Казанский (Приволжский) федеральный университет
Институт вычислительной математики и информационных технологий

Отчёт по дисциплине «Пакеты прикладных программ»

Работу выполнил:
Студент 09-822 группы
Шабаков Ильвар Жомортханович

Казань 2021

Оглавление

Задание 9(DO WHILE)

DO WHILE ... ENDDO Command

Executes a set of commands within a conditional loop.

DO WHILE lExpression

Commands

[LOOP]

[EXIT]

ENDDO

Parameters:

lExpression :

Specifies a logical expression whose value determines whether the commands between DO WHILE and ENDDO are executed. As long as lExpression evaluates to true (.T.), the set of commands are executed.

Commands :

Specifies the set of Visual FoxPro commands to be executed as long as lExpression evaluates to true (.T.).

LOOP :

Returns program control directly back to DO WHILE. LOOP can be placed anywhere between DO WHILE and ENDDO.

EXIT :

Transfers program control from within the DO WHILE loop to the first command following ENDDO. EXIT can be placed anywhere between DO WHILE and ENDDO.

LEX-файл

```
%option noyywrap
%option yylineno
%{
#include "y.tab.h"
```

```
#include <stdio.h>


int column = 0;
int error_count = 0;
%}

%%
DO" "WHILE            {column += yyleng;  printf("DO WHILE\n"); return DOWHILE;  }
ENDDO                 {column += yyleng;  printf("ENDDO\n"); return ENDDO;  }
LOOP                  {column += yyleng;  printf("LOOP\n"); return LOOP;  }
EXIT                  {column += yyleng;  printf("EXIT\n"); return EXIT;  }
"("                           {column += yyleng;  printf("LPAREN\n"); return OP;}
")"                           {column += yyleng;  printf("RPAREN\n"); return CP;}
"+"                          {column += yyleng;  printf("PLUS\n"); return PLUS;}
"-"                           {column += yyleng;  printf("MINUS\n"); return MINUS;}
"*"                           {column += yyleng;  printf("MULTIPLY\n"); return
MULTIPLY;}
"/"                          {column += yyleng;  printf("DIVIDE\n"); return DIVIDE;}
AND                           {column += yyleng;  printf("AND\n"); return AND;}
OR                            {column += yyleng;  printf("OR\n"); return OR;}
NOT                           {column += yyleng;  printf("NOT\n"); return NOT;}
0|([1-9][0-9]*)       {column += yyleng;  yylval.num = atoi(yytext); return NUMBER;}
[a-zA-Z][a-zA-Z0-9]*    {column += yyleng;  printf("IDENTIFIER = %s\n", yytext);
yylval.str = strdup(yytext); return IDENTIFIER;  }
;                     {column += yyleng;  printf("SEMICOLON\n"); return SEMICOLON;  }
[\n]                        {column=0;}
[ \t]                 {column++;}
"<="                  {column += yyleng;  printf("LEQ\n"); return LEQ;  }
">="                  {column += yyleng;  printf("GEQ\n"); return GEQ;  }
"<>"|"#"|"!="           {column += yyleng;  printf("NEQ\n"); return NEQ;  }
"=="                  {column += yyleng;  printf("STR_EQ\n"); return STR_EQ;  }
"="                   {column += yyleng;  printf("EQUAL\n"); return EQUAL;  }
"<"                   {column += yyleng;  printf("LESS\n"); return LESS;  }
">"                   {column += yyleng;  printf("GREATER\n"); return GREATER;  }

%%
%%
```

Пояснения :

Здесь все просто: в секции определений включена опция yylineno, нужная для отслеживания номера строки, на которой произошла ошибка. Так же создана переменная типа int(column) для отслеживания позиции на линии(столбца), в которой присутствует ошибка. Переменная error_count является счетчиком ошибок.

В секции правил прописаны регулярные выражения для определения лексем, которые нам понадобятся в синтаксическом анализаторе. В каждой лексеме прописана инструкция для подсчета позиции символа, которая обновляется при переходе на новую линию(*[\n] {column=0;}*).

