

Приложение Б - Таблица переходов для языка MATH

Сост.	Терминал	Переход	Принять	В Стек	Из Стека	Ошибка	Зн. Ошибки
1	PROGRAM	2	-	-	-	-	None
2	PROGRAM	3	+	-	-	+	Error 2
3	name,E	8	-	4	-	+	Error 3
4	BEGIN	5	+	-	-	+	Error 4
5	TYPE_I,TYPE_R,name,SET_V,GET_V,IF,E	12	-	6	-	+	Error 5
6	END	7	+	-	-	+	Error 6
7	PROGRAM	1	-	-	-	-	None
8	name	10	-	-	-	-	None
9	E	11	-	-	-	-	None
10	name	0	+	-	+	+	Error 10
11	E	0	-	-	+	+	Error 11
12	TYPE_I,TYPE_R	15	-	-	-	-	None
13	name,SET_V,GET_V,IF	19	-	-	-	-	Error 13
14	E	21	-	-	-	+	Error 14
15	TYPE_I,TYPE_R	22	-	16	-	+	Error 15
16	name	27	-	17	-	+	Error 16
17	;	18	+	-	-	+	Error 17
18	TYPE_I,TYPE_R,name,SET_V,GET_V,IF,E	12	-	-	-	+	Error 18
19	name,SET_V,GET_V,IF	34	-	20	-	+	Error 19
20	TYPE_I,TYPE_R,name,SET_V,GET_V,IF,E	12	-	-	-	+	Error 20
21	E	0	-	-	+	+	Error 21
22	TYPE_I	24	-	-	-	-	None
23	TYPE_R	25	-	-	-	+	Error 23
24	TYPE_I	0	+	-	+	+	Error 24
25	TYPE_R	0	+	-	+	+	Error 25
26	name	27	-	-	-	+	Error 26
27	name	28	+	-	-	+	Error 27
28	„E	29	-	-	-	+	Error 28
29	,	31	-	-	-	-	None
30	E	33	-	-	-	+	Error 30
31	,	32	+	-	-	+	Error 31
32	name	26	-	-	-	+	Error 32
33	E	0	-	-	+	+	Error 33
34	name	38	-	-	-	-	None
35	SET_V	43	-	-	-	-	None
36	GET_V	47	-	-	-	-	None
37	IF	51	-	-	-	-	None
38	name	39	+	-	-	+	Error 38
39	:=	40	+	-	-	+	Error 39
40	name,literal	59	-	41	-	+	Error 40
41	;	42	+	-	-	+	Error 41
42	TYPE_I,TYPE_R,name,SET_V,GET_V,IF,E	12	-	-	-	+	Error 42
43	SET_V	44	+	-	-	+	Error 43
44	name,literal,	98	-	45	-	+	Error 44
45	;	46	+	-	-	+	Error 45
46	TYPE_I,TYPE_R,name,SET_V,GET_V,IF,E	12	-	-	-	+	Error 46
47	GET_V	48	+	-	-	+	Error 47
48	name	49	+	-	-	+	Error 48
49	;	50	+	-	-	+	Error 49
50	TYPE_I,TYPE_R,name,SET_V,GET_V,IF,E	12	-	-	-	+	Error 50
51	IF	52	+	-	-	+	Error 51
52	[53	+	-	-	+	Error 52
53	name	104	-	54	-	+	Error 53

