text{MAT}((A_{11}, A_{12}), (A_{21}, A_{22})) * \text{MAT}((Y_1, Z_1, W_1), (Y_2, Z_2, W_2));$$

5,6 ВЫЧИСЛЕНИЕ МАТРИЧНЫХ ЭЛЕМЕНТОВ.

ЕСЛИ ЭЛЕМЕНТУ МАТРИЦЫ ПРИСВОЕНО ЗНАЧЕНИЕ, ТО НА НЕГО МОЖНО ССЫЛАТЬСЯ, КАК И НА ЭЛЕМЕНТ СТАНДАРТНОГО МАССИВА. ТАК, $Y(2, 1)$ ПОДРАЗУМЕВАЕТ ЭЛЕМЕНТ ВО ВТОРОЙ СТРОКЕ И ПЕРВОМ СТОЛБЦЕ МАТРИЦЫ Y.

В ССЫЛКАХ НА ОТДЕЛЬНЫЙ МАТРИЧНЫЙ ЭЛЕМЕНТ ОБЯЗАТЕЛЬНО ДОЛЖНЫ УКАЗЫВАТЬСЯ ДВА ИНДЕКСА.

ЗАМЕЧАНИЕ: ГРЕЧЕСКИЕ БУКВЫ В ЭТОЙ ГЛАВЕ ПРЕДСТАВЛЕНЫ ИХ АНГЛИЙСКИМИ НАЗВАНИЯМИ.

6.1 ВВЕДЕНИЕ.

В СИСТЕМЕ REDUCE ИМЕЕТСЯ НАБОР КОМАНД, ПРЕДНАЗНАЧЕННЫХ ДЛЯ СИМВОЛЬНЫХ РАСЧЕТОВ В ЗАДАЧАХ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ. ВВЕДЕНИЕ БОЛЕЕ СЛОЖНЫХ СТРУКТУР ДАННЫХ, ЕСТЕСТВЕННО, ПРЕДПОЛАГАЕТ НЕКОТОРОЕ РАСШИРЕНИЕ БАЗИСНОГО СИНТАКСИСА СИСТЕМЫ.

6.2 ОПЕРАЦИИ, ИСПОЛЬЗУЕМЫЕ В ЗАДАЧАХ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ.

ВВЕДЕМ ТРИ НОВЫХ ОПЕРАЦИИ.

1. ОПЕРАЦИЯ : (ТОЧКА).

ОПЕРАЦИЯ : — БИНАРНАЯ ОПЕРАЦИЯ, ИСПОЛЬЗУЕМАЯ В АЛГЕБРАИЧЕСКОМ РЕЖИМЕ ДЛЯ ОБОЗНАЧЕНИЯ СКАЛЯРНОГО ПРОИЗВЕДЕНИЯ ДВУХ ЛОРЕНЦЕВЫХ 4-ВЕКТОРОВ.

КОМПОНЕНТЫ ВЕКТОРОВ МОГУТ БЫТЬ ПРЕДСТАВЛЕНЫ ЧЕРЕЗ ЕДИНИЧНЫЙ ВЕКТОР. ТАК, ЕСЛИ E_0 ПРЕДСТАВЛЯЕТ ЕДИНИЧНЫЙ ВЕКТОР $(1,0,0,0)$, ТО $(P.E_0)$ НУЛЕВАЯ КОМПОНЕНТА 4-ВЕКТОРА P .

(НАША МЕТРИКА И ОБОЗНАЧЕНИЯ СЛЕДУЮТ КНИГЕ БЕРКЕНА И ДРЕЛЛА [7]). АНАЛОГИЧНО, ПРОИЗВОЛЬНАЯ КОМПОНЕНТА P МОЖЕТ БЫТЬ ПРЕДСТАВЛЕНА В ВИДЕ $(P.U)$ ЕСЛИ ТРЕБУЕТСЯ СВЕРТКА ПО КОМПОНЕНТАМ ВЕКТОРОВ, ТО НАДО ВОСПОЛЬЗОВАТЬСЯ ОПИСАНИЕМ INDEX :. ТАК,

INDEX U)

ОПИСЫВАЕТ U КАК ИНДЕКС, И СВЕРТКА $(P.U)*(Q.U)$ ДАСТ В РЕЗУЛЬТАТЕ $(P.Q)$.

МЕТРИЧЕСКИЙ ТЕНЗОР G ПРЕДСТАВЛЯЕТСЯ В ВИДЕ $(U.V)$

ЕСЛИ НУЖНА СВЕРТКА ПО U И V , ТО $U.V$ ДОЛЖНЫ БЫТЬ ОПИСАНЫ КАК ИНДЕКСЫ.

ОПИСАНИЕ

REMIIND V1...VN δ

СТИРАЕТ ИНДЕКСНЫЕ ФЛАГИ ДЛЯ ПЕРЕМЕННЫХ $V1...VN$. ОДНАКО, В СИСТЕМЕ ЭТИ ПЕРЕМЕННЫЕ ОСТАЮТСЯ ВЕКТОРАМИ.

2, ОПЕРАЦИЯ G.

ЭТО МНОГОМЕСТНАЯ ОПЕРАЦИЯ, ИСПОЛЬЗУЕМАЯ ДЛЯ ОБОЗНАЧЕНИЯ ПРОИЗВЕДЕНИЯ СВЕРТОК ГАММА-МАТРИЦ (4-ВЕКТОРАМИ ЛОРЕНЦА, ГАММА-МАТРИЦЫ СОПОСТАВЛЕННЫ ФЕРМИОННЫМ ЛИНИЯМ НА ФЕЙНМАНОВСКИХ ДИАГРАММАХ. ЕСЛИ ДИАГРАММА ВКЛЮЧАЕТ БОЛЕЕ, ЧЕМ ОДНУ ТАКУЮ ЛИНИЮ, ТО ДЛЯ ПРЕДСТАВЛЕНИЯ КАЖДОЙ ЛИНИИ ТРЕБУЕТСЯ ДРУГОЙ НАБОР ГАММА-МАТРИЦ, ДЕЙСТВУЮЩИХ В НЕЗАВИСИМЫХ СПИНОВЫХ ПРОСТРАНСТВАХ, ПЕРВЫМ АРГУМЕНТОМ ОПЕРАЦИИ G ЯВЛЯЕТСЯ ИДЕНТИФИКАТОР (НЕ ЧИСЛО), ОБОЗНАЧАЮЩИЙ ФЕРМИОННУЮ ЛИНИЮ, ЧТО ДАЕТ ВОЗМОЖНОСТЬ РАЗЛИЧАТЬ РАЗНЫЕ ЛИНИИ.

ТАК, $G(L1, P) * G(L2, Q)$ ОБОЗНАЧАЕТ ПРОИЗВЕДЕНИЕ 4-ВЕКТОРА P, СВЯЗАННОГО С ФЕРМИОННОЙ ЛИНИЕЙ L1 И 4-ВЕКТОРА Q, СВЯЗАННОГО С ЛИНИЕЙ L2.

ПРОИЗВЕДЕНИЕ ГАММА-МАТРИЦ, СОПОСТАВЛЕННЫХ ОДНОЙ И ТОЙ ЖЕ ЛИНИИ, МОЖЕТ БЫТЬ ЗАПИСАНО В СВЕРНУТОЙ ФОРМЕ, НАПРИМЕР:

$$G(L1, P1, P2, \dots, P3) = G(L1, P1) * G(L1, P2) * \dots * G(L1, P3).$$

ОПЕРАЦИЯ G ИМЕЕТ ЗАРЕЗЕРВИРОВАННЫЙ АРГУМЕНТ A, ПОЛУЧАШИЙ ДЛЯ ОБОЗНАЧЕНИЯ СПЕЦИАЛЬНОЙ ГАММА-МАТРИЦЫ GAMMA, ТАК, ЗАПИСЬ $G(L, A)$ ОБОЗНАЧАЕТ МАТРИЦУ GAMMA, СОПОСТАВЛЕННУЮ ЛИНИИ L, А $G(L, P, A) =$ МАТРИЦУ GAMMA, P + GAMMA, СОПОСТАВЛЕННУЮ ЛИНИИ L.

МАТРИЦА

GAMMA

U

СОПОСТАВЛЕННАЯ ЛИНИИ L, МОЖЕТ БЫТЬ ЗАПИСАНА КАК $G(L, U)$ (ЕСЛИ ТРЕБУЕТСЯ СВЕРТКА ПО U, ТО U ДОЛЖНО БЫТЬ ОПИСАНО КАК ИНДЕКС). ОБОЗНАЧЕНИЯ СОГЛАСНО [7] ПРЕДПОЛАГАЮТСЯ ВО ВСЕХ ОПЕРАЦИЯХ С ГАММА-МАТРИЦАМИ.

3, ОПЕРАЦИЯ EPS.

ОПЕРАЦИЯ EPS ИМЕЕТ 4 АРГУМЕНТА И ИСПОЛЬЗУЕТСЯ ТОЛЬКО ДЛЯ ОБОЗНАЧЕНИЯ ПОЛНОСТЬЮ АНТИСИММЕТРИЧНОГО ТЕНЗОРА 4 РАНГА И ЕГО СВЕРТКИ С ЛОРЕНЦЕВЫМИ 4-ВЕКТОРАМИ. ТАК,

$$EPS_{IJKL}$$

(+1, ЕСЛИ

I, J, K, L =

ЧЕТНАЯ ПЕРЕСТАНОВКА 0, 1, 2, 3

(-1, ЕСЛИ ЭТО НЕЧЕТНАЯ ПЕРЕСТАНОВКА

(0 В ДРУГИХ СЛУЧАЯХ.

СВЕРТКА

EPS P Q

IJUV U V

МОЖЕТ БЫТЬ ПЕРЕПИСАНА КАК $EPS(I, J, P, Q)$ ПРИ ЭТОМ I, J,

ДОЛЖНЫ БЫТЬ ОПИСАНЫ КАК ИНДЕКСЫ.

6.3 ВЕКТОРНЫЕ ПЕРЕМЕННЫЕ

КРОМЕ ИДЕНТИФИКАТОРОВ ФЕРМИОННЫХ ЛИНИЙ В ОПЕРАЦИИ **Q**, ВСЕ АРГУМЕНТЫ ОПЕРАЦИИ В ЭТИХ ГЛАВЕ ЯВЛЯЮТСЯ

ИМЕНА ВЕКТОРОВ (ВЕКТОРНЫЕ ПЕРЕМЕННЫЕ) ДОЛЖНЫ БЫТЬ ВВЕДЕНЫ ПОСРЕДСТВОМ ОПИСАНИЯ ТИПА МЕСТОС.

ПРИМЕР:

VECTOR P1,P2)

ЭТА КОМАНДА ОПИСЫВАЕТ *P1* И *P2* КАК ВЕКТОРЫ. (= (S1,...,S9,T9,R))

ПЕРЕМЕННЫЕ, А ОПИСАННЫЕ КАК ИНДЕКСЫ И И ПЕРЕМЕННЫЕ, КОТОРЫМ ПРИСВОЕНА МАССА (СТРАНА 6) АВТОМАТИЧЕСКИ СЧИТАЮТСЯ ВЕКТОРНЫМИ.

6.4 ДОПОЛНИТЕЛЬНЫЕ ТИПЫ ВЫРАЖЕНИЯ.

ДЛЯ ВЫЧИСЛЕНИЯ В ЗАДАЧАХ ФИЗИКИ ВЫСОКИХ ЭНЕРГИИ НЕОБХОДИМЫ ВЫРАЖЕНИЯ ЕЩЕ ДВУХ ТИПОВ, А ИМЕННО:

1. ВЕКТОРНЫЕ ВЫРАЖЕНИЯ.