YACC-файл

```
//yylval принимает значение int и string
%union
{
	int num;
	char *str;
}
//Числа принимают значение int
%token <num> NUMBER
//Переменные принимают значения string
%token <str> IDENTIFIER
%token DOWHILE ENDDO
%token SEMICOLON
%token LOOP EXIT
//Выражения принимают значения string
%type <num> exp term

//Приоритет операций. Чем ниже строка, тем выше приоритет
%left OP CP
%left LESS GREATER LEQ GEQ NEQ STR_EQ
%left OR
%left AND
%left NOT
%right EQUAL
%left PLUS MINUS
%left MULTIPLY DIVIDE

%{
#include <stdio.h>
#include <stdlib.h>
//Подключение переменных
#include "variables.h"

void yyerror(char *);
extern int yylex(void);
extern int yylineno;
extern int column;
extern int error_count;
```

```
symbol_block *symtab = NULL;
%}


%%
commands: /* пусто */
        | commands exp SEMICOLON  {printf("Вычисленное выражение: %d\n",
$2);}
        | commands command SEMICOLON
        | commands command error { yyerror("Пропущена точка с запятой");}
        | commands error SEMICOLON  { yyerror("Неправильная команда");}
        | commands LOOP SEMICOLON   { yyerror("LOOP вне DO WHILE");}
        | commands EXIT SEMICOLON   { yyerror("EXIT вне DO WHILE");}
        ;

command:
          DOWHILE lexp commands while_st ENDDO
        | IDENTIFIER  EQUAL  exp {
            symbol_assign(&symtab, $1, $3);}
        | IDENTIFIER  error exp { yyerror("Ошибка в присвоении значения
переменной"); }
        ;


while_st:       | EXIT
        | LOOP
        ;

exp:
          term
        | exp PLUS exp          { $$ = $1 + $3;   }
        | exp MINUS exp          { $$ = $1 - $3;   }
        | exp MULTIPLY exp       { $$ = $1 * $3;   }
        | exp DIVIDE  exp        {
                                  if ($3)
                                  $$ = $1 / $3;
                                  else
                                 {
                                  yyerror("Деление на ноль");
                                  $$=0;
                                 }
                                 }
        | OP exp CP             { $$ = $2;       }
        ;

lexp:           exp LESS exp          {printf("Выражение слева: %d\nВыражение справа:
%d\n", $1, $3);}
```

```
                | exp GREATER exp       {printf("Выражение слева: %d\nВыражение справа:
%d\n", $1, $3);}
                | exp LEQ exp          {printf("Выражение слева: %d\nВыражение справа:
%d\n", $1, $3);}
                | exp GEQ exp          {printf("Выражение слева: %d\nВыражение справа:
%d\n", $1, $3);}
                | exp NEQ exp          {printf("Выражение слева: %d\nВыражение справа:
%d\n", $1, $3);}
                | exp STR_EQ exp       {printf("Выражение слева: %d\nВыражение справа:
%d\n", $1, $3);}
                | exp error exp        { yyerror("Пропущен знак сравнения");}
                | lexp AND lexp
                | lexp OR lexp
                | NOT lexp
                | OP lexp CP
                ;

term:
                  NUMBER                { $$ = $1; }
                | IDENTIFIER  {
                    int dst;
                    if (symbol_get_value(&symtab, &dst, $1) == 1) {
                        $$ = dst; printf("Переменная %s имеет значение %d\n", $1, $$);}
                    else {
                        yyerror("Данной переменной не существует");}
                          }
            ;


%%
int main(int argc, char **argv){
   yyparse();
   if (error_count == 0) {
      printf("Ошибок в работе программы не обнаружено.\n");
   } else {
      printf("В ходе выполнения обнаружено %d ошибок.\n", error_count);
   }
}

void yyerror(char *str) {
          if (strcmp(str, "syntax error")) { printf("На %d строке %d столбце обнаружена
ошибка: %s\n", yylineno, column, str); error_count++; }
}
```

Пояснения:

В секции определений мы **1)**подключаем наш lex-файл, **2)**определяем токены для наших лексем, **3)**расширяем тип значения переменной yylval(через %union), **4)**задаем некоторым токенам возможные типы значения, которые они могут принимать(%token <num> NUMBER, %token <str> IDENTIFIER, %type <num> exp term), **5)**задаем приоритет и ассоциативность операторам умножения/деления, сложения/вычитания и т.д. Что не маловажно - **6)**подключаем возможность создания переменных, определенных в файле variables.h и variables.c и **7)**импортируем переменные для обработки ошибок(yylineno, column, error_count)
В секции правил прописаны синтаксические правила нашей грамматики для команды DO WHILE...ENDDO, реализовано вычисление простых выражений(+, -, *, /), добавлена работа с переменными и обработка ошибок.
В секции подпрограмм переопределена функция **main()**, в которой после чтения входного файла выводится информация о количестве обнаруженных ошибок. Так же переопределена функция **yyerror()** для дополнительного вывода номера строки и позиции в строке найденной в ходе выполнения программы ошибки.

В программе имеются как минимум две динамические ошибки(например, деление на ноль и отсутствие переменной) и как минимум 5 синтаксических ошибок(например, пропущенный закрывающий символ после выражения, ввод неверной команды , ошибка в присвоении значения переменной, LOOP/EXIT вне конструкции DO WHILE, пропуск знака сравнения в логических выражениях)