Сост.	Терминал	Переход	Принять	В Стек	Из Стека	Ошибка	Зн. Ошибки
54	„E	108	-	55	-	+	Error 54
55]	56	+	-	-	+	Error 55
56	TYPE_I,TYPE_R,name,SET_V,GET_V,IF,E	12	-	57	-	+	Error 56
57	END_IF	58	+	-	-	+	Error 57
58	TYPE_I,TYPE_R,name,SET_V,GET_V,IF,E	12	-	-	-	+	Error 58
59	name	61	-	-	-	-	None
60	literal	63	-	-	-	+	Error 60
61	name	62	+	-	-	+	Error 61
62	+, -, *, /, >, <, >=, <=, !=, =, E	65	-	-	-	+	Error 62
63	literal	64	+	-	-	+	Error 63
64	+, -, *, /, >, <, >=, <=, !=, =, E	65	-	-	-	+	Error 64
65	+, -, *, /, >, <, >=, <=, !=, =	67	-	-	-	-	None
66	E	69	-	-	-	+	Error 66
67	+, -, *, /, >, <, >=, <=, !=, =	70	-	68	-	+	Error 67
68	name, literal	94	-	-	+	+	Error 68
69	E	0	-	-	+	+	Error 69
70	+, -, *, /	72	-	-	-	-	None
71	>, <, >=, <=, !=, =	73	-	-	-	+	Error 71
72	+, -, *, /	74	-	-	-	+	Error 72
73	>, <, >=, <=, !=, =	82	-	-	-	+	Error 73
74	+	78	-	-	-	-	None
75	-	79	-	-	-	-	None
76	*	80	-	-	-	-	None
77	/	81	-	-	-	+	Error 77
78	+	0	+	-	+	+	Error 78
79	-	0	+	-	+	+	Error 79
80	*	0	+	-	+	+	Error 80
81	/	0	+	-	+	+	Error 81
82	>	88	-	-	-	-	None
83	<	89	-	-	-	-	None
84	>=	90	-	-	-	-	None
85	<=	91	-	-	-	-	None
86	!=	92	-	-	-	-	None
87	=	93	-	-	-	+	Error 87
88	>	0	+	-	+	+	Error 88
89	<	0	+	-	+	+	Error 89
90	>=	0	+	-	+	+	Error 90
91	<=	0	+	-	+	+	Error 91
92	!=	0	+	-	+	+	Error 92
93	=	0	+	-	+	+	Error 93
94	name	96	-	-	-	-	None
95	literal	97	-	-	-	+	Error 95
96	name	0	+	-	+	+	Error 96
97	literal	0	+	-	+	+	Error 97
98	name, literal	100	-	-	-	-	None
99		101	-	-	-	+	Error 99
100	name, literal	59	-	-	-	+	Error 100
101		102	+	-	-	+	Error 101
102	literal	103	+	-	-	+	Error 102
103		0	+	-	+	+	Error 103
104	name	105	-	-	-	+	Error 104
105	name	106	+	-	-	+	Error 105
106	>, <, >=, <=, !=, =	82	-	107	-	+	Error 106
107	name, literal	94	-	-	-	+	Error 107
108	,	110	-	-	-	-	Error 108

Сост.	Терминал	Переход	Принять	В Стек	Из Стека	Ошибка	Зн. Ошибки
109	E	112	-	-	-	+	Error 109
110	,	111	+	-	-	+	Error 110
111	name	104	-	-	-	+	Error 111
112	E	0	-	-	+	+	Error 112

Приложение В – Листинги функций для формирования приложения компилятора

```
//Функция чтения текста из файла
//возвращаемый параметр: 1 - успешно; 0 - безуспешно
//передаваемые параметры: fPath - строка пути к файлу;
//Text - ссылка для возвращения текста из файла
bool GetFileText(CString fPath, CString &Text)
{
    CFile f;           //объект класса работы с файлами
    char BUF[100];     //массив для хранения 100 символов
    int count;         //счетчик прочтенных символов
    CString vs;        //строка для временного хранения текста

    Text.Empty(); //очистка строки по ссылке

    if(f.Open(fPath,CFile::modeRead)) //если есть возможность открыть файл по заданному пути
    {
        while((count=f.Read(BUF,100))>0)//пока вычитывается больше 0 символов
        {
            vs=BUF;           //перезапись массива символов в строку
            Text+=vs.Left(count); //формирование строки с текстом
        }
        f.Close();           //закрытие файла
        return 1;            //вернуть 1
    }
    else                      //если файл не был открыт
    {
        AfxMessageBox("Can't open Classes file!");//вывод сообщения об ошибке
        return 0;            //вернуть 0
    }
}
```

```

//Функция формирования динамического массива основных классов
//возвращаемый параметр: 1 - успешно; 0 - безуспешно
//передаваемые параметры: ClsStr - строка с текстом из файла классов;
//pClsArr - переменная-указатель на указатель структуры классов
//res - ссылка для возвращения размерности сформированного массива
bool FormClassesArr(CString ClsStr, Classes **pClsArr, int& res)
{
    int pos=0;           //позиция курсора строки
    int vpos;            //позиция найденного символа или комбинации
    int num;             //количество символов в подстроке
    int i=0;             //счетчик элементов динамического массива
    int ClsCnt=0;        //счетчик классов
    int WrdCnt=0;        //счетчик служебных слов и операторов
    CString vs;          //подстрока
//ПОДСЧЕТ КОЛИЧЕСТВА ЭЛЕМЕНТОВ МАССИВА КЛАССОВ
//цикл перебора подстрок, разделенных "\r\n"
    while((vpos=ClsStr.Find("\r\n",pos))!=-1) //пока в строке присутствует "\r\n"
    {
        num=vpos-pos;           //подсчет числа символов подстроки
        vs=ClsStr.Mid(pos,num);  //перезапись подстроки
        if(vs.Find(".")!=-1)     //если в подстроке присутствует символ "."
            WrdCnt++;            //увеличение счетчика служебных слов и операторов
        pos+=num+2;              //перемещение курсора
    }
//если была проанализирована не вся строка
    if(pos<ClsStr.GetLength()-1)
    {
        num=ClsStr.GetLength()-pos; //подсчет числа символов подстроки
        vs=ClsStr.Mid(pos,num);      //перезапись подстроки
        if(vs.Find(".")!=-1)         //если в подстроке присутствует символ "."
            WrdCnt++;                //увеличение счетчика служебных слов и операторов
    }
//ФОРМИРОВАНИЕ МАССИВА КЛАССОВ
    if(WrdCnt==0) //если не найдено служебных слов и операторов
        return 0; //вернуть 0
    *pClsArr=new Classes[WrdCnt]; //создание динамического массива