ПРАВИЛА ДЛЯ ОПРЕДЕЛЕНИЯ ТИПА КОМБИНИРОВАННЫХ ВЫРАЖЕНИЙ ОБЫЧНЫЕ. ТАК, ПРОИЗВЕДЕНИЕ СКАЛЯРНОГО ИЛИ ЧИСЛОВОГО ВЫРАЖЕНИЯ НА ВЕКТОРНОЕ ЕСТЬ ВЕКТОР, ТАК ЖЕ, КАК СУММА ИЛИ РАЗНОСТЬ ВЕКТОРНЫХ ВЫРАЖЕНИЙ. ЕСЛИ ЭТИ ПРАВИЛА НЕ СОБЛЮДАЮТСЯ, ПЕЧАТАЕТСЯ СООБЩЕНИЕ ОБ ОШИБКЕ. ДАЛЕЕ, ЕСЛИ СИСТЕМА ОБНАРУЖИТ НЕОПИСАННУЮ ПЕРЕМЕННУЮ НА МЕСТЕ, ГДЕ ОЖИДАЕТСЯ ВЕКТОРНАЯ ПЕРЕМЕННАЯ, ТО ОНА СПРОСИТ У ПРОГРАММИСТА, СЧИТАТЬ ЭТУ ПЕРЕМЕННУЮ ВЕКТОРОМ, ИЛИ НЕТ (ПРИ РАБОТЕ В ИНТЕРАКТИВНОМ РЕЖИМЕ). ПРИ РАБОТЕ В ПАКЕТНОМ РЕЖИМЕ ТАКАЯ ПЕРЕМЕННАЯ АВТОМАТИЧЕСКИ ОПИСЫВАЕТСЯ КАК ВЕКТОРНАЯ, ПРИ ЭТОМ ПЕЧАТАЕТСЯ СООТВЕТСТВУЮЩЕЕ СООБЩЕНИЕ.

ПРИМЕРЫ:

ПРЕДПОЛОЖИМ, ЧТО P И Q БЫЛИ ОПИСАНЫ КАК ВЕКТОРЫ, ТОГДА СЛЕДУЮЩИЕ ВЫРАЖЕНИЯ ЯВЛЯЮТСЯ ВЕКТОРНЫМИ:

P

$P=2*Q/3$

$2*X*Y+P = P,Q*Q/(3*Q.Q)$

ТОГДА КАК P*Q И P/Q НЕ ЯВЛЯЮТСЯ ВЕКТОРНЫМИ.

2. ДИРАКОВСКИЕ ВЫРАЖЕНИЯ.

ТАК НАЗЫВАЮТСЯ ВЫРАЖЕНИЯ, СОДЕРЖАЩИЕ ГАММА-МАТРИЦЫ, ПОД ГАММА-МАТРИЦЕЙ ПОДРАЗУМЕВАЕТСЯ МАТРИЦА 4*4, ТАК ЧТО ПРОИЗВЕДЕНИЕ, СУММА И РАЗНОСТЬ ТАКИХ ВЫРАЖЕНИЙ, ИЛИ ПРОИЗВЕДЕНИЕ СКАЛЯРА И ДИРАКОВСКОГО ВЫРАЖЕНИЯ - СНОВА ДИРАКОВСКОЕ ВЫРАЖЕНИЕ. В СИСТЕМЕ НЕТ ДИРАКОВСКИХ ПЕРЕМЕННЫХ, ПОЭТОМУ, ЕСЛИ В ДИРАКОВСКОЕ ВЫРАЖЕНИЕ ВХОДИТ СКАЛЯРНАЯ ПЕРЕМЕННАЯ, ТО ПОДРАЗУМЕВАЕТСЯ, ЧТО ОНА УМНОЖЕНА НА ЕДИНИЧНУЮ МАТРИЦУ 4*4.

ПРИМЕР:

$G(L,P) + M$ ОБОЗНАЧАЕТ $G(L,P) + M * \langle$ ЕДИНИЧНУЮ МАТРИЦУ 4*4 \rangle .

УМНОЖАЮТСЯ ДИРАКОВСКИЕ ВЫРАЖЕНИЯ КАК МАТРИЧНЫЕ; ОНИ, КОНЕЧНО, НЕКОММУТАТИВНЫ.

6.5 ВЫЧИСЛЕНИЕ ШПУРА.

ПРИ ВЫЧИСЛЕНИИ ДИРАКОВСКОГО ВЫРАЖЕНИЯ СИСТЕМА АВТОМАТИЧЕСКИ ВЫЧИСЛЯЕТ 1/4 ШПУРА ОТ КАЖДОГО ПРОИЗВЕДЕНИЯ ГАММА-МАТРИЦ, 1/4 БЕРЕТСЯ ВО ИЗБЕЖАНИЕ НЕСООТВЕТСТВИЯ МЕЖДУ ШПУРОМ СКАЛЯРА, СКАЖЕМ, M , И M , ПРЕДСТАВЛЕННОМ В ВИДЕ $M * \langle$ ЕДИНИЧНУЮ МАТРИЦУ 4*4 \rangle .

ДЛЯ ВЫЧИСЛЕНИЯ ШПУРОВ ИСПОЛЬЗУЮТСЯ НАИБОЛЕЕ УДАЧНЫЕ ИЗ ИМЕЮЩИХСЯ К МОМЕНТУ СОЗДАНИЯ СИСТЕМЫ АЛГОРИТМЫ. НАПРИМЕР, НАРЯДУ С АЛГОРИТМОМ [8] ДЛЯ СВЕРТКИ ПО ИНДЕКСАМ В ПРОИЗВЕДЕНИИ ШПУРОВ, REDUCE ИСПОЛЬЗУЕТ ЭЛЕГАНТНЫЙ АЛГОРИТМ [9] ДЛЯ СВЕРТКИ ПО ИНДЕКСАМ В ПРОИЗВЕДЕНИЯХ ГАММА-МАТРИЦ.

ЕСЛИ ТРЕБУЕТСЯ "ВЫКЛЮЧИТЬ" ВЫЧИСЛЕНИЕ ШПУРОВ НА КАКОМ-ТО ФЕРМИОННОЙ ЛИНИИ, СЛЕДУЕТ ВОСПОЛЬЗОВАТЬСЯ ОПИСАНИЕМ **WOSPUR**,

ПРИМЕР:

WOSPUR L1,L2

ЭТО ОЗНАЧАЕТ, ЧТО НЕ БЕРУТСЯ ШПУРЫ ГАММА-МАТРИЧНЫХ ЧЛЕНОВ НА ЛИНИЯХ L1, L2. "ВКЛЮЧИТЬ" В ДАЛЬНЕЙШЕМ ВЫЧИСЛЕНИЕ ШПУРОВ НА ЭТИХ ЛИНИЯХ МОЖНО ПОСРЕДСТВОМ ОПИСАНИЯ **SPUR**.

6.6 ОПИСАНИЕ MASS.

В ВЫЧИСЛЕНИЯХ ЧАСТО НЕОБХОДИМО ПОМЕСТИТЬ ЧАСТИЦУ НА "МАССОВУЮ ПОВЕРХНОСТЬ", ТО ЕСТЬ СВЯЗАТЬ 4-ВЕКТОР С СООТВЕТСТВУЮЩЕЙ СКАЛЯРНОЙ ПЕРЕМЕННОЙ, ОБОЗНАЧАЮЩЕЙ МАССУ. ЭТО, КОНЕЧНО, МОЖНО ВЫПОЛНИТЬ ПО КОМАНДЕ **LET**, НАПРИМЕР,

LET P,P=M2**

НО СУЩЕСТВУЕТ ДРУГОЙ МЕТОД С ИСПОЛЬЗОВАНИЕМ КОМАНД **MASS** И **MSHELL**.

MASS ОБРАБАТЫВАЕТ СПИСОК РАВЕНСТВ В ФОРМЕ: $A = B$

<ВЕКТОРНАЯ ПЕРЕМЕННАЯ> = <СКАЛЯРНАЯ ПЕРЕМЕННАЯ>

ПРИМЕР:

MASS P1=M, Q1=MU1

ЕДИНСТВЕННЫЙ РЕЗУЛЬТАТ ЭТОЙ КОМАНДЫ - СКАЛЯРНАЯ ПЕРЕМЕННАЯ (MASS-CA) БУДЕТ СВЯЗАНА С СООТВЕТСТВУЮЩИМ 4-ВЕКТОРОМ.

ЕСЛИ ЗАПИСАТЬ $\vec{p} + (0, 1, 0, 0)$ И $\vec{p} + (0, 1, 0, 0)$

MSHELL <ВЕКТОРНАЯ ПЕРЕМЕННАЯ>,...,<ВЕКТОРНАЯ ПЕРЕМЕННАЯ>

И ЕСЛИ МАССА УЖЕ БЫЛА СВЯЗАНА С ЭТИМИ АРГУМЕНТАМИ, ТО БУДЕТ ВЫПОЛНЕНА ПОДСТАНОВКА ВИДА:

<ВЕКТОРНАЯ ПЕРЕМЕННАЯ>.<ВЕКТОРНАЯ ПЕРЕМЕННАЯ> = <MASS>

ДАЛЕЕ ПРИВОДИТСЯ СООТВЕТСТВУЮЩИЙ ПРИМЕР.

6.7 ПРИМЕР.

В КАЧЕСТВЕ ПРОСТОГО ПРИМЕРА ВЫЧИСЛЕНИЯ ДЛЯ ЗАДАЧИ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ МЫ ДАЕМ РАСЧЕТ СЕЧЕНИЯ КОМПТОНОВСКОГО РАССЕРЯНИЯ (БЬЕРКЕН И ДРЕЙЛ [7], (7.72)-(7.74)).

МЫ ХОТИМ ПОСЧИТАТЬ

$$\text{ALPHA} = \frac{1}{2} \left(\frac{K''}{K} \right) \text{TRACE} \left(\frac{E'' E K_I}{2M} + \frac{E'' E K_F}{2K'' \cdot P_I} \right) \frac{P_I + M}{2M} \frac{K_I E E''}{2K \cdot P_I} + \frac{K_F E E''}{2K'' \cdot P_I}$$

ГДЕ K_I И K_F 4-ИМПУЛЬСЫ ВХОДЯЩИХ И ВЫХОДЯЩИХ ФОТОНОВ (С ВЕКТОРАМИ ПОЛЯРИЗАЦИИ E И E'' И ЭНЕРГИЯМИ K И K'' В ЛАБОРАТОРНОЙ СИСТЕМЕ КООРДИНАТ, СООТВЕТСТВЕННО) И P_I И P_F - НАЧАЛЬНЫЕ И КОНЕЧНЫЕ 4-ИМПУЛЬСЫ ЭЛЕКТРОНА.

НЕПОДЧЕРКНУТЫЙ 4-ВЕКТОР В НАШЕЙ ЗАПИСИ ОБОЗНАЧАЕТ СВЕРТКУ ЭТОГО 4-ВЕКТОРА С ГАММА-МАТРИЦАМИ. К ПРИМЕРУ, $P_F = \text{GAMMA} \cdot P_F$.

ЕСЛИ ОПУСТИТЬ МНОЖИТЕЛЬ

$$\text{ALFA} = 2 / (2 + M^2) * (K'' / K)^2$$

ТО НУЖНО НАЙТИ

$$\frac{1}{4} \text{TRACE} \left(\frac{P_F + M}{2K \cdot P_I} \left(\frac{E'' E K_I}{2K'' \cdot P_I} + \frac{E'' E K_F}{2K'' \cdot P_I} \right) \frac{P_I + M}{2K \cdot P_I} + \frac{K_I E E''}{2K \cdot P_I} + \frac{K_F E E''}{2K'' \cdot P_I} \right)$$

СООТВЕТСТВУЮЩАЯ REDUCE - ПРОГРАММА:

```

ON DIV; COMMENT THIS GIVES US OUTPUT IN SAME FORM AS
      BJORKEN AND DRELL;
MASS KI=0, KF=0, PI=M, PF=M; VECTOR E;
COMMENT IF E IS USED AS A VECTOR, IT LOSES ITS SCALAR
IDENTITY AS THE BASE OF NATURAL LOGARITHMS;
MSHELL KI, KF, PI, PF;
LET PI, E=0, PI, EP=0, PI, PF=M**2+KI, KF, PI, KI=M*K, PI, KF=
-M**K, PF, E=-KF, E, PF, EP=KI, EP, PF, KI=M**K, PF, KF=
M**K, KI, E=0, KI, KF=M*(K-KP), KP, EP=0, E, E=-1,
EP, EP=-1;
FOR ALL P LET GP(P)=G(L,P)+M;
COMMENT THIS IS JUST TO SAVE US A LOT OF WRITING;
GP(PF)+(G(L,EP,E,KI)/(2+KI,PI) + G(L,E,EP,KF)/(2+KF,PI))
+ GP(PI)+(G(L,KI,E,EP)/(2+KI,PI) + G(L,KF,EP,E)/
(2+KF,PI))#
WRITE "THE COMPTON CROSS-SECTION IS",I+ANS;

```

РЕЗУЛЬТАТ РАБОТЫ ЭТОЙ ПРОГРАММЫ :

```

THE COMPTON CXN IS 1/2*K**KP + 1/2*K**KP + 2+E,EP = 1

```

6.8 КОМАНДА VECDIM,

РЕКТОРНАЯ АЛГЕБРА И АЛГЕБРА ГАММА-МАТРИЦ ДИРАКА В СИСТЕМЕ
 БЫЛИ ОБОБЩЕНЫ НА ПРОИЗВОЛЬНОЕ ЧИСЛО ИЗМЕРЕНИЙ. КОМАНДА
VECDIM <РАЗМЕРНОСТЬ>;

УСТАНОВЛИВАЕТ ТРЕБУЕМУЮ РАЗМЕРНОСТЬ, КОТОРАЯ МОЖЕТ ВЫРАЖАТЬСЯ
КАК ЧИСЛЕННО, ТАК И СИМВОЛЬНО.

