**Вариант 65 (***)**

Разработать систему для управления клеточным роботом, осуществляющим передвижение по клеточному лабиринту. Клетка лабиринта имеет форму квадрата.
Робот может передвинуться в соседнюю клетку в случае отсутствия в ней препятствия.

1. Разработать формальный язык для описания действий клеточного робота с поддержкой следующих литералов, операторов и предложений:
   - Логические литералы **TRUE**, **FALSE**;
   - Знаковых целочисленных литералов в десятичном формате;
   - Строковых литералов, строка ASCII символов заключенных в двойные кавычки “”; двойные кавычки не могут встречаться в строке символов, escape последовательности не используются;
   - Объявление переменных в форматах:
     - Переменная **VARIANT <имя переменной>** [**(размерность 1, размерность 2) = {{инициализатор 1 элемента 0,0, инициализатор 2 элемента 0,0,…; инициализатор элемента 0,1 … ;} {инициализатор элемента 1 элемента 1,0,…;…} … }**]; по умолчанию значение переменной логическое FALSE, целочисленное 0, пустая строка. Размерность по умолчанию 1,0; индексы элементов переменных начинаются с 0; в одном элементе может храниться как логическое, так и целочисленное, так и строковое значение; размерность задается арифметическим выражением; размерность может изменяться динамически, например, при обращении к элементу за пределами установленной размерности; возможно отсутствие инициализатора (используются значения по умолчанию), а также дублирование по типам, в этом случае используется последнее значение (пример объявления переменной: VARIABLE X [2,2] = {{TRUE, 775, “MEPhI”; “NRNU”, -100}{FALSE, “CSIT”, 12, TRUE;}});
   - Доступ к элементу массива <**имя переменной**> [**(индекс1, индекс2)**]; индексация элементов с 0; индекс по умолчанию 0,0; в качестве значения используется, значение типа определяемого типом выражения;

   Определены явные операторы преобразования типов **CONVERT (BOOL|DIGIT|STRING) TO (BOOL|DIGIT|STRING) <имя переменной>** [**(индекс1, индекс2)**] и **DIGITIZE <имя переменной>**, преобразующие любую переменную соответственно к логическому или целочисленному типу (TRUE => 1=“TRUE”, FALSE = 0= “FALSE”, (X<>0) = TRUE= “строковое представление числа”, (X=0) = FALSE = “0”; из “строка” из строки извлекается первое вхождение лексемы соответствующей записи числа, или соответствующей логической лексемы);

   В выражениях нельзя смешивать операции над элементом переменной и переменной в целом;

   - Оператор присваивания:
     - **<переменная>** [**(индекс1, индекс2)**] **= <выражение>** присвоение левому операнду значения правого; оператор правоассоциативен.

   Все логические и арифметические операторы выполняются на поэлементной основе.

   - Бинарных операторов сложения / логическое или / конкатенация:
     - **<выражение> + <выражение>**
   - Унарный оператор смены знака / логическое отрицание / замена команд роботу на противоположные (см . далее):
     - **- < выражение >**
   - Операторов цикла <**WHILE|UNTIL> <выражение> <предложение языка / группа предложений> <ENDW|ENDU>**, тело цикла выполняется до тех пор пока выражение истинно | ложно, выражение должно приводится к элементу.
   - Условных операторов **IF[N]LESS | IF[N]ZERO | IF[N]HIGH <выражение> <предложение языка / группа предложений>,** выполняется первое тело оператора, если

арифметическое выражение соответствует условию;

- Операторов управления роботом
  - Робот управляется «вербально» при помощи оператора **COMMAND <строковый литерал | выражение>**, робот выполняет следующие команды **UP**, **DOWN**, **LEFT**, **RIGHT**, перемещаясь в соответствующем направлении; **LOOKUP**, **LOOKDOWN**, **LOOKLEFT**, **LOOKRIGHT** возврат информации о препятствии в соответствующем направлении; пример команды “UP LEFT LEFT LEFT LEFT DOWN DOWN LOOKUP LOOKLEFT LOOKRIGHT LOOK DOWN”; информация о выполнение каждой команды возвращается в соответствующем поле (выполнена или нет в логическом, расстояние до объекта в числовом, информация об объекте (**WALL**, **EXIT**) в строковом), индекс результата соответствует номеру команды в строке в размерности 1, N, где N количество команд в строке.

- Описатель функции
  - **FUNC <имя функции> <предложение / группа предложений языка> ENDFUNC**. Функция является отдельной областью видимости, параметры передаются в функцию по значению; из функции параметр возвращается по значению при помощи оператора **RETURN <выражение>**. Доступ к параметру передаваемому в функцию осуществляется при помощи оператора **PARAM.** Функция может быть объявлена только в глобальной области видимости.
- Оператор вызова функции
  - **CALL <имя функции> <выражение>.**

Предложение языка завершается символом перевода строки. Язык является регистрозависимым.

2. Разработать с помощью flex и bison интерпретатор разработанного языка. При работе интерпретатора следует обеспечить контроль корректности применения языковых конструкций (например, инкремент/декремент константы); грамматика языка должна быть по возможности однозначной.

3. На разработанном формальном языке написать программу для поиска роботом выхода из лабиринта. Описание лабиринта, координаты выхода из лабиринта и начальное положение робота задается в текстовом файле.