```

```

res=WrdCnt;           //фиксация размерности массива классов
pos=0;                //сброс переменной позиции курсора строки
//цикл перебора подстрок, разделенных "\r\n"
while((vpos=ClsStr.Find("\r\n",pos))!=-1) //пока в строке присутствует "\r\n"
{
    num=vpos-pos;           //подсчет числа символов подстроки
    vs=ClsStr.Mid(pos,num); //перезапись подстроки

    if((vpos=vs.Find("."))!=-1) //если в подстроке присутствует символ "."
    {
        (*pClsArr+i)->num.Format("%i-",ClsCnt); //сохранить номер класса
        (*pClsArr+i)->num+=vs.Left(vpos);       //сохранить номер в классе
        (*pClsArr+i)->word=vs.Right(vs.GetLength()-vpos-2); //сохранить слово
        i++;                                     //инкремент счетчика элементов массива
    }
    else if(vs.Find("class")!=-1) //если в строке присутствует слово "class"
        ClsCnt++;               //инкремент счетчика классов
    pos+=num+2;                 //перемещение курсора
}
//если была проанализирована не вся строка
if(pos<ClsStr.GetLength()-1)
{
    num=ClsStr.GetLength()-pos; //подсчет числа символов подстроки
    vs=ClsStr.Mid(pos,num);     //перезапись подстроки

    if((vpos=vs.Find("."))!=-1) //если в подстроке присутствует символ "."
    {
        (*pClsArr+i)->num.Format("%i-",ClsCnt); //сохранить номер класса
        (*pClsArr+i)->num+=vs.Left(vpos);       //сохранить номер в классе
        (*pClsArr+i)->word=vs.Right(vs.GetLength()-vpos-2); //сохранить слово
    }
}
return 1; //вернуть 1
}

```

```
//Функция выполнения лексического анализа
//возвращаемый параметр: лексическая последовательность или строка ошибки
//передаваемые параметры:
//NCCode - код программы без комментариев
//pClsArr - указатель на массив классов
//res - размер массива классов
//pNClsArr - переменная-указатель на указатель структуры классов
//Nres - ссылка для возвращения размерности сформированного массива
CString Lexical(CString NCCode, Classes* pClsArr, int res, Classes** pNClsArr, int& Nres)
{
    CString NewStr;      //лексическая последовательность
    CString result; //результат обработки анализа слова
    CString code; //код класс/в классе слова
    CString Symb;        //последовательность разделителей
    bool flag=0;  //флаг для организации цикла
    int pos;       //позиция найденной последовательности

//значение, характеризующее подстроку
//(0-элемент класса; 1-элемент класса (специальные символы, определяющие значение);
// 2-переменная; 3-значение; 4-недопустимое значение)
    int val;
    CString word; //слово для анализа
    int VarNum=0, ValNum=0; //счетчики значений и имен переменных
    CString VarCls, ValCls; //строки для формирования классов значений и имен переменных
    int lcls;               //номер последнего класса (специальные символы)

//получение номера последнего класса
    lcls=atoi((pClsArr+res-1)->num.Left((pClsArr+res-1)->num.Find("-")));

    Symb.Format("%c%c%c%c%c",10,13,9,32); //разделители, включая пробел
    while(!flag) //цикл до установки флага в 1
    {
        pos=NCCode.FindOneOf(Symb); //поиск позиции символа разделителя
        if(pos!=-1) //если разделитель был найден
        {
            word=NCCode.Left(pos); //получение текущего слова
            NCCode=NCCode.Right(NCCode.GetLength()-pos-1); //урезание строки слева на слово + разделитель
        }
    }
}
```

```

        if(!word.IsEmpty()) //если строка слова не пустая
        {
            val=CheckWord(word,pClsArr,res,code); //анализ слова
//обработка результата анализа
            result=CheckAResult(val,NCCode,pos,NewStr,VarNum,ValNum,VarCls,ValCls,word,code,lcls);
            if(result!="OK") //если обработка прошла с ошибкой
                return result; //вернуть строку с ошибкой
        }
    }
    else
    {
        if(NCCode.GetLength()!=0) //если строка не пуста
        {
            val=CheckWord(NCCode,pClsArr,res,code); //анализ слова
//обработка результата анализа
            result=CheckAResult(val,NCCode,pos,NewStr,VarNum,ValNum,VarCls,ValCls,NCCode,code,lcls);
            if(result!="OK") //если обработка прошла с ошибкой
                return result; //вернуть строку с ошибкой
        }
        flag=1; //установка флага в 1
    }
}

Nres=VarNum+ValNum; //фиксация размера массива по ссылке
*pNClsArr=new Classes[Nres]; //создание динамического массива
int i=0; //счетчик элементов массива новых классов
//перебор всех записей в строке класса значений
while(VarNum>0)
{
    pos=VarCls.Find(","); //определение позиции ","
    (*pNClsArr+i)->num=VarCls.Left(pos); //запись номера класс/в классе
    VarCls=VarCls.Right(VarCls.GetLength()-pos-1); //урезание строки слева
    pos=VarCls.Find("\r\n"); //определение позиции "\r\n"
    (*pNClsArr+i)->word=VarCls.Left(pos); //запись значения
}

