

Задание 2. ЯЗЫК ПАСКАЛЬ. ДИНАМИЧЕСКИЕ СТРУКТУРЫ ДАННЫХ.

2.1. ПОСТАНОВКА ЗАДАЧИ

Варианты 1-11

Дана непустая последовательность слов, в каждом из которых содержится от 1 до 6 заглавных латинских букв; соседние слова разделены запятой, за последним словом следует точка.

Требуется ввести эту последовательность слов в память ЭВМ, преобразовав ее во внутреннее представление (см. ниже), а затем распечатать в алфавитном порядке:

- 1) все слова последовательности;
- 2) все слова с указанием для каждого из них его порядкового номера в исходной последовательности;
- 3) все различные слова с указанием для каждого из них числа его вхождений в исходную последовательность;
- 4) все различные слова с указанием для каждого из них порядкового номера его первого вхождения в исходную последовательность;
- 5) сначала все однобуквенные слова, затем все двухбуквенные слова и т.д.;
- 6) сначала все однобуквенные слова с указанием для каждого из них его порядкового номера в исходной последовательности, затем аналогичным образом все двухбуквенные слова и т.д.;
- 7) сначала все различные однобуквенные слова с указанием для каждого из них числа его вхождений в исходную последовательность, затем аналогичным образом все различные двухбуквенные слова и т.д.;
- 8) сначала все различные однобуквенные слова с указанием для каждого из них порядкового номера его первого вхождения в исходную последовательность, затем аналогичным образом все различные двухбуквенные слова и т.д.;
- 9) та же задача, что и в варианте 1;
- 10) та же задача, что и в варианте 3;
- 11) та же задача, что и в варианте 4.

В качестве внутреннего представления последовательности слов использовать:

- в вариантах 1-4 — однонаправленный список из слов, упорядоченных по алфавиту;
- в вариантах 5-8 — массив из 6 списков, в k -ом из которых хранятся k -буквенные слова, упорядоченные по алфавиту;
- в вариантах 9-11 — двоичное дерево поиска (в нем слева от каждой вершины слова располагаются только те слова, что предшествуют ему по алфавиту, а справа — следующие за ним по алфавиту).

Варианты 12-18

Дана символьная запись (см. ниже) многочлена (многочленов) от одной переменной X с целыми коэффициентами. Требуется ввести этот многочлен в память ЭВМ, преобразовав во внутреннее представление (см. ниже), выполнить операцию, определяемую вариантом задания, и распечатать полученный результат.

Операции над многочленами:

- 12) вычислить значение заданного многочлена при заданном целочисленном значении переменной X ;
- 13) проверить, имеет ли заданный многочлен хотя бы один целый корень (такими корнями могут быть только положительные и отрицательные делители свободного члена, а при нулевом свободном члене корнем является 0);
- 14) проверить на равенство два заданных многочлена;
- 15) получить многочлен, являющийся производной заданного многочлена по переменной X ;
- 16) получить многочлен, являющийся суммой двух заданных многочленов;
- 17) получить многочлен, являющийся произведением двух заданных многочленов;
- 18) привести подобные члены в заданном многочлене, в котором могут повторяться одночлены с одними и теми же степенями X .

Исходный многочлен от переменной X с целыми коэффициентами записывается как алгебраическая сумма одночленов любого из следующих видов (^ — возведение в степень):

ax^k ,
 x^k ,
 ax ,
 x
 a

где k и a — целые числа ($k \geq 2$, $a \geq 1$). При этом по степеням X одночлены могут быть не упорядочены, но одночлены одной и той же степени не повторяются (кроме варианта 18). За последним одночленом следует пробел — признак конца записи многочлена. (Особый случай: нулевой многочлен записывается как 0).

Примеры записи многочленов:

$7x^{30}$
 $x^8+1-139x+5x^{46}$

Если результатом операции является многочлен, то он должен быть распечатан в указанном виде (без нулевых слагаемых, без коэффициентов 1 и без показателей степени 0 и 1), но обязательно по убыванию степеней X .

В памяти ЭВМ многочлен должен быть представлен в виде однонаправленного списка, в котором каждому одночлену соответствует звено, содержащее степень этого одночлена и его коэффициент. Звенья списка должны быть упорядочены по убыванию степеней, звеньев с нулевыми коэффициентами не должно быть.

