

1. На вход программе подается список текстовых файлов, которые содержат инструкции вида:

```
{  
    ShowVar;  
    var1=5;  
    {  
        var2=45;  
        var1=14;  
        ShowVar;  
        {  
            {  
                var3=677;  
            }  
            var5=1;  
            ShowVar;  
            var21=234;  
        }  
        var22=500;  
    }  
    var267=7;  
    ShowVar;  
}
```

Количество инструкций и их порядок произволен. Оператор “=” создает новую переменную и присваивает ей значение. Оператор ShowVar выводит на экран все видимые из данного места кода переменные, с указанием их значений. Фигурные скобки определяют блоки кода, которые определяют области видимости переменных. Если переменная покидает область видимости она становится недоступной. При объявлении в новой области видимости переменной, которая уже была

ранее объявлена, происходит «перекрытие» переменной. При возвращении в исходную область видимости становится доступно старое значение переменной.

2. Написать программу для интерпретации инструкций элементарного языка программирования. Рассматриваемый язык программирования позволяет оперировать переменными, имена которых произвольны, и создавать пользовательские функции. Через аргументы командной строки в программу подается файл с инструкциями вида:

```
foo(x, y): ((x*y+2)*(25-x/y))/(3-(x+2*y));  
myfoo2(z): z*z+4;  
myvar(i)=15;  
bg(f)=25;  
ccc=myfoo2(bg+myvar)*15+foo(bg*25+(6*myfoo2(myvar-10)));  
print ccc;  
bg=ccc*myvar;  
print;
```

Файл не содержит ошибок. Реализовать считывание файла и выполнение всех инструкций и операции print (вывод на экран либо текущего значения переменной, либо значений всех переменных с указанием имен). Имя переменной может иметь произвольную длину и содержать символы букв и цифр. Переменная создается в момент инициализации, при этом различают целые переменные (i) и вещественные (f). Заглавные и прописные буквы не отождествляются. В конце программы необходимо очистить всю выделенную память. В случае использования в вычислениях не объявленной переменной или функции вывести сообщение об ошибке.

3. Усовершенствуйте задание 2, разрешив функциям быть многострочными. Добавьте возможность вводить области видимости и

стандартное поведение переменных при вхождении в новую область  
видимости и при покидании текущей области видимости.