```



```
VarCls=VarCls.Right(VarCls.GetLength()-pos-2);//урезание строки слева
i++;    //инкремент счетчика элементов
VarNum--;    //декремент счетчика значений
}
//перебор всех записей в строке класса имен переменных
while(ValNum>0)
{
    pos=ValCls.Find(",");    //определение позиции ","
    (*pNClsArr+i)->num=ValCls.Left(pos);//запись номера класс/в классе
    ValCls=ValCls.Right(ValCls.GetLength()-pos-1);//урезание строки слева
    pos=ValCls.Find("\r\n");    //определение позиции "\r\n"
    (*pNClsArr+i)->word=ValCls.Left(pos);//запись имени переменной
    ValCls=ValCls.Right(ValCls.GetLength()-pos-2);//урезание строки слева
    i++;    //инкремент счетчика элементов
    ValNum--;    //декремент счетчика имен переменных
}

return NewStr;    //вернуть лексическую последовательность
}
```

```
//Функция удаления и подсчета комментариев
//возвращаемый параметр: строка с удаленными комментариями
//передаваемые параметры: Code - строка с текстом программы;
//CntCom - ссылка для возвращения числа удаленных комментариев;
//MLC1 - открывающая последовательность многострочного комментария;
//MLC2 - закрывающая последовательность многострочного комментария;
//SLC - последовательность однострочного комментария
CString DelComments(CString Code, int &CntCom, CString MLC1, CString MLC2, CString SLC)
{
    CntCom=0; //обнуление переменной по ссылке
    int fpos,lpos; //переменные для фиксации позиций найденных последовательностей

    //пока в программе присутствуют многострочные комментарии
    while((fpos=Code.Find(MLC1,0))!=-1)
    {
        lpos=Code.Find(MLC2,fpos+MLC2.GetLength()); //поиск закрывающей последовательности
        CntCom++; //инкремент счетчика комментариев
        if(lpos!=-1) //если закрывающая последовательность найдена
            Code.Delete(fpos,lpos+MLC2.GetLength()-fpos); //удаление комментария
        else //если закрывающая последовательность не найдена
            Code=Code.Left(fpos); //удаление комментария
    }

    //пока в программе присутствуют однострочные комментарии
    while((fpos=Code.Find(SLC,0))!=-1)
    {
        lpos=Code.Find(13,fpos); //поиск символа за комментарием (перехода на новую строку)
        CntCom++; //инкремент счетчика комментариев
        if(lpos!=-1) //если символ за комментарием найден (переход на новую строку)
            Code.Delete(fpos,lpos-fpos); //удаление комментария
        else //если символ за комментарием не найден (переход на новую строку)
            Code=Code.Left(fpos); //удаление комментария
    }
    return Code; //вернуть строку, очищенную от комментариев
}
```