ПРЕДВАРИТЕЛЬНЫЕ ЗАМЕЧАНИЯ.

7.1 ПРЕДВАРИТЕЛЬНЫЕ ЗАМЕЧАНИЯ.

ХОТЯ СИСТЕМА REDUCE БЫЛА КОНСТРУИРОВАНА ДЛЯ АЛГЕБРАИЧЕСКИХ ВЫЧИСЛЕНИЙ, ОНА ПОЗВОЛЯЕТ ИСПОЛЬЗОВАТЬ ПОЛНЫЙ НАБОР СИМВОЛЬНЫХ КОМАНД ЯЗЫКА LISP (СВОИМИ ЯЗЫКОВЫМИ СРЕДСТВАМИ). ЮДНАКО, ЧТОБЫ ВОСПОЛЬЗОВАТЬСЯ ЭТИМ, НУЖНО ОБЕСПЕЧИТЬ РАБОТУ СИСТЕМЫ В ДВУХ РЕЖИМАХ: СИМВОЛЬНОМ И АЛГЕБРАИЧЕСКОМ. ДЛЯ ПЕРЕХОДА В СИМВОЛЬНЫЙ РЕЖИМ НУЖНО НАБРАТЬ КОМАНДУ

```
SYMBOLIC) (ИЛИ (LISP))
```

ДЛЯ ВОЗВРАТА К АЛГЕБРАИЧЕСКОМУ - КОМАНДУ

```
ALGEBRAIC)
```

ВЫЧИСЛЕНИЯ ОСУЩЕСТВЛЯЮТСЯ ПО-РАЗНОМУ В КАЖДОМ РЕЖИМЕ, ПОЭТОМУ ПРОГРАММИСТУ РЕКОМЕНДУЕТСЯ ПРОВЕРИТЬ, В КАКОМ ИМЕННО РЕЖИМЕ ОН РАБОТАЕТ (ЕСЛИ ПОЯВЛЯЕТСЯ ОШИБКА, ВЫЗЫВАЮЩАЯ ЗАТРУДНЕНИЯ), ЭТО МОЖНО СДЕЛАТЬ, НАБРАВ КОМАНДУ

```
(MODE)
```

В РЕЗУЛЬТАТЕ БУДЕТ НАПЕЧАТАН ТЕКУЩИЙ РЕЖИМ: SYMBOLIC ИЛИ ALGEBRAIC.

ЕСЛИ НЕОБХОДИМО СМЕНИТЬ РЕЖИМ НА ОГРАНИЧЕННОЕ ВРЕМЯ, ТО ЭТО ЛЕГКО СДЕЛАТЬ, НАБРАВ

```
SYMBOLIC <КОМАНДА>;
(ИЛИ
LISP <КОМАНДА>;
ИЛИ ЖЕ
ALGEBRAIC <КОМАНДА>;
```

ПОСЛЕ ОКОНЧАНИЯ ЭТОЙ КОМАНДЫ ВЫ АВТОМАТИЧЕСКИ ВЕРНЕТЕСЬ В ИСХОДНЫЙ РЕЖИМ.

ПРИМЕР:

ЕСЛИ ТЕКУЩИЙ РЕЖИМ ALGEBRAIC, ТО КОМАНДЫ

```
SYMBOLIC CAR T (A);
X+Y;
```

БУДУТ РАБОТАТЬ КАК СИМВОЛЬНАЯ И АЛГЕБРАИЧЕСКАЯ, СООТВЕТСТВЕННО.

ЭТОТ РАЗДЕЛ ПРЕДПОЛАГАЕТ, ЧТО ЧИТАТЕЛЬ ИМЕЕТ ПРЕДСТАВЛЕНИЕ О ЯЗЫКЕ ПРОГРАММИРОВАНИЯ LISP 1.5 НА УРОВНЕ [4] ИЛИ [5]. ЧЕЛОВЕКУ, НЕЗНАКОМОМУ С ЭТИМ МАТЕРИАЛОМ, БУДЕТ ТРУДНО ПОЧАТЬ ДАННЫЙ РАЗДЕЛ.

КРОМЕ СЛУЧАЕВ, ГДЕ БЫЛИ СДЕЛАНЫ ЯВНЫЕ ОГРАНИЧЕНИЯ, БОЛЬШИНСТВО АЛГЕБРАИЧЕСКИХ КОНСТРУКЦИЙ ЯЗЫКА REDUCE ПЕРЕНОСЯТСЯ В СИМВОЛИЧЕСКУЮ МОДУ. ОДНАКО, ЗДЕСЬ ЕСТЬ НЕКОТОРЫЕ РАЗЛИЧИЯ.

ВО-ПЕРВЫХ, ВЫЧИСЛЕНИЕ ВЫРАЖЕНИЯ ТЕПЕРЬ СТАНОВИТСЯ LISP-ВЫЧИСЛЕНИЕМ.

ВО-ВТОРЫХ, ОПЕРАТОРЫ ПРИСВАИВАНИЯ ОБРАБАТЫВАЮТСЯ ИНАЧЕ, ЧТО ОБСУЖДАЕТСЯ В П.6.7 .

В-ТРЕТЬИХ, ЛОКАЛЬНЫМ ПЕРЕМЕННЫМ И ЭЛЕМЕНТАМ МАССИВА ПЕРВОНАЧАЛЬНО ПРИСВАИВАЕТСЯ ЗНАЧЕНИЕ NIL, А НЕ 0. (ФАКТИЧЕСКИ, ТАКИЕ ПЕРЕМЕННЫЕ И ЭЛЕМЕНТЫ ПРИНИМАЮТ ЗНАЧЕНИЕ NIL ФАКТИЧЕСКИ АЛГЕБРАИЧЕСКОМ РЕЖИМЕ, НО АЛГЕБРАИЧЕСКИЙ ВЫЧИСЛИТЕЛЬ РАСПОЗНАЕТ NIL КАК 0).

В-ЧЕТВЕРТЫХ, ОПЕРАТОРЫ SUM И PRODUCT (П.2.11.3,) НЕ ОПРЕДЕЛЕНЬ В СИМВОЛЬНОМ РЕЖИМЕ. НАКОНЕЦ, ОПРЕДЕЛЕНИЯ ФУНКЦИИ СООТВЕТСТВУЮТ [6].

ДЛЯ НАЧАЛА РАССМОТРИМ НЕКОТОРОЕ ОБОБЩЕНИЕ ПОНЯТИЯ БАЗИСНОГО СИНТАКСИСА, КОТОРЫЕ БЫЛИ ВВЕДЕНЬ РАНЕЕ, ЕСЛИ ТОЛЬКО ОНИ ВООБЩЕ ДОПУСТИМЫ В СИМВОЛЬНОМ РЕЖИМЕ.

7.2 ОБОБЩЕНИЕ ИДЕНТИФИКАТОРОВ.

В П. 2.1.2 МЫ ОПРЕДЕЛИЛИ ИДЕНТИФИКАТОР КАК ПОСЛЕДОВАТЕЛЬНОСТЬ БУКВ И ЦИФР. ОДНАКО В СИСТЕМЕ REDUCE ИМЕЕТСЯ СПЕЦИАЛЬНЫЙ "ЗНАК ПЕРЕКЛЮЧЕНИЯ ЛЕКСИЧЕСКОГО КЛАССА", С ПОМОЩЬЮ КОТОРОГО В ПОСЛЕДОВАТЕЛЬНОСТЬ ЗНАКОВ, СОСТАВЛЯЮЩИХ ИДЕНТИФИКАТОР, МОЖНО ВВЕСТИ ЛЮБОЙ НЕБУКВЕННО-ЦИФРОВОЙ СИМВОЛ. "ЗНАК ПЕРЕКЛЮЧЕНИЯ" КАЖДЫЙ РАЗ ДОЛЖЕН ПРЕДВАРИТЬ ВВОДИМЫЙ СИМВОЛ. ("ЗНАК ПЕРЕКЛЮЧЕНИЯ" МЕНЯЕТСЯ В ЗАВИСИМОСТИ ОТ РЕАЛИЗАЦИИ СИСТЕМЫ - ЭТО МОЖЕТ БЫТЬ ВОСКЛИЦАТЕЛЬНЫЙ ЗНАК, ИЛИ ПРАВАЯ ЗАКРЫВАЮЩАЯ КВАДРАТНАЯ СКОБКА, ИЛИ ЗНАК ПОДЧЕРКИВАНИЯ).

ПРИМЕР:

A!(SVI;

- ЭТО ДОПУСТИМЫЙ ИДЕНТИФИКАТОР, ОН БУДЕТ ПЕЧАТАТЬСЯ КАК

A(SV;

7.3 СИМВОЛЬНЫЕ ИНФИКСНЫЕ ОПЕРАЦИИ.

В ЯЗЫКЕ REDUCE СУЩЕСТВУЮТ ДВЕ БИНАРНЫЕ ИНФИКСНЫЕ ОПЕРАЦИИ, ПРЕДНАЗНАЧЕННЫЕ ДЛЯ ИСПОЛЬЗОВАНИЯ ПРИ СИМВОЛЬНОМ РЕЖИМЕ. ЭТО EQ И CONS. СТАРШИНСТВО ЭТИХ ОПЕРАЦИЙ ПО ОТНОШЕНИЮ К ДРУГИМ ОПЕРАЦИЯМ УКАЗАНО В П.2.6.

7.4 СИМВОЛЬНЫЕ ВЫРАЖЕНИЯ.

ОНИ СОДЕРЖАТ СКАЛЯРНЫЕ ПЕРЕМЕННЫЕ И СИМВОЛЫ ОПЕРАЦИЙ И СЛЕДУЮТ ОБЫЧНЫМ ПРАВИЛАМ МЕТАЯЗЫКА LISP.

ПРИМЕРЫ:

X
CAR U, REVERSE V
SIMP (U+V**2)

((S (U V)) (X Y))

7.5 QUOTE - ВЫРАЖЕНИЯ.

ТАК КАК LISP ТРЕБУЕТ, ЧТОБЫ КАЖДАЯ ПЕРЕМЕННАЯ ИЛИ ВЫРАЖЕНИЕ ИМЕЛИ ЗНАЧЕНИЕ, ТО В ЯЗЫК REDUCE НЕОБХОДИМО ВВЕСТИ QUOTE -ВЫРАЖЕНИЯ, ПО АНАЛОГИИ С QUOTE -ФУНКЦИЕЙ ЯЗЫКА LISP. ВВЕДЕМ НОВЫЙ СИМВОЛ - АПОСТРОФ.