Файл для работы с переменными(variables.h)

```
#ifndef SYMBOLS_H
#define SYMBOLS_H

#include <string.h>
#include <stdlib.h>

#define BLOCK_SIZE 15
#define NAME_LEN 30

typedef struct symbol {
    char name[NAME_LEN]; // symbol name
    int val; // symbol value
    int hasVal; // 0 if symbol has notany value, 1 if has
} symbol;
```

```
typedef struct symbol_block {
    symbol *symbols; // pointer to symbol array of BLOCK_SIZE  size
    int cur_index; // current index put new element to
    struct symbol_block *next; // pointer to the next block
} symbol_block;

/* Read pointer to symbol to dst.
Return value:
-1 if symbols not found
0 if symbol found but hasVal is 0
1 if symbol found and hasVal is 1
Note: this function can be used to just check if some symbol exist or not*/
int symbol_get(symbol_block **root, symbol **dst, char *name);


/* Read symbol value to dst.
Return value:
-1 if symbols not found
0 if symbol found but hasVal is 0
1 if symbol found and hasVal is 1
Note: this function can be used to just check if some symbol exist or not*/
int symbol_get_value(symbol_block **root, int *dst, char *name);

/* Add symbol without assigning it any value
Note: if symbol already exists, function won't add a new one */
void symbol_add(symbol_block **root, char *name);

/* Add symbol and assign given value to it
Note: if symbol already exists, function will change its value,
if doesnt, function will create a new one with given value */
void symbol_assign(symbol_block **root, char *name, int value);

/* If root block is filled up, then alloc new and insert it before root */
void enlarge_if_need(symbol_block **root);

/*
Example usage:
symbol_block *root;
...
int x;
symbol_get_value(&root, &x, "foobar");

symbol *sym;
symbol_get(&root, &sym, "foo");
*/

#endif // SYMBOLS_H
```

Файл для работы с переменными(variables.c)

```
#include "variables.h"

int symbol_get(symbol_block **root, symbol **dst, char *name) {
symbol_block *cur_block = *root; int i;
while (cur_block != NULL) {
for (i = 0; i < cur_block->cur_index; i++) {
if (strcmp(cur_block->symbols[i].name, name) == 0) {
if (cur_block->symbols[i].hasVal == 1) {
*dst = &(cur_block->symbols[i]);
return 1;
}
else { return 0; }
}
}
cur_block = cur_block->next;
}
return -1;
}

int symbol_get_value(symbol_block **root, int *dst, char *name) {
symbol *sym;
int retval = symbol_get(root, &sym, name);
if (retval == 1) {
*dst = sym->val;
}
return retval;
}

void enlarge_if_need(symbol_block **root)
{
if (*root == NULL || (*root)->cur_index >= BLOCK_SIZE)
{
symbol_block *sb = (symbol_block*)malloc(sizeof(symbol_block));
sb->symbols = (symbol*)malloc(BLOCK_SIZE*sizeof(symbol));
sb->cur_index = 0;
sb->next = *root;
*root = sb;
}
}

void symbol_add(symbol_block **root, char *name)
{
int tmp; if (symbol_get_value(root, &tmp, name) != -1) { return; }
enlarge_if_need(root);
int ind = (*root)->cur_index;
```

```
strcpy((*root)->symbols[ind].name, name);
(*root)->symbols[ind].hasVal = 0;
(*root)->cur_index++;
}

void symbol_assign(symbol_block **root, char *name, int value)
{
symbol *sym;
if (symbol_get(root, &sym, name) != -1) {
sym->val = value;
sym->hasVal = 1;
return;
}
enlarge_if_need(root);
int ind = (*root)->cur_index;
strcpy((*root)->symbols[ind].name, name);
(*root)->symbols[ind].hasVal = 1;
(*root)->symbols[ind].val = value;
(*root)->cur_index++;
}
```

Входной тестовый файл

1. x=4;
2. y;
3. y=1;
4. y;
5. 1+2;
6. 4/2;
7. 6*2;
8. 3-1;
9. 4*5+4/2-3*5;
10. 1-(5+1);
11. 4+2*5;
12. 4*(6-4);
13. z=4*(6-3)+10*4-2;
14. DO WHILE(1<5) 5+5; ENDDO;
15. DO WHILE(x<=5) ENDDO;
16. DO WHILE(x<5) DO WHILE(1>5) 1+1; 1+2; ENDDO; ENDDO;
17. 4/0;
18. x.5;
19. var_test;
20. x+5lll;
21. DO WHILE(4>=1) x=4; y=x*6; DO WHILE(1<3) 1+1; LOOP ENDDO; ENDDO;
22. LOOP;
23. EXIT;