//Функция анализа текущего слова (оператора)

```
//возвращаемый параметр: характеристика подстроки
//0-элемент класса (простой);
//1-элемент класса (специальные символы, определяющие значение);
//2-переменная;
//3-значение;
//4-недопустимая комбинация
//передаваемые параметры: word - анализируемое слово;
//pClsArr - указатель на классы;
//res - размер массива классов
//code - ссылка для возвращения кода из класса
//fSC - флаг, определяющий наличие в качестве последнего класса со спец. символами
char CheckWord(CString word, Classes *pClsArr, int res, CString& code, bool fSC)
{
    int lcls;                //номер последнего класса (специальные символы)
    int ncls;                //номер текущего класса
    //получение номера последнего класса
    lcls=atoi((pClsArr+res-1)->num.Left((pClsArr+res-1)->num.Find("-")));
    for(int i=0; i<res; i++)    //сопоставление слова классам
    {
        if((pClsArr+i)->word==word)    //если слово соответствует записи классов
        {
            code=(pClsArr+i)->num;    //записать в ссылку номер класса/в классе
            ncls=atoi(code.Left(code.Find("-")));    //получение номер текущего класса
            if (fSC && lcls==ncls)    //если присутствуют спец. символы и этот символ один из них
                return 1;    //вернуть 1
            else
                return 0;    //вернуть 0
        }
    }
    char ch;                //значение текущего символа слова
    bool f=0;                //флаг контроля последовательности
    int n=word.GetLength();    //количество символов в слове
    //перебор всех символов слова
    for(i=0;i<n;i++)        //проверка слова на принадлежность именам
    {
        ch=word.GetAt(i);    //получение текущего символа
```

```

        if(i==0)
        {
            if(!((ch<=122 && ch>=97)|| (ch<=90 && ch>=65)|| ch==95)) //если символ - не буква или "_"
                f=1; //установить флаг в 1
        }
        else //если текущий символ не 0й
//если символ - не буква, число или "_"
            if(!((ch<=122 && ch>=97)|| (ch<=90 && ch>=65)|| ch==95)|| (ch<=57 && ch>=48)))
                f=1; //установить флаг в 1
    }
    if(!f) //если флаг не был установлен
        return 2; //вернуть 2
    f=0;
//перебор всех символов слова
    for(i=0;i<n;i++) //проверка слова на принадлежность числам
    {
        ch=word.GetAt(i); //получение текущего символа
        if(i==0) //если текущий символ 0й
        {
//если символ - не число
            if(!(ch<=57 && ch>=48))
                f=1; //установить флаг в 1
        }
        else //если текущий символ не 0й
//если символ - не число или точка
            if(!(ch==46 || (ch<=57 && ch>=48)))
                f=1; //установить флаг в 1
    }
    if(!f) //если слово - число
        return 3; //вернуть 3
    else //если слово - недопустимая комбинация
        return 4; //вернуть 4
}
//Функция проверки результата анализа слова
//возвращаемый параметр: номер ошибки или "ОК"
//передаваемые параметры: val - результат анализа

```

```
//NCCode - ссылка на строку с кодом
//pos - ссылка на текущую позицию в коде
//NewStr - ссылка на формируемую лексическую последовательность
//VarNum - ссылка на счетчик записей в классе имен
//ValNum - ссылка на счетчик записей в классе значений
//VarCls - ссылка на строку - класс имен
//ValCls - ссылка на строку - класс значений
//word - анализированное слово
//code - номер класса/в классе слова
//LCV - номер последнего основного класса
CString CheckAResult(char val, CString& NCCode, int& pos, CString& NewStr, int& VarNum, int& ValNum, CString& VarCls, CString& ValCls,
CString word, CString code, char LCV)
{
    CString Symb;           //строка разделителей
    int spos, epos;         //позиции первого и последнего символов номера класса/в классе
    CString s;              //переменная для формирования строки
    Symb.Format("%c%c%c",10,13,9); //разделители, не включая пробел
    if(val<2)               //если слово найдено в классах
    {
        NewStr+=code+","; //добавление кода в лексическую последовательность
        if(val==1)        //если слово является спец. символом
        {
            pos=NCCode.Find(word); //поиск следующего спец. символа
            if(NCCode.FindOneOf(Symb)<pos) //если спец. символ идет за разделителями
                return "Error 1 - Специальный символ следует за разделителями"; //вернуть ошибку
            if(pos!=-1)     //если специальный символ был найден
            {
                if(NCCode.GetAt(pos-1)!=32) //если перед спец. символом нет пробела
                    return "Error 2 - Перед специальным символом нет пробела"; //вернуть ошибку
            }
        }
        //если значение еще не занесено в класс
        if((epos=VarCls.Find(NCCode.Left(pos-1)+"\r\n"))==-1)
        {
            //создание номера класса/в классе
            s.Format("%i-%i,",LCV+1,++VarNum);
            NewStr+=s; //добавление кода в лексическую последовательность
            VarCls+=s+NCCode.Left(pos)+"\r\n"; //добавление номера в класс
        }
    }
}
```

```

    }
//если значение занесено в класс
else
{
    epos=-1;      //коррекция позиции последнего символа
    s=VarCls.Left(epos+1); //помещение в строку класса до слова
    spos=s.ReverseFind(10); //поиск позиции символа начала кода
    if(spos==-1)   //если позиция найдена
        NewStr+=s;  //добавление кода в лексическую последовательность
    else           //если позиция не найдена
        NewStr+=s.Mid(spos+1,epos-spos); //добавление кода в лексическую последовательность
}
NCCode=NCCode.Right(NCCode.GetLength()-pos-1); //урезание строки слева
NewStr+=code+","; //добавление кода в лексическую последовательность
}
}
}
else if(val==2)    //если слово является значением
{
//если значение еще не занесено в класс
if((epos=ValCls.Find(", "+word+"\r\n"))==-1)
{
//создание номера класса/в классе
s.Format("%i-%i,",LCV+2,++ValNum);
NewStr+=s; //добавление кода в лексическую последовательность
ValCls+=s+word+"\r\n"; //добавление номера в класс
}
//если значение занесено в класс
else
{
s=ValCls.Left(epos+1); //помещение в строку класса до слова
spos=s.ReverseFind(10); //поиск позиции символа начала кода
if(spos==-1) //если позиция найдена
    NewStr+=s; //добавление кода в лексическую последовательность
else //если позиция не найдена
    NewStr+=s.Mid(spos+1,epos-spos); //добавление кода в лексическую последовательность