ПРИМЕР:

!A

-СООТВЕТСТВУЕТ ЛИСПОВСКОМУ ВЫРАЖЕНИЮ

(QUOTE A)

!(A B C)

-СООТВЕТСТВУЕТ ЛИСПОВСКОМУ ВЫРАЖЕНИЮ

(QUOTE (A B C))

ЗАМЕТЬ, ОДНАКО, ЧТО ПРИ СИМВОЛЬНОЙ ОБРАБОТКЕ СТРОКИ АВТОМАТИЧЕСКИ ЗАКЛЮЧАЮТСЯ В КАВЫЧКИ, ТАК, ЧТОБЫ НАПЕЧАТАТЬ СТРОКУ

"A STRING"

НУЖНО НАПИСАТЬ:

PRINT "A STRING")

ВНУТРИ QUOTE -ВЫРАЖЕНИЙ СИНТАКСИЧЕСКИЕ ПРАВИЛА ДЛЯ ИДЕНТИФИКАТОРОВ ТЕ ЖЕ, ЧТО И В ЯЗЫКЕ REDUCE.

ТАК, ЗАПИСЬ

(A . B)

БУДЕТ РАССМАТРИВАТЬСЯ КАК СПИСОК ИЗ ТРЕХ ЭЛЕМЕНТОВ :

A

.

B

ТОГДА КАК

(A.B)

- ПРОСТЕЙШАЯ ТОЧЕЧНАЯ ПАРА.

7.6 LAMBDA - ВЫРАЖЕНИЯ.

LAMBDA - ВЫРАЖЕНИЯ ОБЕСПЕЧИВАЮТ СРЕДСТВА ДЛЯ КОНСТРУИРОВАНИЯ ЛИСПОВСКИХ LAMBDA - ВЫРАЖЕНИЯ В СИМВОЛЬНОМ РЕЖИМЕ. В АЛГЕБРАИЧЕСКОМ РЕЖИМЕ ИХ НЕЛЬЗЯ ИСПОЛЬЗОВАТЬ. СИНТАКСИС:

<LAMBDA ВЫРАЖЕНИЕ> ::= LAMBDA <СПИСОК ПЕРЕМЕННЫХ><ОГРАНИЧИТЕЛЬ> <ОПЕРАТОР>

<СПИСОК ПЕРЕМЕННЫХ> ::= (<ПЕРЕМЕННАЯ>...<ПЕРЕМЕННАЯ>)

ПРИМЕР:

LAMBDA (X,Y) CAR X . CDR Y

ЭТА ЗАПИСЬ ЭКВИВАЛЕНТНА LAMBDA -ВЫРАЖЕНИЮ

(LAMBDA (X Y) (CONS (CAR X) (CDR Y)))

СКОБКИ В <СПИСКЕ ПЕРЕМЕННЫХ> МОГУТ БЫТЬ ОПУЩЕНЫ, ЕСЛИ ЭТО НЕОБХОДИМО. LAMBDA - ВЫРАЖЕНИЯ МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ В СИМВОЛЬНОМ РЕЖИМЕ ВМЕСТО ПРЕФИКСНЫХ ОПЕРАЦИЙ ИЛИ В КАЧЕСТВЕ АРГУМЕНТОВ ЗАРЕЗЕРВИРОВАННОГО СЛОВА FUNCTION.

7.7 СИМВОЛЬНЫЕ ОПЕРАТОРЫ ПРИСВАИВАНИЯ.

ЕСЛИ ЛЕВАЯ ЧАСТЬ ОПЕРАТОРА ПРИСВАИВАНИЯ (В СИМВОЛЬНОМ РЕЖИМЕ) ЕСТЬ ПЕРЕМЕННАЯ, ТО ПРАВАЯ ЧАСТЬ ПРИСВАИВАЕТСЯ ЕЙ ОПЕРАТОРОМ **SETQ**,

ПРИМЕР:

X := Y

ПРЕОБРАЗУЕТСЯ В

(SETQ X Y),

ТОГДА КАК

```
DEFUN ASSOC (U,V) := IF NULL V THEN NIL
                    ELSE IF U EQ CAAR V THEN CAR V
                    ELSE ASSOC (U,CDR V)
```

ОПРЕДЕЛЯЕТ ЛИСПОВСКУЮ ФУНКЦИЮ **ASSOC**,

ФУНКЦИИ **MACRO** И **FEEXPR** МОЖНО ОПРЕДЕЛИТЬ, ПРЕДВАРИЯ ОПЕРАТОР ПРИСВАИВАНИЯ СЛОВИМИ **MACRO** И **FEEXPR**, СООТВЕТСТВЕННО.

ПРИМЕР:

MACRO CONSONS

МОЖНО ОПРЕДЕЛИТЬ ПОСРЕДСТВОМ

MACRO CONSONS L := EXPAND(CDR L, CONS)

ФУНКЦИОНАЛЬНЫЕ ОПРЕДЕЛЕНИЯ МОГУТ БЫТЬ ВВЕДЕНЫ ТАКЖЕ КАК ПРОЦЕДУРЫ (П.2.18,). ТИП ПРОЦЕДУРЫ В ЭТОМ СЛУЧАЕ **SYMBOLIC** (ИЛИ **LISP**), К ПРИМЕРУ, ПРЕДЫДУЩЕЕ ОПРЕДЕЛЕНИЕ **ASSOC** МОЖЕТ БЫТЬ ЗАПИСАНО В ВИДЕ:

```
SYMBOLIC PROCEDURE ASSOC (U,V)
  IF NULL V THEN NIL
  ELSE IF V EQ CAAR V THEN CAR V
  ELSE ASSOC (U,CDR V)
```

FEEXPR И **MACRO** МОГУТ БЫТЬ ТАКЖЕ ВВЕДЕНЫ ТАКИМ СПОСОБОМ С ПРОЦЕДУРНЫМ ТИПОМ **FEEXPR** И **MACRO**, СООТВЕТСТВЕННО.

7.8 СВЯЗЬ С АЛГЕБРАИЧЕСКИМ РЕЖИМОМ.

ЕСЛИ В СИМВОЛЬНОМ РЕЖИМЕ ФУНКЦИЯ ОПРЕДЕЛЕНА ДЛЯ ИСПОЛЬЗОВАНИЯ В КАЧЕСТВЕ ОПЕРАЦИИ В АЛГЕБРАИЧЕСКИХ ВЫЧИСЛЕНИЯХ, ТО НЕОБХОДИМО СВЯЗАТЬ ЕЕ С АЛГЕБРАИЧЕСКИМ ПРОЦЕССОРОМ. ЭТО МОЖНО СДЕЛАТЬ С ПОМОЩЬЮ КОМАНДЫ **OPERATOR**, ТАК,

OPERATOR FUN1, FUN2;

ОБЪЯВИТ ФУНКЦИИ **FUN1** И **FUN2** АЛГЕБРАИЧЕСКИМИ ОПЕРАЦИЯМИ, ЭТО ОПИСАНИЕ ДОЛЖНО БЫТЬ СДЕЛАНО В СИМВОЛЬНОМ РЕЖИМЕ, Т.К. В АЛГЕБРАИЧЕСКОМ РЕЖИМЕ ОНО ДЕЙСТВУЕТ ПО-ДРУГОМУ.

ДАЛЕЕ, ЕСЛИ ВЫ ХОТИТЕ ПРОИЗВЕСТИ АЛГЕБРАИЧЕСКИЕ ДЕЙСТВИЯ НАД СИМВОЛЬНЫМ АРГУМЕНТОМ, МОЖНО ВОСПОЛЬЗОВАТЬСЯ ФУНКЦИЕЙ **REVAL**. ЕДИНСТВЕННЫЙ АРГУМЕНТ ФУНКЦИИ **REVAL** ДОЛЖЕН БЫТЬ ЛИСПОВСКИМ ПРЕФИКСНЫМ ЭКВИВАЛЕНТОМ СКАЛЯРНОГО ВЫРАЖЕНИЯ, НАПРИМЕР, **(COS(PLUS X Y))**. **REVAL** ВОЗВРАЩАЕТ ВЫЧИСЛЕННОЕ ВЫРАЖЕНИЕ В

8.1 ЗАРЕЗЕРВИРОВАННЫЕ ИДЕНТИФИКАТОРЫ.

Мы даем здесь список всех зарезервированных идентификаторов. В него включены зарезервированные идентификаторы, приведенные в главе 2 и все имена встроенных команд и операций системы, здесь же приводятся зарезервированные идентификаторы, используемые в вычислениях, проводимых при решении задач физики высоких энергий.

ЗАРЕЗЕРВИРОВАННЫЕ СЛОВА:

BEGIN DO ELSE END FOR FUNCTION GO GOTO
LAMBDA NIL PRODUCT RETURN STEP SUM TO WHILE

ЗАРЕЗЕРВИРОВАННЫЕ СКАЛЯРНЫЕ ПЕРЕМЕННЫЕ:

E I

ИНФИКСНЫЕ ОПЕРАЦИИ:

SETQ AND NOT OR MEMBER EQUAL UNEQ EQ GEO
GREATERP LEQ LESSP PLUS MINUS TIMES
QUOTIENT EXPT CONS

ПРЕФИКСНЫЕ ОПЕРАЦИИ:

COEFF COS DEN DET DF EPS G LOG MAT NUM SIN
SUB TRACE

КОМАНДЫ:

ALGEBRAIC ARRAY CLEAR COMMENT DOWN END FACTOR FOR
FORALL GO GOTO IF IN INTEGER LET LISP MASS
MATCH MATRIX MSHLL NOSPUR OFF ON OPERATOR
ORDER OUT PROCEDURE REAL RETURN SAVEAS SCALAR
SHUT SPUR SYMBOLIC UP VECTOR WEIGHT WRITE
WTLEVEL PAUSE

ЗНАКИ

<< >>

8.2 СТАНДАРТНЫЙ НАБОР КОМАНД ЯЗЫКА REDUCE.

E, E_1, \dots, E_N - ОБОЗНАЧАЮТ ВЫРАЖЕНИЯ. V, V_1, \dots, V_N - ОБОЗНАЧАЮТ ПЕРЕМЕННЫЕ.

ALGEBRAIC (E)

ЕСЛИ E ОТСУТСТВУЕТ, ТО ВКЛЮЧАЕТСЯ АЛГЕБРАИЧЕСКИЙ РЕЖИМ. В ПРОТИВНОМ СЛУЧАЕ E ВЫЧИСЛЯЕТСЯ АЛГЕБРАИЧЕСКИ, А РЕЖИМ РАБОТЫ СИСТЕМЫ НЕ МЕНЯЕТСЯ (П.7.1)

ARRAY V1 <РАЗМЕР>, ..., Vn <РАЗМЕР>;
ОПИСЫВАЕТ V1, ..., Vn КАК ИМЕНА МАССИВОВ.
<РАЗМЕР> УКАЗЫВАЕТ МАКСИМАЛЬНЫЙ РАЗМЕР МАССИВА (П.2.4)

CLEAR E1, ..., EN)

УНИЧТОЖАЕТ В СИСТЕМЕ ВСЕ ПОДСТАНОВКИ ДЛЯ E1, ..., EN (П. 4.4.2)

COMMENT <ПРОИЗВОЛЬНАЯ ЗАПИСЬ>;
ИСПОЛЬЗУЕТСЯ ДЛЯ ВКЛЮЧЕНИЯ КОММЕНТАРИЕВ В ТЕКСТ. <ПРОИЗВОЛЬНАЯ ЗАПИСЬ> - ПРОИЗВОЛЬНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ, НЕ ВКЛЮЧАЮЩАЯ ОГРАНИЧИТЕЛЯ (П. 2.8, У

CONT)

ИНТЕРАКТИВНАЯ КОМАНДА, КОТОРАЯ ПРИКАЗЫВАЕТ СИСТЕМЕ ПРОДОЛЖАТЬ ВЫЧИСЛЕНИЯ С ТОГО МЕСТА В ФАЙЛЕ ВЫВОДА, ГДЕ ПОСЛЕДНИЙ РАЗ ВСТРЕЧАЛОСЬ PAUSE (П. 4.11)

DOWN E1, ..., EN)

ПРИКАЗЫВАЕТ СИСТЕМЕ ПОМЕСТИТЬ E1, ..., EN В ЗНАМЕНАТЕЛЬ (П. 4.8.3).

