

Блок-10. Машины (дополнительные) по теме «Двоичные деревья»

(во всех программах этого блока должен быть подключён модуль `heaptrc`)

Примечание. После занятия в субботу (12.12.2020) – актуально решить задачу 2. В задачах 1 и 3 используется *дерево поиска*, которое мы рассмотрим качественно только в понедельник (14.12.2020) и среду (16.12.2020).

задача 1 (10 очков)

В сданной обязательной 4-ой машинной задаче на файлы вместо списка воспользоваться *деревом поиска*. Т.е. вместо "процедуры вставки в список" написать "процедуру вставки в дерево поиска". Вместо "процедуры печати списка" написать "процедуру обхода дерева поиска (слева-направо) и попутной печати его элементов". Обе процедуры описать рекурсивно. Не допускать утечки памяти (т.е. дерево удалить перед окончанием работы программы)!

задача 2 (максимум 45 очков)

Дан текстовый файл с именем `form1.txt`, в каждой строке которого записана некоторая формула, синтаксис которой определяется в соответствии с БНФ:

`<ФОРМУЛА> ::= <ЧИСЛО> | <ПЕРЕМЕННАЯ> | (<ФОРМУЛА><ЗНАК><ФОРМУЛА>)`
`<ЗНАК> ::= + | - | *`
`<ЧИСЛО> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9`
`<ПЕРЕМЕННАЯ> ::= X`

ФОРМУЛУ МОЖНО ПРЕДСТАВИТЬ В ВИДЕ ДВОИЧНОГО ДЕРЕВА СОГЛАСНО СЛЕДУЮЩИМ ПРАВИЛАМ:

- 1) ЧИСЛУ ИЛИ ПЕРЕМЕННОЙ СООТВЕТСТВУЕТ ДЕРЕВО ИЗ ОДНОЙ ВЕРШИНЫ, СОДЕРЖАЩЕЙ ЭТО ЧИСЛО ИЛИ ПЕРЕМЕННУЮ;
- 2) ФОРМУЛЕ СО ЗНАКОМ ОПЕРАЦИИ СООТВЕТСТВУЕТ ДЕРЕВО, КОРНЕВАЯ ВЕРШИНА КОТОРОГО - ЗНАК, ЛЕВОЕ ПОДДЕРЕВО - СООТВЕТСТВУЮЩЕЕ ПРЕДСТАВЛЕНИЕ ПЕРВОГО ОПЕРАНДА, А ПРАВОЕ ПОДДЕРЕВО - ВТОРОГО ОПЕРАНДА.

См. также вступление к задаче **17.15** (где на примере *рисунка 21* даётся графическая иллюстрация *дерева-формулы*).

Требуется по файлу `form1.txt` сформировать текстовый файл `form2.txt` по следующему правилу. Для каждой формулы из исходного файла требуется выполнить следующую работу:

- 1) Требуется ввести эту формулу в память ЭВМ, преобразовав ее в двоичное дерево **T** (написав для этого соответствующую рекурсивную процедуру построения дерева-формулы по её записи);

2) **Распечатать** в отдельной строке результирующего файла **form2.txt** **введённую формулу** (выполнив рекурсивный обход построенного дерева **T** слева-направо), т.е. напечатанная формула должна совпасть с исходной (и тем самым можно проверить правильность выполненного построения);

3) По полученному дереву **T** **построить** дерево-формулу **T1** – **производную** дерева-формулы **T** по переменной **X** (продумать и рекурсивно реализовать необходимые процедуры);

4) **Распечатать** в следующей строке результирующего файла **найденную производную** в виде формулы (выполнив рекурсивный обход построенного дерева **T1**);

5) **Упростить** полученную дерево-формулу **T1**, заменив в нём все поддеревья, соответствующие формулам **(F+0)**, **(0+F)**, **(F-0)**, **(F*1)**, **(1*F)**, на поддерево, соответствующее формуле **F**, а поддеревья, соответствующие формулам **(F*0)** и **(0*F)**, на вершину с **0** (можно воспользоваться решением задачи **17.16a** – из задачника Пильщикова);

6) **Распечатать** в очередной строке формируемого файла **form2.txt** **результат упрощения формулы** для производной (выполнив рекурсивный обход преобразованного дерева **T1**);

Внимание! Нельзя допускать утечки памяти в процессе решения задачи (т.е. освобождать память, занятую под деревья **T** и **T1**, перед считыванием очередной строки исходного файла **form1.txt**)

Например, если в некоторой строке исходного файла **form1.txt** записана формула:

$$((X+1)*X)$$

То в трёх последовательных строках результирующего файла **form2.txt** должны появиться такие формулы:

$$\begin{aligned} &((X+1)*X) \\ &(((1+0)*X)+((X+1)*1)) \\ &(X+(X+1)) \end{aligned}$$

Желательно в файле **form2.txt** каждую новую тройку формул не сливать с предыдущей (отделять, например, пустой строкой или строкой из минусов) – для облегчения проверки.

Самостоятельно подготовить исходный файл **form1.txt** (в нём должно быть записано не менее пяти разнообразных формул). Прислать вместе с программой оба файла: исходный **form1.txt** и результирующий **form2.txt** .

Расценки: **15 очков** (за пункты 1 и 2) + **15 очков** (за пункты 3 и 4) + **15 очков** (за пункты 5 и 6). Т.е. **максимум 45 очков**.

задача 3 (максимум **45 очков**)

1) Формулировка этой задачи дана в номере **17.18** (Пильщикова). Умение строить *дерево поиска* и выполнять его обход Вы продемонстрируете в **задаче 1**. Так что основной проблемой в этой задаче будет разобраться с *анализом текста*. Каких-либо рекомендаций или строгих требований к задаче я не даю. Главное, не отступайте в решении от стандарта Паскаля. Можете в решении считать, что внутри диапазонов 'a..z' и 'A..Z' нет иных символов, кроме латинских букв. Пришлите результаты работы программы над двумя правильными текстами (**prog1.pas** - около 20 строк, **prog2.pas** - около 50 строк). Результаты поместите в файлы **ident1.txt** и **ident2.txt** соответственно (а на экране ничего печатать не надо, кроме отчёта модуля **heaptrc**).

2) Если сделаете задачу **17.18**, то дальше можно подправить её решение с учетом требований к **17.19**.

Расценки: **30 очков** (за пункт 1) + **15 очков** (за пункт 2). Т.е. **максимум 45 очков**.

Последний бой – он трудный самый!