```

```

    }
}
else if(val==3)    //если слово является именем
{
//если значение еще не занесено в класс
    if((epos=VarCls.Find(", "+word+"\r\n"))== -1)
    {
//создание номера класса/в классе
        s.Format("%i-%i",LCV+1,++VarNum);
        NewStr+=s;    //добавление кода в лексическую последовательность
        VarCls+=s+word+"\r\n"; //добавление номера в класс
    }
//если значение занесено в класс
    else
    {
        s=VarCls.Left(epos+1); //помещение в строку класса до слова
        spos=s.ReverseFind(10); //поиск позиции символа начала кода
        if(spos== -1) //если позиция найдена
            NewStr+=s;    //добавление кода в лексическую последовательность
        else //если позиция не найдена
            NewStr+=s.Mid(spos+1,epos-spos); //добавление кода в лексическую последовательность
    }
}
else //если слово недопустимая последовательность
{
    AfxMessageBox(word);
    return "Error 3 - Ошибочная последовательность";    //вернуть ошибку
}
return "OK"; //вернуть "OK"
}
//Функция получения значений полей строки таблицы переходов
//возвращаемый параметр: нет
//передаваемые параметры:
//SyntTable - строка с таблицей переходов
//Num - номер строки для чтения
//Val - ссылка для возвращения поля "Терминал"

```

```
//Next - ссылка для возвращения значения поля "Переход"
//Access - ссылка для возвращения значения поля "Принять"
//StackIn - ссылка для возвращения значения поля "В Стек"
//StackOut - ссылка для возвращения значения поля "Из Стекa"
//Err - ссылка для возвращения значения поля "Ошибка"
//Error - ссылка для возвращения значения поля "Значение Ошибки"
void GetSyntTableString(CString SyntTable, int Num, CString &Val, int &Next, bool &Access, int &StackIn, bool &StackOut, bool &Err, CString
&Error)
{
    CString vs;    //переменная для хранения строки таблицы
    CString st=SyntTable;    //таблица синтаксического разбора
    int pos;    //позиция последовательности
    if(Num==1)    //если обрабатывается первая строка таблицы
    {
        pos=st.Find("\r\n",0); //получение позиции перехода на новую строку
        vs=st.Left(pos);    //пересохранение содержимого строки
    }
    else    //если обрабатывается не первая строка таблицы
    {
        vs.Format("\r\n%i\t",Num); //формирование подстроки поиска
        int n=st.Find(vs);    //получение позиции искомой подстроки
        if((pos=st.Find("\r\n",n+2))!=-1)//если получение позиции перехода на новую строку было успешным
            vs=st.Mid(n+2,pos-n-2);    //пересохранение содержимого строки
        else    //иначе, - если текущая строка - последняя
            vs=st.Mid(n+2,st.GetLength()-n-2); //пересохранение содержимого строки
    }
    pos=vs.Find("\t");    //поиск разделителя записей

    vs=vs.Right(vs.GetLength()-pos-1);//урезание подстроки слева
    pos=vs.Find("\t");    //поиск разделителя записей
    Val=vs.Left(pos);    //возврат по ссылке поля "Терминал"

    vs=vs.Right(vs.GetLength()-pos-1);//урезание строки слева
    pos=vs.Find("\t");    //поиск разделителя записей
    Next=atoi(vs.Left(pos));    //возврат по ссылке значения поля "Переход"
```



```

vs=vs.Right(vs.GetLength()-pos-1);//урезание строки слева
pos=vs.Find("\t");    //поиск разделителя записей
if(vs.GetAt(0)=='+') //возврат по ссылке значения поля "Принять"
    Access=1;
else    Access=0;

vs=vs.Right(vs.GetLength()-pos-1);//урезание строки слева
pos=vs.Find("\t");    //поиск разделителя записей
if(vs.GetAt(0)=='-') //возврат по ссылке значения поля "В Стек"
    StackIn=0;
else
    StackIn=atoi(vs.Left(pos));

vs=vs.Right(vs.GetLength()-pos-1);//урезание строки слева
pos=vs.Find("\t");    //поиск разделителя записей
if(vs.GetAt(0)=='+') //возврат по ссылке значения поля "Из Стекa"
    StackOut=1;
else
    StackOut=0;

vs=vs.Right(vs.GetLength()-pos-1);//урезание строки слева
pos=vs.Find("\t");    //поиск разделителя записей
if(vs.GetAt(0)=='+') //возврат по ссылке значения поля "Ошибка"
    Err=1;
else
    Err=0;

vs=vs.Right(vs.GetLength()-pos-1);//урезание строки слева
Error=vs;    //возврат по ссылке значения поля "Значение Ошибки"
}
//Функция получения значений полей строки таблицы переходов
//возвращаемый параметр: строка с результатом разбора
//передаваемые параметры:
//Table - строка с таблицей переходов
//LexStr - лексическая последовательность
//LCN - число основных классов