END <ПРОИЗВОЛЬНАЯ ЗАПИСЬ>;
ЗАКАНЧИВАЕТ (ОГРАНИЧИВАЕТ) ФАЙЛ ВЫВОДА. <ПРОИЗВОЛЬНАЯ ЗАПИСЬ> - ПРОИЗВОЛЬНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ, НЕ ВКЛЮЧАЮЩАЯ ОГРАНИЧИТЕЛЕЙ (ИЛИ ЗАРЕЗЕРВИРОВАННЫХ СИМВОЛ) END, ELSE, UNTIL (П. 2.1.3)

FACTOR E1, ..., EN)

ОПИСЫВАЕТ ВЫРАЖЕНИЯ КАК МНОЖИТЕЛИ НА ВЫВОДЕ (П. 4.8.2)

FOR

КОМАНДА, ИСПОЛЬЗУЕМАЯ ДЛЯ ОПРЕДЕЛЕНИЯ РАЗЛИЧНЫХ ПРОГРАММНЫХ ЦИКЛОВ (П. 3.5.2)

FORALL V1, ..., VN <КОМАНДА>

ОПИСЫВАЕТ V1, ..., VN КАК ФОРМАЛЬНЫЕ ПАРАМЕТРЫ В ПОДСТАНОВОЧНЫХ ПРАВИЛАХ, ЗАДАННЫХ В <КОМАНДЕ> (П. 4.1).

GOTO V)

ОСУЩЕСТВЛЯЕТ БЕЗУСЛОВНЫЙ ПЕРЕХОД ПО МЕТКЕ V. МОЖЕТ ИСПОЛЬЗОВАТЬСЯ ТОЛЬКО В ГРУППОВЫХ ОПЕРАТОРАХ И БЛОКАХ (П. 3.8.3).

IF

ИСПОЛЬЗУЕТСЯ ДЛЯ ОПРЕДЕЛЕНИЯ УСЛОВНЫХ ОПЕРАТОРОВ (П. 3.4).

IN V1, ..., VN)

ВВОДИТ ВНЕШНИЕ ФАЙЛЫ СИСТЕМЫ REDUCE ОТ V1 ДО VN ВКЛЮЧИТЕЛЬНО (П. 4.5.1)

INTEGER V1, ..., VN)

ОБЪЯВЛЯЕТ V1, ..., VN ЦЕЛЫМИ ПЕРЕМЕННЫМИ (П. 2.3.1)

LET E1, ..., EN)

ЗАДАЕТ ПОДСТАНОВКИ ДЛЯ ЛЕВЫХ ЧАСТЕЙ ВЫРАЖЕНИЙ E1, ..., EN (П. 4.1.1). КРОМЕ ТОГО,

LET

МОЖЕТ БЫТЬ ИСПОЛЬЗОВАНА ДЛЯ ВВЕДЕНИЯ ПРАВИЛ ДИФФЕРЕНЦИРОВАНИЯ (П. 4.2)

LISP E)

ЕСЛИ **E** ОТСУТСТВУЕТ, ТО СИСТЕМА ПЕРЕХОДИТ В СИМВОЛЬНЫЙ РЕЖИМ. В ПРОТИВНОМ СЛУЧАЕ **E** ОБРАБАТЫВАЕТСЯ СИМВОЛЬНО, А РЕЖИМ РАБОТЫ СИСТЕМЫ НЕ МЕНЯЕТСЯ (П. 7.1)

MATCH E1,...,EN;

ЗАДАЕТ ПОДСТАНОВКИ ДЛЯ ЛЕВЫХ ЧАСТЕЙ **E1,...,EN**, КОГДА ТРЕБУЕТСЯ СОГЛАСОВАНИЕ РАВНЫХ СТЕПЕНЕЙ (П. 4.1.3)

MATRIX E1,...,EN;

ОПИСЫВАЕТ МАТРИЧНЫЕ ПЕРЕМЕННЫЕ, **E1** МОЖЕТ БЫТЬ ИМЕНЕМ МАТРИЧНОЙ ПЕРЕМЕННОЙ, А МОЖЕТ И ВКЛЮЧАТЬ СВЕДЕНИЯ О РАЗМЕРЕ МАТРИЦЫ (П. 5.2)

OFF V1,...,VN;

СБРАСЫВАЕТ ФЛАГИ **V1,...,VN** (П. 4.4.1)

ON V1,...,VN)

УСТАНОВЛИВАЕТ ФЛАГИ **V1,...,VN** (П. 4.4.1)

OPERATOR V1,...,VN;

ОБЪЯВЛЯЕТ **V1,...,VN** АЛГЕБРАИЧЕСКИМИ ОПЕРАЦИЯМИ (П.П. 2.5.6, 7.8)

ORDER V1,...,VN;

УКАЗЫВАЕТ ПОРЯДОК ПЕРЕМЕННЫХ **V1,...,VN** НА ВЫВОДЕ (П. 4.8.1)

OUT V)

ОБЪЯВЛЯЕТ **V** ФАЙЛОМ ВЫВОДА (П. 4.5.3)

PAUSE)

ИНТЕРАКТИВНАЯ КОМАНДА, ИСПОЛЬЗУЕМАЯ В ФАЙЛЕ ВВОДА. КОГДА ОНА ВОСТРЕЧАЕТСЯ, УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ НА ТЕРМИНАЛ (П. 4.11)

PROCEDURE

ЗАДАЕТ ОПЕРАТОР ДЛЯ НЕОДНОКРАТНОГО ИСПОЛЬЗОВАНИЯ В ВЫЧИСЛЕНИЯХ. ТИП ПРОЦЕДУРЫ ПРЕДШЕСТВУЕТ ИМЕНИ КОМАНДЫ (П.П. 3.10, 7.7)

REAL V1,...,VN;

ОБЪЯВЛЯЕТ ПЕРЕМЕННЫЕ **V1,...,VN** ДЕЙСТВИТЕЛЬНЫМИ (П. 2.3.1)

RETURN E)

ОСУЩЕСТВЛЯЕТ ВЫХОД ИЗ БЛОКА НА БОЛЕЕ ВЫСОКИЙ ПРОГРАММНЫЙ УРОВЕНЬ. ЗНАЧЕНИЕ **E** ПЕРЕДАЕТСЯ ИЗ БЛОКА. **E** МОЖЕТ ОТСУТСТВОВАТЬ (П. 3.8.4)

SAVEAS E1 ПРИСВАИВАЕТ E ТЕКУЩЕЕ ЗНАЧЕНИЕ РАБОЧЕГО ПЕРЕМЕННОЙ (П. 4.3)

SCALAR V1, ..., VN ОБЪЯВЛЯЕТ ПЕРЕМЕННЫЕ V1, ..., VN СКАЛЯРНЫМИ (П. 2.3.1)

SHUT V1 ЗАКРЫВАЕТ ФАЙЛ ВЫВОДА V1 (П. 4.5.4)

SUMBOLIC E1 ТО ЖЕ САМОЕ, ЧТО И LISP E1

UP E1, ..., EN ПРИКАЗЫВАЕТ СИСТЕМЕ ПОМЕСТИТЬ E1, ..., EN В ЧИСЛИТЕЛЬ (П. 4.8.3).

WEIGHT E1, ..., EN ПРИСВАИВАЕТ АСИМПТОТИЧЕСКИЕ ВЕСА ЛЕВЫМ ЧАСТЯМ E1, ..., EN (П. 4.14)

WRITE E1, ..., EN УКАЗЫВАЕТ, ЧТО ЗНАЧЕНИЯ E1, ..., EN ДОЛЖНЫ БЫТЬ ЗАПИСАНЫ В ТЕКУЩИЙ ФАЙЛ ВЫВОДА (П. 3.9)

WTLEVEL V1 ПРИСВАИВАЕТ АСИМПТОТИЧЕСКИЙ ВЕСОМ УРОВЕНЬ СИСТЕМЫ ПЕРЕМЕННОЙ V1 (П. 4.14)

А.3 ФЛАГИ В СИСТЕМЕ REDUCE.

СТАНДАРТНОЕ ПОЛОЖЕНИЕ ФЛАГОВ :

ON ALLFAC, ECHO, EXP, MCD, MSG, NAT, PRI, RESUBS
OFF DEFN, DIV, FLOAT, FORT, GCD, LIST, NERO, RAT

ДАЛЕЕ ПРИВОДИТСЯ СПИСОК ИМЕН ФЛАГОВ, КОТОРЫЕ МОГУТ ПЯВЛЯТЬСЯ В ПРОГРАММЕ ТОЛЬКО КАК АРГУМЕНТЫ ОПИСАНИЯ **ON**, **OFF**. ДЕЙСТВИЕ УСТАНОВЛЕННЫХ ФЛАГОВ (СЛУЧАЙ **ON**), ОПИСЫВАЕТСЯ ДАЛЕЕ. В СЛУЧАЕ **OFF** ИХ ДЕЙСТВИЕ ПРОТИВОПОЛОЖНО.

ALLFAC ДАЕТ УКАЗАНИЕ СИСТЕМЕ ВЫНЕСТИ ОБЩИЕ МНОЖИТЕЛИ ПРИ ВЫВОДЕ ВЫРАЖЕНИЙ (П. 4.8.4)

DEFN ДАЕТ УКАЗАНИЕ СИСТЕМЕ СДЕЛАТЬ ТРАНСЛЯЦИЮ В БЕЗ ВЫЧИСЛЕНИЯ (П. 7.9)

DIV УКАЗЫВАЕТ СИСТЕМЕ, ЧТО НУЖНО ВЫВЕСТИ ПРОСТЫЕ

МНОЖИТЕЛИ (НА ВЫВОДЕ ПРИ ЭТОМ МОГУТ ПОЯВЛЯТЬСЯ
ОТРИЦАТЕЛЬНЫЕ СТЕПЕНИ И РАЦИОНАЛЬНЫЕ ДРОБИ) (П. 4.8.4.)

ESNO

ДАЕТ УКАЗАНИЕ ОТПЕЧАТАТЬ ФАЙЛ ВВОДА (НА ЛИСТИНГЕ
ИЛИ НА ЭКРАНЕ ТЕРМИНАЛА В ИНТЕРАКТИВНОМ РЕЖИМЕ)
(П. 4.5.1)

EXP

ПРИКАЗЫВАЕТ РАСКРЫТЬ ВЫРАЖЕНИЕ В ПРОЦЕССЕ
ВЫЧИСЛЕНИЯ (П. 4.6)

FLOAT

ПРЕДОТВРАЩАЕТ ПРЕОБРАЗОВАНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ
ТОЧКОЙ В ОТНОШЕНИЕ ДВУХ ЦЕЛЫХ ЧИСЕЛ В ПРОЦЕССЕ
ВЫЧИСЛЕНИЯ (П. 4.9)

FORT

ОСУЩЕСТВЛЯЕТ ВЫВОД В ФОРТРАН-ФОРМАТЕ (П. 4.9)

GCD

ПРИКАЗЫВАЕТ СИСТЕМЕ СОКРАТИТЬ НАИБОЛЬШИЙ ОБЩИЙ
ДЕЛИТЕЛЬ В РАЦИОНАЛЬНЫХ ВЫРАЖЕНИЯХ (П. 4.7)

INT

ИНИЦИИРУЕТ ИНТЕРАКТИВНЫЙ РЕЖИМ (П. 4.11)

LIST

УКАЗЫВАЕТ СИСТЕМЕ, ЧТО ПРИ ВЫВОДЕ КАЖДЫЙ ЧЛЕН
ВЫРАЖЕНИЯ СЛЕДУЕТ ПЕЧАТАТЬ НА ОТДЕЛЬНОЙ СТРОКЕ,

MCD

УКАЗЫВАЕТ СИСТЕМЕ, ЧТО СУММИРУЕМЫЕ ВЫРАЖЕНИЯ
СЛЕДУЕТ ПРИВЕСТИ К ОБЩЕМУ ЗНАМЕНАТЕЛЮ,
(П. 4.6)

MSG

ИНИЦИИРУЕТ ПЕЧАТЬ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

NAT

СПЕЦИФИЦИРУЕТ "ЕСТЕСТВЕННЫЙ" ВЫВОД (П. 4.5.4)

NERO

ПРЕДОТВРАЩАЕТ ПЕЧАТЬ НУЛЕВЫХ ЗНАЧЕНИЙ (П. 4.12)

PRI

СПЕЦИФИЦИРУЕТ ПРОИЗВОЛЬНЫЙ ФОРМАТ ВЫВОДА (4.8)

RAT

ФАГ ВЫВОДА, ИСПОЛЬЗУЕМЫЙ СОВМЕСТНО С FACTOR,
ОН УКАЗЫВАЕТ, ЧТО ЗНАМЕНАТЕЛЬ ВСЕГО ВЫРАЖЕНИЯ
ДОЛЖЕН БЫТЬ НАПЕЧАТАН С КАЖДЫМ
ФАКТОРИЗОВАННЫМ ПОДВЫРАЖЕНИЕМ (П. 3.2.3,)

RESUBS

КОГДА ЭТОТ ФЛАГ СБРОШЕН, (OFF),
СИСТЕМА НЕ ПЕРЕСМАТРИВАЕТ ВЫРАЖЕНИЕ НА ДАЛЬНЕЙШИЕ
ПОДСТАВКИ, ЕСЛИ ОНА УЖЕ СДЕЛАНА (П. 4.4.3)

8.3 ДИАГНОСТИКА И СООБЩЕНИЯ ОБ ОШИБКАХ В СИСТЕМЕ REDUCE.