Варианты 19-22

Дана формула, содержащая переменную X (см. ниже). Требуется ввести эту формулу в память ЭВМ, преобразовав ее в двоичное дерево (см. ниже), выполнить операцию, указанную в варианте задания, и распечатать полученный результат.

Операции над формулами: w

- 19) по заданной формуле и целому числу вычислить значение этой формулы при значении переменной X , равном этому числу;
- 20) получить производную заданной формулы по переменной X ;
- 21) напечатать заданную формулу в префиксном виде;
- 22) напечатать заданную формулу в постфиксном виде.

Под «формулой» понимается запись следующего вида:

$\langle \text{формула} \rangle ::= \langle \text{число} \rangle \mid \langle \text{переменная} \rangle \mid (\langle \text{формула} \rangle \langle \text{знак} \rangle \langle \text{формула} \rangle)$
 $\langle \text{знак} \rangle ::= + \mid - \mid *$
 $\langle \text{число} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$
 $\langle \text{переменная} \rangle ::= X$

Примеры формул:

5
 $((2 * X) + (X - (8 + X)))$

Если результатом операции является снова формула, то она должна быть распечатана в таком же виде.

Формулу можно представить в виде двоичного дерева согласно следующим правилам:

- числу или переменной соответствует дерево из одной вершины, содержащей это число или переменную,
- формуле со знаком операции соответствует дерево, корневая вершина которого — знак, левое поддерево — соответствующее представление первого операнда, а правое поддерево — второго операнда.

Префиксной записью формулы $(a \# b)$, где $\#$ — знак некоторой операции, называется конструкция $(\# ab)$, а постфиксной записью — конструкция $(ab \#)$. Примеры записи формул:

обычная (инфиксная)	префиксная	постфиксная
7	7	7
$(a - (b * c))$	$(-a(*bc))$	$(a(bc*)-)$
$((a - b) * c)$	$(*(-ab)c)$	$((ab-)c*)$

2.2. ТРЕБОВАНИЯ К ПРОГРАММЕ

1. В программе должна быть предусмотрена проверка правильности задания исходной информации.
2. В вариантах 1-11 в программе должны быть определены процедура выделения очередного слова из исходной последовательности и процедура вставки слова в упорядоченный список (в дерево поиска). Кроме того, в вариантах 9-11 должна быть определена рекурсивная процедура печати слов, входящих в дерево.

3. Аналогичные процедуры (выделение очередного одночлена, вставка нового звена в упорядоченный список) должны быть определены в вариантах 12-18.
4. В вариантах 19-22 должны быть определены процедура ввода исходной формулы и построения соответствующего двоичного дерева, а также процедура вычисления значения формулы при заданной величине X (в варианте 19) или процедура печати формулы в нужном виде (варианты 20-22); все эти процедуры должны быть рекурсивными.

2.3. ЛИТЕРАТУРА

- [1]. Вирт Н. Алгоритмы + структуры данных = программы. — М.: Мир, 1985.
- [2]. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык Паскаль. — М.: Наука, 1988.
- [3]. Епанешников А.М., Епанешников В.А. Программирование в среде Turbo Pascal 7.0 — М.: «ДИАЛОГ-МИФИ», 2000.

2.4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

1. Вводить последовательность слов, многочлен или формулу следует посимвольно. Конец ввода определяется соответственно по точке, по пробелу или по балансу скобок. Ввод целого числа в вариантах 12 и 19 можно осуществлять с помощью процедуры `read(n)`, где n — целочисленная переменная.
2. В вариантах задания, где в качестве внутреннего представления используются списки, следует упорядочивать списки (по алфавиту или по степеням одночленов) одновременно с вводом слов или одночленов: введенное слово или одночлен следует сразу вставлять на «свое» место в ранее упорядоченный список.
3. Для упрощения операций над списками, рекомендуется использовать списки с заглавными звеньями. В этом случае следует в начале выполнения программы построить список (списки) с одним заглавным звеном.
4. В вариантах 1-11 в звеньях списков (вершинах дерева) следует хранить не только слова, но и дополнительную информацию (порядковый номер слова в исходной последовательности или число его вхождений в эту последовательность). В частности, даже в вариантах 1 и 5 для каждого слова лучше иметь только одно звено, фиксируя в нем число вхождений этого слова в последовательность; при печати же надо продублировать слово данное число раз. В варианте 9 такой прием является единственно возможным, поскольку в дереве поиска не должно быть нескольких вершин с одним и тем же словом.