```

```
CString Syntactical(CString Table, CString LexStr, int LCN)
{
    int Stack[100];          //стек на 100 значений
    int index=0;             //индекс стека

    int Cur=1;               //номер строки таблицы переходов
    CString Val;             //значение поля "Терминал"
    int Next;                //значение поля "Переход"
    bool Access;             //значение поля "Принять"
    int StackIn;             //значение поля "В Стек"
    bool StackOut;           //значение поля "Из Стекa"
    bool Err;                //значение поля "Ошибка"
    CString Error;           //значение поля "Значение Ошибки"

    CString st;              //переменная для формирования строки отчета
    CString SyntReport;      //строка, содержащая отчет по разбору
    CString vs;              //переменная для хранения текущего кода последовательности
    int spos=0;              //начальная позиция текущего номера последовательности
    int epos=LexStr.Find(", ", spos); //конечная позиция текущего номера последовательности
    while(epos!=-1)          //пока в последовательности есть номера
    {
        //получение значений полей строки таблицы переходов
        GetSyntTableString(Table, Cur, Val, Next, Access, StackIn, StackOut, Err, Error);
        vs=LexStr.Mid(spos, epos-spos);         //получение текущего номера последовательности

        //формирование отчета
        st.Format("'%s' - |%.3i| %s|i%i|i%i|i%i|i%i| %s\r\n", vs, Cur, Val, Next, Access, StackIn, StackOut, Err, Error);
        Report(st);
        SyntReport+=st;

        if(IsInString(Val, vs, LCN))            //если номер присутствует среди терминалов
        {
            if(Access)                          //если необходимо принять
            {
                spos=epos+1;                    //сменить начальную позицию номера последовательности
                epos=LexStr.Find(", ", spos);    //сменить конечную позицию номера последовательности
            }
        }
    }
}
```

```
    if(StackIn!=0)                //если необходимо сохранение в стек
        Stack[index++]=StackIn; //запись в стек
    if(StackOut && Next==0)        //если необходимо извлечение из стека и переход на 0
        Cur=Stack[--index];      //текущему номеру строки присвоить значение из стека
    else
        Cur=Next;                //текущему номеру строки присвоить значение перехода
}
else                             //если номер отсутствует среди терминалов
{
    if(Err)                       //если это влечет ошибку
    {
        FormReport(SyntReport,Error); //сформировать отчет, завершающийся ошибкой
        return Error;                //вернуть значение ошибки
    }
    else                          //если это не влечет ошибки
        Cur++;                     //перейти на следующую строку
}
}
FormReport(SyntReport,"No Errors"); //сформировать отчет
return "No Errors!";                //вернуть строку
}
```

```
//Функция проверки наличия номера в строке терминалов
//возвращаемый параметр: 1-присутствует; 0-отсутствует
//передаваемые параметры:
//Str - строка с таблицей переходов
//SubStr - лексическая последовательность
//LCN - число основных классов
bool IsInString(CString Str, CString SubStr, int LCN)
{
    if(Str.Find("E")!=-1) //если среди терминалов есть E
        return 1;        //вернуть 1
    if(atoi(SubStr)==(LCN+1) && Str.Find("literal")!=-1)//если номер соответствует значению
        return 1;        //вернуть 1
    if(atoi(SubStr)==(LCN+2) && Str.Find("name")!=-1)//если номер соответствует имени
        return 1;        //вернуть 1
    if(Str.Find(SubStr+",")!=-1) //если номер присутствует внутри строки терминалов
        return 1;        //вернуть 1
    if(SubStr==Str.Right(SubStr.GetLength()))//если номер присутствует в конце строки терминалов
        return 1;        //вернуть 1
    return 0;            //вернуть 0
}
```

```
//Функция формирования файла отчета
//возвращаемый параметр: нет
//передаваемые параметры:
//SyntReport - строка, содержащая порядок переходов в ходе разбора
//AddStr - строка с итогом отчета
void FormReport(CString SyntReport, CString AddStr)
{
    CFile f;          //объект класса работы с файлами
    //если файл можно создать или открыть для записи
    if(f.Open("Report.txt",CFile::modeCreate | CFile::modeWrite))
    {
        //выполнить запись в файл
        f.Write(SyntReport+AddStr,SyntReport.GetLength()+AddStr.GetLength());
        f.Close();    //закрыть файл
    }
    else    //если файл не удается создать или открыть
        AfxMessageBox("Can't form Report file!");
}
```