В СИСТЕМЕ REDUCE СУЩЕСТВУЮТ ДИАГНОСТИЧЕСКИЕ СООБЩЕНИЯ ДВУХ ТИПОВ : СООБЩЕНИЯ ОБ ОШИБКЕ И ПРЕДУПРЕЖДАЮЩИЕ СООБЩЕНИЯ. ПЕРВЫЕ ОБЫЧНО ВЫЗЫВАЮТ ОСТАНОВКУ ПРОГРАММЫ. ТОГДА КАК ВТОРЫЕ ПРЕДУПРЕЖДАЮТ ПРОГРАММИСТА О ВСТРЕТИВШЕЙСЯ НЕОДНОЗНАЧНОСТИ ИЛИ НЕКОТОРОМ ДЕЙСТВИИ, КОТОРОЕ МОЖЕТ ПОВЛЕЧЬ ЗА СОБОЙ ОШИБКУ. ЕСЛИ СИСТЕМА РАБОТАЕТ В ИНТЕРАКТИВНОМ РЕЖИМЕ, ТО ОНА, КОГДА ВСТРЕТИТСЯ НЕОДНОЗНАЧНОСТЬ, МОЖЕТ ПОПРОСИТЬ КОРРЕКТИРОВКИ. В ПАКЕТНОМ РЕЖИМЕ В ТАКОЙ СИТУАЦИИ ОНА СДЕЛАЕТ НАИБОЛЕЕ ВЕРОЯТНУЮ ДОГАДКУ О НАМЕРЕНИИ ПРОГРАММИСТА, НАПЕЧАТАЕТ ИНФОРМИРУЮЩЕЕ СООБЩЕНИЕ И ПРОДОЛЖИТ РАБОТУ.

ЕСЛИ НА ВВОДЕ ОБНАРУЖИТСЯ СИНТАКСИЧЕСКАЯ ОШИБКА, ТО ТЕКУЩАЯ ФРАЗА ПЕРЕПЕЧАТЫВАЕТСЯ С СООБЩЕНИЕМ ОБ ОШИБКЕ. В НЕКОТОРЫХ СИСТЕМАХ В ТАКОЙ СИТУАЦИИ ВОЗМОЖНО ОТРЕДАКТИРОВАТЬ СТРОКУ И ИСПРАВИТЬ ОШИБКУ.

ДЛЯ ЗАКОНЧЕННОСТИ МЫ ВКЛЮЧАЕМ В ЭТОТ ПАРАГРАФ СООБЩЕНИЯ, КОТОРЫЕ МОГУТ ПОЯВИТЬСЯ ПРИ ПРОВЕДЕНИИ ВЫЧИСЛЕНИЯ В ЗАДАЧАХ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ .

1. СООБЩЕНИЯ ОБ ОШИБКАХ.

ЗАМЕЧАНИЕ.

СО ВРЕМЕНИ НАПИСАНИЯ [1] НАБОР СООБЩЕНИЙ ОБ ОШИБКАХ ВСЕ ВРЕМЯ ПОПОЛНЯЕТСЯ С ЦЕЛЬЮ ОБЕСПЕЧИТЬ ПРОГРАММИСТА НАИБОЛЕЕ ТОЧНОЙ ИНФОРМАЦИЕЙ О ДОПУЩЕННОЙ ОШИБКЕ. ПОЭТОМУ В ДАННОМ ПАРАГРАФЕ ПРИВОДЯТСЯ НЕ ВСЕ СООБЩЕНИЯ, ИМЕЮЩИЕСЯ В СИСТЕМЕ.

A REPRESENTS ONLY GAMMAS IN VECTOR EXPRESSIONS

A МОЖЕТ БЫТЬ ИСПОЛЬЗОВАНО ТОЛЬКО КАК GAMMAS
В ВЕКТОРНОМ ВЫРАЖЕНИИ.

ARRAY TOO SMALL

МАССИВ, ИСПОЛЬЗУЕМЫЙ В СОЕФФ,
СЛИШКОМ МАЛ ДЛЯ ХРАНЕНИЯ ВСЕХ СТЕПЕНЕЙ
ЕГО ПОДВЫРАЖЕНИЯ.

CATASTROPHIC ERROR

ЕСЛИ ПОЯВИТСЯ ТАКОЕ СООБЩЕНИЕ,
ПОЖАЛУЙСТА, ОТОШЛИТЕ ЛИСТИНГ С КОПИЕЙ
ВВОДА И ВЫВОДА АВТОРУ.

DIFFERENTIATION WRT <ВЫРАЖЕНИЕ> NOT ALLOWED

ЗАПРЕЩЕНО ДИФФЕРЕНЦИРОВАНИЕ ПО ЧЕМУ-
ЛИБО, КРОМЕ ПЕРЕМЕННОЙ.

DOT CONTEXT ERROR

QUOTE

-ВЫРАЖЕНИЕ, ОПРЕДЕЛЕННОЕ В СИМВОЛЬНОМ
РЕЖИМЕ, ИМЕЕТ НЕВЕРНУЮ С ТОЧКИ ЗРЕНИЯ

ЯЗЫКА LISP
СИНТАКСИС.

ELEMENT OUT OF BOUNDS

ДАНА ССЫЛКА НА ЭЛЕМЕНТ МАССИВА С
НОМЕРОМ, ВЫХОДЯЩИМ ЗА ПРЕДЕЛЫ
РАЗМЕРНОСТИ МАССИВА.

**INCORRECT ARRAY
ARGUMENTS FOR**
<ИМЯ>

НЕЧИСЛОВЫЙ ИНДЕКС БЫЛ ИСПОЛЬЗОВАН В
МАТРИЦЕ ИЛИ МАССИВЕ С УКАЗАННЫМ ИМЕНЕМ.

LARGER SYSTEM NEEDED

ЭТО ОЗНАЧАЕТ, ЧТО ВАША ЗАДАЧА УЧАСТКОВАТЬ
ТРЕБУЕТ ВОЗМОЖНОСТЕЙ СИСТЕМЫ,
НЕ ПРЕДУСМОТРЕННЫХ В ВАШЕЙ
РЕАЛИЗАЦИИ.

**LOCAL VARIABLE USED
AS OPERATOR**

БЫЛО ДОПУЩЕНО НЕЗАКОННОЕ
ИСПОЛЬЗОВАНИЕ ЛОКАЛЬНОЙ ПЕРЕМЕННОЙ
В КАЧЕСТВЕ СИМВОЛА ОПЕРАЦИИ.

MATRIX MISMATCH

ДВА МАТРИЧНЫХ ВЫРАЖЕНИЯ НЕКОРРЕКТНО
ИСПОЛЬЗУЮТСЯ ПРИ СЛОЖЕНИИ ИЛИ
УМНОЖЕНИИ.

MATRIX
<ИМЯ>

NOT SET

БЫЛА ССЫЛКА НА МАТРИЦУ, ЧЬИ ЭЛЕМЕНТЫ
ЭЛЕМЕНТЫ НЕИЗВЕСТНЫ.

MISMATCH OF ARGUMENTS

УКАЗЫВАЕТ, ЧТО ОПЕРАТОР, КОТОРЫЙ БЫЛ
ОПРЕДЕЛЕН КАК ПРОЦЕДУРА, ВЫЗЫВАЕТСЯ С
НЕВЕРНЫМ ЧИСЛОМ АРГУМЕНТОВ.

**MISSING ARGUMENTS FOR
G OPERATOR**

В ГАММА-МАТРИЧНОМ ВЫРАЖЕНИИ ПРОПУЩЕН
СИМВОЛ ЛИНИИ.

MISSING OPERATOR

ОШИБКА НА ВВОДЕ.

MISSING VECTOR

ВЫРАЖЕНИЕ, ВСТРЕЧАЮЩЕЕСЯ В
ВЕКТОРНОМ ВЫРАЖЕНИИ, НЕ СОДЕРЖИТ
ВЕКТОРА.

NON-NUMERICAL ARGUMENT IN
<ВЫРАЖЕНИЕ>

БЫЛА ССЫЛКА НА МАССИВ ИЛИ МАТРИЧНЫЙ
ЭЛЕМЕНТ С НЕЧИСЛОВЫМ ИНДЕКСОМ.

NON SQUARE MATRIX

ПОТРЕБОВАНО ОШИБОЧНОЕ ПРИМЕНЕНИЕ
 ОПЕРАЦИИ К НЕКВАДРАТНОЙ МАТРИЦЕ
 (НАПРИМЕР, ВЗЯТИЕ ШПУРА).

NOT YET IMPLEMENTED

ЕСЛИ ВЫ ПОЛУЧИЛИ ТАКОЕ СООБЩЕНИЕ, ТО,
 ПОЖАЛУЙСТА, ПОИЩИТЕ КОПИЮ ВАШЕГО
 ЛИСТИНГА АВТОРУ.

OPERATOR

<ИМЯ>

CANNOT BE ARBITRARY

ПРОИЗВОЛЬНАЯ ОПЕРАЦИЯ НЕ МОЖЕТ БЫТЬ
 ИСПОЛЬЗОВАНА В ПОДСТАНОВКЕ.

REDUNDANT OPERATOR

ОШИБКА НА ВВОДЕ

REDUNDANT VECTOR

В ВЕКТОРНОМ ВЫРАЖЕНИИ БЫЛ НАЙДЕН
 ОШИБОЧНЫЙ ВЕКТОР

SINGULAR MATRIX

ОТ СИСТЕМЫ ПОТРЕБОВАЛИ ОБРАЩЕНИЯ
 СИНГУЛЯРНОЙ МАТРИЦЫ.

SUBSTITUTION FOR

<ВЫРАЖЕНИЕ>

NOT ALLOWED

В ПРАВИЛЕ ПОДСТАНОВКИ ВСТРЕЧАЕТСЯ
 НЕРАЗРЕШЕННОЕ ПОДСТАНОВОЧНОЕ ВЫРАЖЕНИЕ.

SYNTAX ERROR

СИНТАКСИЧЕСКАЯ ОШИБКА НА ВВОДЕ. ЭТО
 СООБЩЕНИЕ ПОЯВЛЯЕТСЯ, ЕСЛИ СИСТЕМА НЕ
 В СОСТОЯНИИ ОПРЕДЕЛИТЬ, ЧТО ЯВЛЯЕТСЯ
 ПРИЧИНОЙ ОШИБКИ.

SYNTAX

<ВЫРАЖЕНИЕ>

INCORRECT

ПРИ ВЫЧИСЛЕНИИ <ВЫРАЖЕНИЯ> НАЙДЕНА
 СИНТАКСИЧЕСКАЯ ОШИБКА.

**TOO FEW RIGHT
 PARENTHESES**

ОШИБКА НА ВВОДЕ (СЛИШКОМ МАЛО ПРАВЫХ
 СКОБОК).

**TOO MANY RIGHT
 PARENTHESES**

ОШИБКА НА ВВОДЕ (СЛИШКОМ МНОГО
 ПРАВЫХ СКОБОК).

