Лабораторная работа №4

Нормальные алгоритмы Маркова

Задание 1*: Составить программу-эмулятор работы нормального алгоритма Маркова. Внешний алфавит, программа решения конкретной задачи и входной текст считываются из файла input.txt

Пример файла input.txt для решения задачи о нахождении целой части и остатка от деления на 3 числа, записанного в унарной системе счисления:

```
/
1.*///>/*
2.*!*
3.>*
/////
```

Замечание: завершающую команду будем обозначать !
Не завершающую >

Алгоритм решения задачи:

```
Program Alg_Markova;
Uses Crt;
Type Command=Record     {команда           }
      oldstr : String;  {искомый фрагмент  }
      step : Boolean;   {вид команды       }
      Newstr : String;  {строка подстановки}
      End;
      Arr_Com=Array[1..100] Of Command; {массив команд}
Var a : Arr_Com;
    alf : Set Of Char; {входной алфавит              }
    inp_str : String;  {входная строка               }
    Count : Integer;   {количество команд в алгоритме}
     F : Boolean;       {признак корректности ввода   }

Procedure Init_Alf;
Var i : Integer;
     s : String;
Begin
 ReadLn(s);
       {< формирование alf - множества символов строки s (входного алфавита) >}
End;

Procedure Init_A;
Var i : Integer;
     alg : String;
Begin
 {< пока не конец файла повторять: >}
 Begin
  { < считывание строки команды алгоритма > }
  { < определение позиции '.' - признака окончания номера команды > }
   { < определение i - номера команды > }
   { < удаление номера очередной команды > }
  { < формирование искомой подстроки (a[i].oldstr) i-ой команды > }
   { < формирование вида команды (a[i].step) > }
   { < формирование строки подстановки команды (a[i].newstr) > }
 End;
```

```
 Count:=i; {запомнить количество команд в алгоритме}
End;

Procedure Init_str;
Var i : Integer;
Begin
 { < считывание из файла входной строки inp_str > }
 { < проверка корректности ввода:                  }
 {  если ввод корректен, то f=true, иначе f=false > }
End;



Procedure Init;
Begin
 Assign(input,'input.txt');
 Reset(input);
 Init_alf; {процедура формирования alf - множества символов строки alfav}
 Init_A;   {формирование массива команд алгоритма}
             {a[i] - i-ая команда алгоритма, состоящая из трех полей:}
             {oldstr - преобразуемая подстрока                       }
             {step - вид команды: false - завершающая команда, true - в прот.случае}
             {newstr - подстрока подстановки}
 Init_str; {процедура ввода из файла входной строки и проверка корректности ввода}
 Close(input);
End;

Procedure Solve;
{процедура работы со строкой (inp_str) сформированного алгоритма Маркова (A)}
Begin
End;

Begin
 ClrScr;
 Init;
 WriteLn('Input string: ',inp_str);
 If f Then Solve;
 WriteLn('Answer: ',inp_str);
End.
```

Задание 2: Составить нормальные алгоритмы Маркова для решения следующих задач:

– уменьшение числа на единицу в системе счисления с основанием *p*;
– увеличение числа в два раза в системе счисления с основанием *p*;
– уменьшение числа в два раза в системе счисления с основанием *p*;
– удвоение каждого вхождения гласных букв в слове в алфавите *A*={*a*, *b*, *c*, *d*, *e*};
– нахождение остатка от деления числа на три в десятичной системе счисления.

Задание 3: Каков результат работы нормального алгоритма для входного слова ‘aababc’ в алфавите {a, b, c}? Определить область применимости данного алгоритма.

1. *a → #
2. *b → #
3. *c → #
4. #a → a#
5. #b → b#
6. #c → c#
7. a# !
8. b# !
9. c# !
10. → *