24. DO WHILE(CHECK<1) ENDDO;
25. DOWHILE;
26. DO WHILE
27. DO WHILE(x<5) 1+4; LOOP ENDDO;
28. DO WHILE(x>5 AND y<=3) x=3; y=3; EXIT ENDDO;
29. DO WHILE(x>5 OR y<=3) x=3; y=4; EXIT ENDDO;
30. DO WHILE(3>5 > y<=3) 5+5; EXIT ENDDO;
31. DO WHILE(1>6 AND y<=6) x=3; y=4; EXIT ENDDO;
32. DO WHILE(y!=3) x=3; y=4; EXIT ENDDO;
33. DO WHILE(x>5 AND y<=3) x=3; y=4; EXIT ENDDO;
34. DO WHILE(x>5 AND y<=3) EXIT 1+1; ENDDO;
35. DO WHILE(x>5 AND y<=3 AND z!=3) 1+1; ENDDO;
36. DO WHILE(x>5 OR y<=3 OR z!=3) LOOP ENDDO;
37. DO WHILE(x>5 AND y<=3 OR z!=3) EXIT ENDDO;
38. DO WHILE(1>5) 1+1 ENDDO;
39. DO WHILE(1>5) 1+1 ENDDO
40. DO WHILE(1>3) DO WHILE(1>5 AND x>3) 1+1; LOOP ENDDO; ENDDO;
41. DO WHILE(2>5) DO WHILE(1>5 AND x>3) 1+1; LOOP; ENDDO; ENDDO;
42. DO WHILE(x>1) DO WHILE(2>3 AND x>3) DO WHILE(x>3) 1+1; ENDDO; LOOP; ENDDO; ENDDO;
43. DO WHILE ENDDO;
44. DO WHILE ; ENDDO;
45. DO WHILE ENDDO
46. DO <> ENDDO;
47. 1/0;
48. 1/0
49. 4+4;
50. 4.5;
51. 4<.5
52. 3**;
53. DO WHILE(x>Z) DO WHILE(Z>1 AND x>3) DO WHILE(x>3) 1+1; ENDDO; LOOP; ENDDO; ENDDO;
54. DO WHILE(1>0) 1/0; ENDDO;
55. DO WHHILE(3>0) 1+1; ENDDO;
56. DO WHILE(x>1) ENDDO;
57. 1>5;
58. 1<3;
59. 1>2
60. DO WHILE(5>1) END DO;
61. DO WHILE(5>1) END DO;
62. DO WHILE(1>0 AND 5+5>15 OR 1+1<=3) ENDDO;
63. DO WHILE;
64. ENDDO;
65. LOOP;
66. EXIT;
67. DO WHILE(1>3) EXIT ENDDO;
68. DO WHILE(2<1) EXIT ENDDO;
69. 1+1

```
70. ;
71. 1+3;
72. DO WHILE(NOT z) EXIT ENDDO;
73. DO WHILE(x<y) 1+4; LOOP ENDDO;
74. DO WHILE(x>1 AND y<=3) x=3;EXIT ENDDO;
75. DO WHILE(x>3 OR y<=3)  y=4; EXIT ENDDO;
76. DO WHILE(x>5 OR y<=3) x=3; y=4; EXIT ENDDO;
77. DO WHILE(x>5 OR y<=3) EXIT 1+1; ENDDO;
78. DO WHILE(x>y AND y<=3 AND z!=2) 1+1; ENDDO;
79. DO WHILE(x>5 OR y<=3 OR z!=3) LOOP ENDDO;
80. DO WHILE(3>2 > y<3) 5+5;  ENDDO;
81. DO WHILE(1>6 OR y<=6) x=3; y=4; EXIT ENDDO;
82. DO WHILE(y!=3) x=3; y=4; EXIT ENDDO;
83. DO WHILE(x>5 AND y<=3 OR z!=3) EXIT ENDDO;
84. DO WHILE(2>5) DO WHILE(1>5 OR x>3) 1+1; LOOP; ENDDO; ENDDO;
85. DO WHILE(x>1) DO WHILE(2>3 OR x>3) DO WHILE(x>y) 1+3; ENDDO; EXIT;
    ENDDO; ENDDO;
86. 1=;2;
87. 1+3*4;
88. 1+1
89. 1+1;
90. X=1+3*4/0;
91. DO WHILE(x<Z) 1+3; EXIT ENDDO;
92. DO WHILE(NOT y<=3) X=3;EXIT ENDDO;
93. DO WHILE(x>3 AND y<=3)  y=4; LOOP ENDDO;
94. DO WHILE(x>5 OR y<=3 AND 1+1*2>5) y=4; LOOP; ENDDO;
95. DO WHILE(x>5 OR y<=3) EXIT 1+1; ENDDO;
96. 1.//2;
97. DO WHILE(x<+y) 1+1; EXIT LOOP ENDDO;
98. DO WHILE(x>1 AND y<=3) x=3;LOOOP ENDDO
99. DO WHILE(y<=3)  1+1; EXIT ENDDO;
100.        DO WHILE(x>5 AND y<=3) x=3; y=4; LOOP ENDDO;
101.        DO WHILE(x=>5 AND X<=1) EXIT 1+1; ENDDO;
```

Вывод тестового файла

```
IDENTIFIER  = x
EQUAL
SEMICOLON
IDENTIFIER  = y
SEMICOLON
На 2 строке 2 столбце обнаружена ошибка: Данной переменной не существует
Вычисленное выражение: 1955313120
IDENTIFIER  = y
EQUAL
```

SEMICOLON
IDENTIFIER = y
SEMICOLON
Переменная y имеет значение 1
Вычисленное выражение: 1
PLUS
SEMICOLON
Вычисленное выражение: 3
DIVIDE
SEMICOLON
Вычисленное выражение: 2
MULTIPLY
SEMICOLON
Вычисленное выражение: 12
MINUS
SEMICOLON
Вычисленное выражение: 2
MULTIPLY
PLUS
DIVIDE
MINUS
MULTIPLY
SEMICOLON
Вычисленное выражение: 7
MINUS
LPAREN
PLUS
RPAREN
SEMICOLON
Вычисленное выражение: -5
PLUS
MULTIPLY
SEMICOLON
Вычисленное выражение: 14
MULTIPLY
LPAREN
MINUS
RPAREN
SEMICOLON
Вычисленное выражение: 8
IDENTIFIER = z
EQUAL
MULTIPLY
LPAREN
MINUS
RPAREN
PLUS
MULTIPLY