TYPE CONFLICT FOR

<ВЫРАЖЕНИЕ>

ВЫРАЖЕНИЕ БЫЛО ОБНАРУЖЕНО В КОНТЕКСТЕ
 НЕСООТВЕТСТВУЮЩЕГО ТИПА.

UNMATCHED INDEX ERROR
<СПИСОК>

В ПРОЦЕССЕ ВЫЧИСЛЕНИЯ ВЕКТОРНОГО
ВЫРАЖЕНИЯ ВСТРЕТИЛОСЬ НЕСООТВЕТСТВИЕ
ИНДЕКСОВ.

ZERO DENOMINATOR

ВСТРЕТИЛОСЬ ВЫРАЖЕНИЕ СО ЗНАМЕНАТЕЛЕМ,
РАВНЫМ НУЛЮ.

0/0 FORMED

СИСТЕМА НЕ МОЖЕТ ВЫПОЛНИТЬ ДЕЛЕНИЕ 0 НА
0 (0/0)

<ВЫРАЖЕНИЕ>

CAN NOT BE AN OPERATOR

<ВЫРАЖЕНИЕ> НЕ МОЖЕТ БЫТЬ ИСПОЛЬЗОВАНО
КАК СИМВОЛ ОПЕРАЦИИ

<ВЫРАЖЕНИЕ>

**INVALID PROCEDURE
NAME**

ПРОЦЕДУРЕ ДАНО НЕЗАКОННОЕ ИМЯ,

<ВЫРАЖЕНИЕ>

NOT FOUND

НЕ БЫЛО НАЙДЕНО НУЖНОЕ СИСТЕМЕ
(НАПРИМЕР, ДЛЯ ОПЕРАТОРА
CLEAR),
ВЫРАЖЕНИЕ,

<ИМЯ ФАЙЛА>

NOT OPEN

В КОМАНДЕ
SHUT
УКАЗАНО ИМЯ ФАЙЛА, КОТОРЫЙ НЕ БЫЛ ОТКРЫТ
ИЛИ УЖЕ ЗАКРЫТ.

<ИДЕНТИФИКАТОР>

UNDEFINED

БЫЛО ОБНАРУЖЕНО НЕРАСПОЗНАННОЕ ИМЯ
КОМАНДЫ,

<ЧИСЛО>

**TOO LONG FOR
FORTRAN**

БЫЛО ПОДГОТОВЛЕНО СЛИШКОМ
БОЛЬШОЕ ЦЕЛОЕ ЧИСЛО
ДЛЯ ФОРТРАН-ВЫВОДА.

<ТИП><ПЕРЕМЕННАЯ>

USED AS SCALAR

ПЕРЕМЕННАЯ ТИПА <ТИП> БЫЛА ИСПОЛЬЗОВАНА
В СКАЛЯРНОМ ВЫРАЖЕНИИ.

<ПЕРЕМЕННАЯ>

ALREADY DEFINED AS

<ТИП>

СДЕЛАНО ОПИСАНИЕ ДЛЯ УЖЕ ОПИСАННОЙ КАК
<ТИП> ПЕРЕМЕННОЙ.

<ПЕРЕМЕННАЯ>

INVALID

ИЛИ

<ПЕРЕМЕННАЯ>

INVALID IN

<ИМЯ ОПЕРАТОРА>

STATEMENT

СИНТАКСИЧЕСКАЯ ОШИБКА В ОПЕРАТОРЕ.

<ПЕРЕМЕННАЯ>

HAS NO MASS

ПЕРЕМЕННОЙ, ВСТРЕТИВШЕЙСЯ В ОПИСАНИИ

MSHELL ;

НЕ ПРИСВОЕНА МАССА.

A.4.2 ДИАГНОСТИЧЕСКИЕ СООБЩЕНИЯ.

ASSIGNMENT FOR

<ВЫРАЖЕНИЕ>

REDEFINED

<ВЫРАЖЕНИЮ> БЫЛО ПЕРЕПРИСВОЕНО НОВОЕ
ЗНАЧЕНИЕ.

COEFF GIVEN EXPRESSION

WITH DENOMINATOR

<ВЫРАЖЕНИЕ>

В

COEFF

ЗАДАНО ВЫРАЖЕНИЕ СО ЗНАМЕНАТЕЛЕМ,
ЯВЛЯЮЩИМСЯ ФУНКЦИЕЙ ЯДРА,
СТЕПЕНИ КОТОРОГО ДОЛЖНЫ БЫТЬ РАЗДЕЛЕННЫ.

MISSING END IN FILE

<ИМЯ>

ФАЙЛ <ИМЯ> НЕ ОГРАНИЧЕН ОПЕРАТОРОМ
END

<ВЫРАЖЕНИЕ>

REPRESENTED BY

<ВЫРАЖЕНИЕ>

СИСТЕМА ПРЕДСТАВЛЯЕТ ОДНО ВЫРАЖЕНИЕ ЧЕРЕЗ
ДРУГОЕ.

<ИМЯ>

REDEFINED

ОПЕРАЦИЯ ИЛИ ПРОЦЕДУРА БЫЛИ ПЕРЕ-
ОПРЕДЕЛЕННЫ.

<ПЕРЕМЕННАЯ>

DECLARED

<ТИП>

ПЕРЕМЕННОЙ БЫЛ ПРИСВОЕН ТИП <ТИП>.

В ОИЯИ СИСТЕМА ВНЕДРЕНА НА ЭВМ CDC=6500 (ОТВЕТСТВЕННЫЙ - РОСТОВЦЕВ ВИТАЛИЙ АЛЕКСАНДРОВИЧ, ТЕЛ.63536) И НА ЭВМ ЕС-1060 И ЕС-1040 (ОТВЕТСТВЕННАЯ - КРУГЛОВА ЛЮМИЛА ЮРЬЕВНА, ТЕЛ.64346),

НА CDC=6500 РЕАЛИЗОВАНА ВЕРСИЯ СИСТЕМЫ ОТ 15 МАРТА 1981 ГОДА, НА ЕС ЭВМ - ВЕРСИЯ ОТ 15 АПРЕЛЯ 1979 ГОДА,

В ВАРИАНТ СИСТЕМЫ НА ЭВМ CDC=6500 НЕ ВКЛЮЧЕНЫ БЛОКИ МАТРИЧНОЙ АЛГЕБРЫ, ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ И ИНТЕГРАТОР.

1. ОРГАНИЗАЦИЯ ПАКЕТА ЗАДАЧИ (ДЛЯ ЭВМ CDC=6500),

```

<ШИФР>.<ФАМИЛИЯ>
ACCOUNT(.,.,.)
REDUCE,
ATTACH,LISP4,ID=LCTA,MR=1,
LOAD,LISP4,
NOGO,
ATTACH,REDUCE2,IO=LCTA,MR=1,
LISP,S=REDUCE2.
7/8/9
<ПРОГРАММА НА ЯЗЫКЕ REDUCE >
END;
FIN

```

КАРТЫ ПРОБИВАЮТСЯ С 1 КОЛОНКИ ПО 72 ВКЛЮЧИТЕЛЬНО. НАПОМИНИМ, ЧТО ВМЕСТО ДВОЕТОЧИЯ СЛЕДУЕТ ПРОБИВАТЬ ЗНАК X, А ВМЕСТО ЗНАКА ПРОЦЕНТА, ПРЕДВАРЯЮЩЕГО КОММЕНТАРИИ - ЗНАК ВОПРОСА,

2. РАБОТА С СИСТЕМОЙ НА ЕС ЭВМ

ТАК ЖЕ, КАК И В ПРЕДЫДУЩЕМ СЛУЧАЕ, КАРТЫ ПРОБИВАЮТСЯ С 1 КОЛОНКИ ПО 72 ВКЛЮЧИТЕЛЬНО.

ОФОРМЛЕНИЕ ПАКЕТА ЗАДАНИЯ ДЛЯ РАБОТЫ СО "СТАНДАРТНОЙ" КОНФИГУРАЦИЕЙ СИСТЕМЫ (БЕЗ ИНТЕГРАТОРА),

```

//REDUCE JOB
// EXEC PGM=LISP,REGION=400K
//STEPLIB DD DSN=SYS1,LISP,DISP=SHR
//REDUCE DD DSN=SYS1,REDUCE,DISP=SHR
//LISPOUT DD SYSOUT=A
//LISPIN DD *
(RESTORE (QUOTE REDUCE))
(BEGIN)
<ПРОГРАММА НА ЯЗЫКЕ REDUCE >
END;
/*
//

```

400K - МИНИМАЛЬНЫЙ РАЗМЕР ПАМЯТИ, НЕОБХОДИМЫЙ ДЛЯ РАБОТЫ СИСТЕМЫ.

ДЛЯ ТОГО, ЧТОБЫ ОТКЛЮЧИТЬ БЛОК ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ (СИ

УВЕЛИЧИТЬ ТЕМ САМЫМ РАЗМЕР ПАМЯТИ НА 2000 ЯЧЕЕК). МЕЖДУ КАРТАМИ (RESTORE (QUOTE REDUCE)) И (BEGIN) СЛЕДУЕТ ПОМЕСТИТЬ КАРТУ (BPSWIPE (QUOTE INDEX)).

НА ЭВМ ЕС-1060 МОЖНО ВОСПОЛЬЗОВАТЬСЯ СТАНДАРТНЫМИ ПРОЦЕДУРАМИ :

ВЫЗОВ СИСТЕМЫ REDUCE :

```
//<ИМЯ ЗАДАНИЯ> JOB <ШИФР>, <ФАМИЛИЯ ПРОГРАММИСТА>
//      EXEC REDUCE[, REGN=NK]
// LISPIN DD *
// (RESTORE (QUOTE REDUCE))
// (BEGIN)
<ПРОГРАММА НА ЯЗЫКЕ REDUCE >
END
/*
//
```

ГДЕ N - ТРЕБУЕМЫЙ РАЗМЕР ПАМЯТИ. ПО УМОЛЧАНИЮ N = 400K.

ВЫЗОВ ИНТЕГРАТОРА :

```
//<ИМЯ ЗАДАЧИ> JOB <ШИФР>, <ФАМИЛИЯ ПРОГРАММИСТА>
// EXEC INT[, REGN=NK]
// LISPIN DD*
// (RESTORE (QUOTE INT))
// (BEGIN)
...
INT( <ПОДЫНТЕГРАЛЬНОЕ ВЫРАЖЕНИЕ>, <ПЕРЕМЕННАЯ ИНТЕРПРИРОВАНИЯ>)
...
/*
//
```

ВЫЗОВ СИСТЕМЫ REDUCE + ИНТЕГРАТОР :

ФОРМИРУЕТСЯ ТАКОЖЕ ПАКЕТ, КАК И В ПРЕДЫДУЩЕМ СЛУЧАЕ, ИЗМЕНЯЕТСЯ ТОЛЬКО КАРТА EXEC :

```
// EXEC REDINT[, REGN=NK]
```

ИНТЕРАКТИВНЫЙ РЕЖИМ (ЕС-1060).