MINUS
SEMICOLON
DO WHILE
LPAREN
LESS
RPAREN
Выражение слева: 1
Выражение справа: 5
PLUS
SEMICOLON
Вычисленное выражение: 10
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 4
LEQ
RPAREN
Выражение слева: 4
Выражение справа: 5
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 4
LESS
RPAREN
Выражение слева: 4
Выражение справа: 5
DO WHILE
LPAREN
GREATER
RPAREN
Выражение слева: 1
Выражение справа: 5
PLUS
SEMICOLON
Вычисленное выражение: 2
PLUS
SEMICOLON
Вычисленное выражение: 3
ENDDO
SEMICOLON
ENDDO
SEMICOLON
DIVIDE

На 17 строке 3 столбце обнаружена ошибка: Деление на ноль
SEMICOLON
Вычисленное выражение: 0
IDENTIFIER = x
.SEMICOLON
На 18 строке 3 столбце обнаружена ошибка: Ошибка в присвоении значения переменной
IDENTIFIER = var
_IDENTIFIER = test
На 19 строке 7 столбце обнаружена ошибка: Данной переменной не существует
SEMICOLON
На 19 строке 8 столбце обнаружена ошибка: Ошибка в присвоении значения переменной
IDENTIFIER = x
PLUS
Переменная x имеет значение 4
IDENTIFIER = lll
SEMICOLON
На 20 строке 7 столбце обнаружена ошибка: Неправильная команда
DO WHILE
LPAREN
GEQ
RPAREN
Выражение слева: 4
Выражение справа: 1
IDENTIFIER = x
EQUAL
SEMICOLON
IDENTIFIER = y
EQUAL
IDENTIFIER = x
Переменная x имеет значение 4
MULTIPLY
SEMICOLON
DO WHILE
LPAREN
LESS
RPAREN
Выражение слева: 1
Выражение справа: 3
PLUS
SEMICOLON
Вычисленное выражение: 2
LOOP
ENDDO
SEMICOLON
ENDDO
SEMICOLON

LOOP
SEMICOLON
На 22 строке 5 столбце обнаружена ошибка: LOOP вне DO WHILE
EXIT
SEMICOLON
На 23 строке 5 столбце обнаружена ошибка: EXIT вне DO WHILE
DO WHILE
LPAREN
IDENTIFIER = CHECK
На 24 строке 14 столбце обнаружена ошибка: Данной переменной не существует
LESS
RPAREN
Выражение слева: 1955313568
Выражение справа: 1
ENDDO
SEMICOLON
IDENTIFIER = DOWHILE
SEMICOLON
На 25 строке 8 столбце обнаружена ошибка: Данной переменной не существует
Вычисленное выражение: 1955313600
DO WHILE
DO WHILE
LPAREN
IDENTIFIER = x
LESS
RPAREN
PLUS
SEMICOLON
На 27 строке 18 столбце обнаружена ошибка: Неправильная команда
LOOP
ENDDO
SEMICOLON
На 27 строке 30 столбце обнаружена ошибка: Неправильная команда
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 4
GREATER
AND
Выражение слева: 4
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 24
LEQ
RPAREN
Выражение слева: 24
Выражение справа: 3
IDENTIFIER = x

EQUAL
SEMICOLON
IDENTIFIER = y
EQUAL
SEMICOLON
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
OR
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 3
LEQ
RPAREN
Выражение слева: 3
Выражение справа: 3
IDENTIFIER = x
EQUAL
SEMICOLON
IDENTIFIER = y
EQUAL
SEMICOLON
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
GREATER
GREATER
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
LEQ
RPAREN
PLUS
SEMICOLON
На 30 строке 25 столбце обнаружена ошибка: Неправильная команда
EXIT
ENDDO
SEMICOLON
На 30 строке 37 столбце обнаружена ошибка: Неправильная команда
DO WHILE

LPAREN
GREATER
AND
Выражение слева: 1
Выражение справа: 6
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
RPAREN
Выражение слева: 4
Выражение справа: 6
IDENTIFIER = x
EQUAL
SEMICOLON
IDENTIFIER = y
EQUAL
SEMICOLON
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = y
Переменная y имеет значение 4
NEQ
RPAREN
Выражение слева: 4
Выражение справа: 3
IDENTIFIER = x
EQUAL
SEMICOLON
IDENTIFIER = y
EQUAL
SEMICOLON
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
AND
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 4
LEQ

RPAREN
Выражение слева: 4
Выражение справа: 3
IDENTIFIER = x
EQUAL
SEMICOLON
IDENTIFIER = y
EQUAL
SEMICOLON
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
AND
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
RPAREN
Выражение слева: 4
Выражение справа: 3
EXIT
PLUS
SEMICOLON
На 34 строке 32 столбце обнаружена ошибка: Неправильная команда
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
AND
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
AND
Выражение слева: 4
Выражение справа: 3
IDENTIFIER = z
Переменная z имеет значение 50

NEQ
RPAREN
Выражение слева: 50
Выражение справа: 3
PLUS
SEMICOLON
Вычисленное выражение: 2
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
OR
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
OR
Выражение слева: 4
Выражение справа: 3
IDENTIFIER = z
Переменная z имеет значение 50
NEQ
RPAREN
Выражение слева: 50
Выражение справа: 3
LOOP
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
AND
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
OR
Выражение слева: 4
Выражение справа: 3
IDENTIFIER = z
Переменная z имеет значение 50

NEQ
RPAREN
Выражение слева: 50
Выражение справа: 3
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
GREATER
RPAREN
Выражение слева: 1
Выражение справа: 5
PLUS
ENDDO
SEMICOLON
На 38 строке 24 столбце обнаружена ошибка: Неправильная команда
DO WHILE
LPAREN
GREATER
RPAREN
Выражение слева: 1
Выражение справа: 5
PLUS
ENDDO
DO WHILE
LPAREN
GREATER
RPAREN
DO WHILE
LPAREN
GREATER
AND
IDENTIFIER = x
GREATER
RPAREN
PLUS
SEMICOLON
На 40 строке 40 столбце обнаружена ошибка: Неправильная команда
LOOP
ENDDO
SEMICOLON
ENDDO
SEMICOLON
DO WHILE
LPAREN
GREATER
RPAREN

Выражение слева: 2
Выражение справа: 5
DO WHILE
LPAREN
GREATER
AND
Выражение слева: 1
Выражение справа: 5
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
RPAREN
Выражение слева: 3
Выражение справа: 3
PLUS
SEMICOLON
Вычисленное выражение: 2
LOOP
SEMICOLON
На 41 строке 46 столбце обнаружена ошибка: LOOP вне DO WHILE
ENDDO
SEMICOLON
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
RPAREN
Выражение слева: 3
Выражение справа: 1
DO WHILE
LPAREN
GREATER
AND
Выражение слева: 2
Выражение справа: 3
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
RPAREN
Выражение слева: 3
Выражение справа: 3
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3

GREATER
RPAREN
Выражение слева: 3
Выражение справа: 3
PLUS
SEMICOLON
Вычисленное выражение: 2
ENDDO
SEMICOLON
LOOP
SEMICOLON
На 42 строке 67 столбце обнаружена ошибка: LOOP вне DO WHILE
ENDDO
SEMICOLON
ENDDO
SEMICOLON
DO WHILE
ENDDO
SEMICOLON
На 43 строке 15 столбце обнаружена ошибка: Неправильная команда
DO WHILE
SEMICOLON
На 44 строке 10 столбце обнаружена ошибка: Неправильная команда
ENDDO
SEMICOLON
На 44 строке 17 столбце обнаружена ошибка: Неправильная команда
DO WHILE
ENDDO
IDENTIFIER = DO
NEQ
ENDDO
SEMICOLON
На 46 строке 12 столбце обнаружена ошибка: Неправильная команда
DIVIDE
На 47 строке 3 столбце обнаружена ошибка: Деление на ноль
SEMICOLON
Вычисленное выражение: 0
DIVIDE
На 48 строке 3 столбце обнаружена ошибка: Деление на ноль
PLUS
SEMICOLON
На 49 строке 4 столбце обнаружена ошибка: Неправильная команда
.SEMICOLON
На 50 строке 3 столбце обнаружена ошибка: Неправильная команда
LESS
.MULTIPLY
MULTIPLY
SEMICOLON

На 52 строке 4 столбце обнаружена ошибка: Неправильная команда
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
IDENTIFIER = Z
На 53 строке 12 столбце обнаружена ошибка: Данной переменной не существует
RPAREN
Выражение слева: 3
Выражение справа: 1955314848
DO WHILE
LPAREN
IDENTIFIER = Z
На 53 строке 24 столбце обнаружена ошибка: Данной переменной не существует
GREATER
AND
Выражение слева: 1955314880
Выражение справа: 1
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
RPAREN
Выражение слева: 3
Выражение справа: 3
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
RPAREN
Выражение слева: 3
Выражение справа: 3
PLUS
SEMICOLON
Вычисленное выражение: 2
ENDDO
SEMICOLON
LOOP
SEMICOLON
На 53 строке 67 столбце обнаружена ошибка: LOOP вне DO WHILE
ENDDO
SEMICOLON
ENDDO
SEMICOLON
DO WHILE
LPAREN
GREATER