ДЛЯ ТОГО, ЧТОБЫ РАБОТАТЬ НА ТЕРМИНАЛЕ В ИНТЕРАКТИВНОМ РЕЖИМЕ, НЕОБХОДИМО СНАЧАЛА СОЗДАТЬ СЛЕДУЮЩИЙ ЛОКАЛЬНЫЙ ФАЙЛ :

```
//<ИМЯ ЗАДАНИЯ> JOB <ШИФР>, <ФАМИЛИЯ ПРОГРАММИСТА>
//JOBLIB DD DSN=SYS1.LISP, DISP=SHR
// EXEC PGM=LOADGO, PARM=|LISP|, REGION=400K
//REDUCE DD DSN=SYS1.REDUCE, DISP=SHR
//LISPIN DD UNIT=SYSDA, SPACE=(80,(2)), DCB=(BLKSIZE=80,
//      BUFNO=1)
//LISPOUT DD UNIT=SYSDA, SPACE=(80,(2))
//
```

ЗАТЕМ СЛЕДУЕТ ЗАПОМНИТЬ ЭТОТ ФАЙЛ ПОД КАКИМ-ТО ИМЕНЕМ И ВЫПОЛНИТЬ КОМАНДУ TERMA

```
XRFQ, < ИМЯ ЛОКАЛЬНОГО ФАЙЛА >
```

ДОЖДАВШИСЬ ПОЯВЛЕНИЯ НА ЭКРАНЕ ТЕРМИНАЛА СИСТЕМНОГО СООБЩЕНИЯ О ВХОДЕ В ЛИСП, ВХОДИТЕ В REDUCE :

(RESTORE (QUOTE REDUCE))
(BEGIN)

ВЫЙТИ ИЗ ИНТЕРАКТИВНОГО РЕЖИМА МОЖНО, НАБРАВ ХА

СПИСОК ЛИТЕРАТУРЫ,

- [1] HEARN A.C., REDUCE 2 USER'S MANUAL, 2ND ED., UNIV. OF UTAH, 1973.
 - [2] REDUCE INTERACTIVE LESSONS, BY HEARN A.C., REDUCE NEWSLETTER, (UNIV. OF UTAH), 1978.
 - [3] ГЕРАТ В.П., ТАРАСОВ О.В., ШИРКОВ Д.В., УФН, 130, N 1 (1981).
 - [4] MCCARTHY, J., ABRAHAMS, P.W., EDWARDS, D.J., HART, T.P. AND LEVIN, M.I., LISP 1.5 PROGRAMMERS MANUAL, M.I.T. PRESS, 1965
 - [5] WEISSMAN, CLARK, LISP 1.5 PRIMER, DICKENSON, 1967
 - [6] HEARN, A.C., STANDARD LISP, STANFORD ARTIFICIAL INTEL. PROJECT MEMO A1-90 (MAY 1969).
 - [7] АЖ, А. БЪЕРКЕН, С. А. ДРЕЛЛ, "РЕЛЯТИВИСТСКАЯ КВАНТОВАЯ МЕХАНИКА", М, "НАУКА", 1978.
 - [8] CHIZHOLM, J.C.R., II NUOVO CIMENTO X, 30, 426 (1963)
 - [9] KAHANE, J., JOURNAL MATH, PHLS, 9, 1932 (1968)
 - [10] REDUCE-HANDBUCH, TECHNISCHE HOCHSCHULE KARL-MARX-STADT, 1980.
 - [11] SCONZO, P., LESCHACK, A.R., AND TOBEY, R., SYMBOLIC COMPUTATION OF F AND G SERIES BY COMPUTER, ASTRONOMICAL JOURNAL 70 (MAY 1965).
 - [12] ЕРШОВА Т.А., ОЛЕЯНИКОВА С.Г., СЫЧЕВ П.П., ОИЯИ.10-82-463. АУБНА, 1982.
 - [13] ЖЕЛЕЗНОВА К.М., КОРНЕЯЧУК А.А., ШАРАПОВА Э.В., ОИЯИ. 5-83-226. АУБНА, 1983.
 - [14] ГУСЕВ А.В., ОИЯИ.Б1-11-83-56, АУБНА, 1983.
- ****

1. 1. ВВЕДЕНИЕ.	1
1. КРАТКИЙ ОБЗОР СИСТЕМЫ REDUCE.	1
2. ПРИМЕРЫ ПРОГРАММ НА ЯЗЫКЕ REDUCE . ФОРМАТ ПРОГРАММ,	3
2. ДАННЫЕ И ОПЕРАЦИИ.	5
2.1 НЕКОТОРЫЕ СИНТАКСИЧЕСКИЕ ПРАВИЛА.	5
1. СТАНДАРТНЫЙ НАБОР ОСНОВНЫХ ЗНАКОВ ЯЗЫКА REDUCE .	5
2. ИДЕНТИФИКАТОРЫ.	5
3. КОММЕНТАРИИ.	6
2.2 ЧИСЛА,	6
2.3 ПЕРЕМЕННЫЕ.	7
1. ТИПЫ ПЕРЕМЕННЫХ.	7
2. ЗАРЕЗЕРВИРОВАННЫЕ ПЕРЕМЕННЫЕ.	8
2.4 МАССИВЫ.	8
2.5 ОПЕРАЦИИ.	9
1. ИНФИКСНЫЕ ОПЕРАЦИИ.	9
2. ПРЕФИКСНЫЕ ОПЕРАЦИИ.	11
3. ВСТРОЕННЫЕ ФУНКЦИИ.	11
4. ОПЕРАЦИЯ DF.	12
5. ОПЕРАЦИЯ SUB.	13
6. ОПЕРАЦИИ NUM И DEN.	13
7. ВВЕДЕНИЕ НОВЫХ ПРЕФИКСНЫХ ОПЕРАЦИЙ (ФУНКЦИЙ).	14
2.6 ЯДРА.	14
3. ВЫЧИСЛЯЕМЫЕ КОМАНДЫ : ВЫРАЖЕНИЯ И ОПЕРАТОРЫ.	16
3.1 ВЫРАЖЕНИЯ.	16
1. ЧИСЛОВЫЕ ВЫРАЖЕНИЯ.	16
2. СКАЛЯРНЫЕ (АЛГЕБРАИЧЕСКИЕ) ВЫРАЖЕНИЯ.	16
3. БУЛЕВСКИЕ ВЫРАЖЕНИЯ.	17
4. РАВЕНСТВА.	17
5. ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ВЫРАЖЕНИЯ В ЯЗЫКЕ REDUCE.	18
3.2 ОПЕРАТОРЫ.	18
3.3 ОПЕРАТОР ПРИСВАИВАНИЯ.	19

3.4 УСЛОВНЫЕ ОПЕРАТОРЫ.	22
3.5 ОПЕРАТОРЫ ЦИКЛА.	22
1. ОПЕРАТОРЫ WHILE И REPEAT.	22
2. ОПЕРАТОР FOR.	23
3.6 ОПЕРАТОР СУММИРОВАНИЯ.	24
3.7 ОПЕРАТОР ПРОИЗВЕДЕНИЯ.	24
3.8 БЛОК И ГРУППОВОЙ ОПЕРАТОР.	25
1. БЛОК.	25
2. ГРУППОВОЙ ОПЕРАТОР.	25
3. ОПЕРАТОР GO TO.	26
4. ОПЕРАТОР RETURN.	26
3.9 ОПЕРАТОР WRITE.	27
3.10 ПРОЦЕДУРЫ.	28
4. НЕВЫЧИСЛЯЕМЫЕ КОМАНДЫ.	31
4.1 КОМАНДЫ ПОДСТАНОВОК.	31
1. КОМАНДА LET .	31
2. ОПЕРАЦИЯ SUB.	33
3. ПОДСТАНОВКИ ДЛЯ ОБЩИХ ВЫРАЖЕНИЙ.	33
4.2 ДИФФЕРЕНЦИРОВАНИЕ ВЫРАЖЕНИЙ.	36
4.3 РАБОЧЕЕ ПОЛЕ ВЫРАЖЕНИЙ.	37
4.4 ОПИСАНИЯ.	38
1. ОПИСАНИЯ ON И OFF.	38
2. ОПИСАНИЕ CLEAR.	38
4.5 РАБОТА С ФАЙЛАМИ.	40
1. КОМАНДА IN , ФЛАГ ЕСО .	40
2. КОМАНДА END.	40
3. КОМАНДА OUT .	41
4. КОМАНДА SHUT.	41
4.6 КОНТРОЛЬ ЗА ВЫЧИСЛИТЕЛЬНЫМ ПРОЦЕССОМ.	42
4.7 НАХОЖДЕНИЕ ОБЩЕГО ДЕЛИТЕЛЯ ДВУХ ПОЛИНОМОВ.	42
4.8 ВЫВОД ВЫРАЖЕНИЯ.	43
1. ОПИСАНИЕ ORDER.	43

2. ОПИСАНИЕ FACTOR.	44
3. ОПИСАНИЯ UP И DOWN.	45
4. ФЛАГИ ВЫВОДА.	45
4.9 ЧИСЛЕННЫЕ РАСЧЕТЫ.	48
4.10 РАЗБИЕНИЕ ВЫРАЖЕНИЯ.	50
4.11 КОМАНДЫ, ИСПОЛЬЗУЕМЫЕ В ИНТЕРАКТИВНЫХ СИСТЕМАХ.	51
4.12 КОМАНДА DEFINE.	52
4.13 УНИЧТОЖЕНИЕ НУЛЯ НА ВЫВОДЕ. ФЛАГ ZERO.	52
4.14 АСИМПТОТИЧЕСКИЕ КОМАНДЫ.	52
5. МАТРИЧНЫЕ ВЫЧИСЛЕНИЯ.	54
5.1 ОПЕРАЦИЯ MAT.	54
5.2 МАТРИЧНЫЕ ПЕРЕМЕННЫЕ.	54
5.3 МАТРИЧНЫЕ ВЫРАЖЕНИЯ.	55
5.4 ОПЕРАЦИИ С МАТРИЧНЫМИ АРГУМЕНТАМИ.	56
1. ОПЕРАЦИЯ DET.	56
2. ОПЕРАЦИЯ TR.	56
3. НАХОЖДЕНИЕ ОБРАТНОЙ МАТРИЦЫ.	56
5.5 МАТРИЧНЫЕ ПРИСВАИВАНИЯ.	57
5.6 ВЫЧИСЛЕНИЕ МАТРИЧНЫХ ЭЛЕМЕНТОВ.	57
6. РАСЧЕТЫ В ФИЗИКЕ ВЫСОКИХ ЭНЕРГИЙ.	58
6.1 ВВЕДЕНИЕ.	58
6.2 ОПЕРАЦИИ, ИСПОЛЬЗУЕМЫЕ В ЗАДАЧАХ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ.	58
1. ОПЕРАЦИЯ . (ТОЧКА).	58
2. ОПЕРАЦИЯ G.	59
3. ОПЕРАЦИЯ EPS.	59
6.3 ВЕКТОРНЫЕ ПЕРЕМЕННЫЕ.	60
6.4 ДОПОЛНИТЕЛЬНЫЕ ТИПЫ ВЫРАЖЕНИЯ.	60
1. ВЕКТОРНЫЕ ВЫРАЖЕНИЯ.	60
2. ДИРАКОВСКИЕ ВЫРАЖЕНИЯ.	61
6.5 ВЫЧИСЛЕНИЕ ШПУРА.	61
6.6 ОПИСАНИЕ MASS.	61
6.7 ПРИМЕР.	62

6.8 КОМАНДА REDIM.	63
7. СИМВОЛЬНЫЙ РЕЖИМ.	64
7.1 ПРЕДВАРИТЕЛЬНЫЕ ЗАМЕЧАНИЯ.	64
7.2 ОБОБЩЕНИЕ ИДЕНТИФИКАТОРОВ.	65
7.3 СИМВОЛЬНЫЕ ИНФИКСНЫЕ ОПЕРАЦИИ.	65
7.4 СИМВОЛЬНЫЕ ВЫРАЖЕНИЯ.	65
7.5 QUOTE - ВЫРАЖЕНИЯ.	66
7.6 LAMBDA - ВЫРАЖЕНИЯ.	66
7.7 СИМВОЛЬНЫЕ ОПЕРАТОРЫ ПРИСВАИВАНИЯ.	67
7.8 СВЯЗЬ С АЛГЕБРАИЧЕСКИМ РЕЖИМОМ.	67
7.9 ТРАНСЛЯЦИЯ REDUCE - ПРОГРАММЫ В LISP.	68
8. ПРИЛОЖЕНИЕ А. ОСНОВНЫЕ ПОЛОЖЕНИЯ СИСТЕМЫ.	69
8.1 ЗАРЕЗЕРВИРОВАННЫЕ ИДЕНТИФИКАТОРЫ.	69
8.2 СТАНДАРТНЫЙ НАБОР КОМАНД ЯЗЫКА REDUCE.	69
8.3 ДИАГНОСТИКА И СООБЩЕНИЯ ОБ ОШИБКАХ В СИСТЕМЕ REDUCE.	74
1. СООБЩЕНИЯ ОБ ОШИБКАХ.	74
9. РАБОТА С СИСТЕМОЙ НА ЭВМ ОЯИ.	79
1. ОРГАНИЗАЦИЯ ПАКЕТА ЗАДАЧИ (ДЛЯ ЭВМ СРС-6500).	79
2. РАБОТА С СИСТЕМОЙ НА ЕС ЭВМ.	79