RPAREN
Выражение слева: 1
Выражение справа: 0
DIVIDE
На 54 строке 17 столбце обнаружена ошибка: Деление на ноль
SEMICOLON
Вычисленное выражение: 0
ENDDO
SEMICOLON
IDENTIFIER = DO
IDENTIFIER = WHHILE
На 55 строке 9 столбце обнаружена ошибка: Данной переменной не существует
LPAREN
На 55 строке 10 столбце обнаружена ошибка: Ошибка в присвоении значения переменной
На 55 строке 10 столбце обнаружена ошибка: Пропущена точка с запятой
GREATER
RPAREN
PLUS
SEMICOLON
На 55 строке 19 столбце обнаружена ошибка: Неправильная команда
ENDDO
SEMICOLON
На 55 строке 26 столбце обнаружена ошибка: Неправильная команда
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
RPAREN
Выражение слева: 3
Выражение справа: 1
ENDDO
SEMICOLON
GREATER
SEMICOLON
На 57 строке 4 столбце обнаружена ошибка: Неправильная команда
LESS
SEMICOLON
На 58 строке 4 столбце обнаружена ошибка: Неправильная команда
GREATER
DO WHILE
LPAREN
GREATER
RPAREN
IDENTIFIER = END
IDENTIFIER = DO
SEMICOLON

На 60 строке 21 столбце обнаружена ошибка: Неправильная команда
DO WHILE
LPAREN
GREATER
RPAREN
Выражение слева: 5
Выражение справа: 1
IDENTIFIER = END
IDENTIFIER = DO
На 61 строке 20 столбце обнаружена ошибка: Данной переменной не существует
SEMICOLON
На 61 строке 21 столбце обнаружена ошибка: Ошибка в присвоении значения переменной
DO WHILE
LPAREN
GREATER
AND
Выражение слева: 1
Выражение справа: 0
PLUS
GREATER
OR
Выражение слева: 10
Выражение справа: 15
PLUS
LEQ
RPAREN
Выражение слева: 2
Выражение справа: 3
ENDDO
SEMICOLON
DO WHILE
SEMICOLON
На 63 строке 9 столбце обнаружена ошибка: Неправильная команда
ENDDO
SEMICOLON
LOOP
SEMICOLON
На 65 строке 5 столбце обнаружена ошибка: LOOP вне DO WHILE
EXIT
SEMICOLON
На 66 строке 5 столбце обнаружена ошибка: EXIT вне DO WHILE
DO WHILE
LPAREN
GREATER
RPAREN
Выражение слева: 1
Выражение справа: 3

EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
LESS
RPAREN
Выражение слева: 2
Выражение справа: 1
EXIT
ENDDO
SEMICOLON
PLUS
SEMICOLON
Вычисленное выражение: 2
PLUS
SEMICOLON
Вычисленное выражение: 4
DO WHILE
LPAREN
NOT
IDENTIFIER = z
Переменная z имеет значение 50
RPAREN
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
LESS
IDENTIFIER = y
Переменная y имеет значение 4
RPAREN
На 73 строке 13 столбце обнаружена ошибка: Пропущен знак сравнения
PLUS
SEMICOLON
Вычисленное выражение: 5
LOOP
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
AND

Выражение слева: 3
Выражение справа: 1
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
RPAREN
Выражение слева: 4
Выражение справа: 3
IDENTIFIER = x
EQUAL
SEMICOLON
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
OR
Выражение слева: 3
Выражение справа: 3
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
RPAREN
Выражение слева: 4
Выражение справа: 3
IDENTIFIER = y
EQUAL
SEMICOLON
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
OR
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
RPAREN
Выражение слева: 4
Выражение справа: 3

IDENTIFIER = x
EQUAL
SEMICOLON
IDENTIFIER = y
EQUAL
SEMICOLON
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
OR
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
RPAREN
Выражение слева: 4
Выражение справа: 3
EXIT
PLUS
SEMICOLON
На 77 строке 31 столбце обнаружена ошибка: Неправильная команда
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
IDENTIFIER = y
Переменная y имеет значение 4
AND
Выражение слева: 3
Выражение справа: 4
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
AND
Выражение слева: 4
Выражение справа: 3
IDENTIFIER = z
Переменная z имеет значение 50
NEQ

RPAREN
Выражение слева: 50
Выражение справа: 2
PLUS
SEMICOLON
Вычисленное выражение: 2
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
OR
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
OR
Выражение слева: 4
Выражение справа: 3
IDENTIFIER = z
Переменная z имеет значение 50
NEQ
RPAREN
Выражение слева: 50
Выражение справа: 3
LOOP
ENDDO
SEMICOLON
DO WHILE
LPAREN
GREATER
GREATER
Выражение слева: 3
Выражение справа: 2
IDENTIFIER = y
LESS
RPAREN
PLUS
SEMICOLON
На 80 строке 24 столбце обнаружена ошибка: Неправильная команда
ENDDO
SEMICOLON
На 80 строке 32 столбце обнаружена ошибка: Неправильная команда
DO WHILE
LPAREN

GREATER
OR
Выражение слева: 1
Выражение справа: 6
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
RPAREN
Выражение слева: 4
Выражение справа: 6
IDENTIFIER = x
EQUAL
SEMICOLON
IDENTIFIER = y
EQUAL
SEMICOLON
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = y
Переменная y имеет значение 4
NEQ
RPAREN
Выражение слева: 4
Выражение справа: 3
IDENTIFIER = x
EQUAL
SEMICOLON
IDENTIFIER = y
EQUAL
SEMICOLON
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
AND
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
OR

Выражение слева: 4
Выражение справа: 3
IDENTIFIER = z
Переменная z имеет значение 50
NEQ
RPAREN
Выражение слева: 50
Выражение справа: 3
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
GREATER
RPAREN
Выражение слева: 2
Выражение справа: 5
DO WHILE
LPAREN
GREATER
OR
Выражение слева: 1
Выражение справа: 5
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
RPAREN
Выражение слева: 3
Выражение справа: 3
PLUS
SEMICOLON
Вычисленное выражение: 2
LOOP
SEMICOLON
На 84 строке 45 столбце обнаружена ошибка: LOOP вне DO WHILE
ENDDO
SEMICOLON
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
RPAREN
Выражение слева: 3
Выражение справа: 1
DO WHILE

LPAREN
GREATER
OR
Выражение слева: 2
Выражение справа: 3
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
RPAREN
Выражение слева: 3
Выражение справа: 3
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
IDENTIFIER = y
Переменная y имеет значение 4
RPAREN
Выражение слева: 3
Выражение справа: 4
PLUS
SEMICOLON
Вычисленное выражение: 4
ENDDO
SEMICOLON
EXIT
SEMICOLON
На 85 строке 66 столбце обнаружена ошибка: EXIT вне DO WHILE
ENDDO
SEMICOLON
ENDDO
SEMICOLON
EQUAL
SEMICOLON
На 86 строке 3 столбце обнаружена ошибка: Неправильная команда
SEMICOLON
Вычисленное выражение: 2
PLUS
MULTIPLY
SEMICOLON
Вычисленное выражение: 13
PLUS
PLUS
SEMICOLON
На 89 строке 4 столбце обнаружена ошибка: Неправильная команда
IDENTIFIER = X
EQUAL

PLUS
MULTIPLY
DIVIDE
На 90 строке 9 столбце обнаружена ошибка: Деление на ноль
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
LESS
IDENTIFIER = Z
На 91 строке 12 столбце обнаружена ошибка: Данной переменной не существует
RPAREN
Выражение слева: 3
Выражение справа: 1955316448
PLUS
SEMICOLON
Вычисленное выражение: 4
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
NOT
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
RPAREN
Выражение слева: 4
Выражение справа: 3
IDENTIFIER = X
EQUAL
SEMICOLON
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
AND
Выражение слева: 3
Выражение справа: 3
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
RPAREN

Выражение слева: 4
Выражение справа: 3
IDENTIFIER = y
EQUAL
SEMICOLON
LOOP
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
OR
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
AND
Выражение слева: 4
Выражение справа: 3
PLUS
MULTIPLY
GREATER
RPAREN
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
EQUAL
SEMICOLON
LOOP
SEMICOLON
На 94 строке 44 столбце обнаружена ошибка: LOOP вне DO WHILE
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
OR
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
RPAREN

Выражение слева: 4
Выражение справа: 3
EXIT
PLUS
SEMICOLON
На 95 строке 31 столбце обнаружена ошибка: Неправильная команда
ENDDO
SEMICOLON
.DIVIDE
DIVIDE
SEMICOLON
На 96 строке 5 столбце обнаружена ошибка: Неправильная команда
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
LESS
PLUS
IDENTIFIER = y
Переменная y имеет значение 4
RPAREN
На 97 строке 14 столбце обнаружена ошибка: Пропущен знак сравнения
PLUS
SEMICOLON
Вычисленное выражение: 2
EXIT
LOOP
ENDDO
SEMICOLON
На 97 строке 36 столбце обнаружена ошибка: Неправильная команда
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
AND
Выражение слева: 3
Выражение справа: 1
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
RPAREN
Выражение слева: 4
Выражение справа: 3
IDENTIFIER = x
EQUAL
SEMICOLON
IDENTIFIER = LOOOP

ENDDO
DO WHILE
LPAREN
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
RPAREN
На 99 строке 14 столбце обнаружена ошибка: Ошибка в присвоении значения переменной
На 99 строке 14 столбце обнаружена ошибка: Пропущена точка с запятой
PLUS
SEMICOLON
На 99 строке 20 столбце обнаружена ошибка: Неправильная команда
EXIT
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
GREATER
AND
Выражение слева: 3
Выражение справа: 5
IDENTIFIER = y
Переменная y имеет значение 4
LEQ
RPAREN
Выражение слева: 4
Выражение справа: 3
IDENTIFIER = x
EQUAL
SEMICOLON
IDENTIFIER = y
EQUAL
SEMICOLON
LOOP
ENDDO
SEMICOLON
DO WHILE
LPAREN
IDENTIFIER = x
Переменная x имеет значение 3
EQUAL
GREATER
AND
На 101 строке 17 столбце обнаружена ошибка: Пропущен знак сравнения
IDENTIFIER = X

Переменная X имеет значение 3
LEQ
RPAREN
Выражение слева: 3
Выражение справа: 1
EXIT
PLUS
SEMICOLON
На 101 строке 33 столбце обнаружена ошибка: Неправильная команда
ENDDO
SEMICOLON
В ходе выполнения обнаружено 65 ошибок.