Приложение Г – Ход синтаксического анализа входной программы языка MATH («Report.txt»)

```
'1-1' - |001|1-1|2|0|0|0|None|
'1-1' - |002|1-1|3|1|0|0|1|Error 2|
'6-1' - |003|name,E|8|0|4|0|1|Error 3|
'6-1' - |008|name|10|0|0|0|0|None|
'6-1' - |010|name|0|1|0|1|1|Error 10|
'1-2' - |004|1-2|5|1|0|0|1|Error 4|
'1-8' - |005|1-8,1-9,name,1-6,1-7,1-4,E|12|0|6|0|1|Error 5|
'1-8' - |012|1-8,1-9|15|0|0|0|0|None|
'1-8' - |015|1-8,1-9|22|0|16|0|1|Error 15|
'1-8' - |022|1-8|24|0|0|0|0|None|
'1-8' - |024|1-8|0|1|0|1|1|Error 24|
'6-2' - |016|name|27|0|17|0|1|Error 16|
'6-2' - |027|name|28|1|0|0|1|Error 27|
'3-3' - |028|3-3,E|29|0|0|0|1|Error 28|
'3-3' - |029|3-3|31|0|0|0|0|None|
'3-3' - |031|3-3|32|1|0|0|1|Error 31|
'6-3' - |032|name|26|0|0|0|1|Error 32|
'6-3' - |026|name|27|0|0|0|1|Error 26|
'6-3' - |027|name|28|1|0|0|1|Error 27|
'3-3' - |028|3-3,E|29|0|0|0|1|Error 28|
'3-3' - |029|3-3|31|0|0|0|0|None|
'3-3' - |031|3-3|32|1|0|0|1|Error 31|
'6-4' - |032|name|26|0|0|0|1|Error 32|
'6-4' - |026|name|27|0|0|0|1|Error 26|
'6-4' - |027|name|28|1|0|0|1|Error 27|
'3-4' - |028|3-3,E|29|0|0|0|1|Error 28|
'3-4' - |029|3-3|31|0|0|0|0|None|
'3-4' - |030|E|33|0|0|0|1|Error 30|
'3-4' - |033|E|0|0|0|1|1|Error 33|
'3-4' - |017|3-4|18|1|0|0|1|Error 17|
'1-9' - |018|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 18|
'1-9' - |012|1-8,1-9|15|0|0|0|0|None|
'1-9' - |015|1-8,1-9|22|0|16|0|1|Error 15|
'1-9' - |022|1-8|24|0|0|0|0|None|
'1-9' - |023|1-9|25|0|0|0|1|Error 23|
'1-9' - |025|1-9|0|1|0|1|1|Error 25|
'6-5' - |016|name|27|0|17|0|1|Error 16|
'6-5' - |027|name|28|1|0|0|1|Error 27|
'3-4' - |028|3-3,E|29|0|0|0|1|Error 28|
'3-4' - |029|3-3|31|0|0|0|0|None|
'3-4' - |030|E|33|0|0|0|1|Error 30|
'3-4' - |033|E|0|0|0|1|1|Error 33|
'3-4' - |017|3-4|18|1|0|0|1|Error 17|
'1-6' - |018|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 18|
'1-6' - |012|1-8,1-9|15|0|0|0|0|None|
'1-6' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
'1-6' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
'1-6' - |034|name|38|0|0|0|0|None|
'1-6' - |035|1-6|43|0|0|0|0|None|
```

'1-6' - |043|1-6|44|1|0|0|1|Error 43|
 '4-1' - |044|name,literal,4-1|98|0|45|0|1|Error 44|
 '4-1' - |098|name,literal|100|0|0|0|0|None|
 '4-1' - |099|4-1|101|0|0|0|1|Error 99|
 '4-1' - |101|4-1|102|1|0|0|1|Error 101|
 '5-1' - |102|literal|103|1|0|0|1|Error 102|
 '4-1' - |103|4-1|0|1|0|1|1|Error 103|
 '3-4' - |045|3-4|46|1|0|0|1|Error 45|
 '1-7' - |046|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 46|
 '1-7' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-7' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-7' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-7' - |034|name|38|0|0|0|0|None|
 '1-7' - |035|1-6|43|0|0|0|0|None|
 '1-7' - |036|1-7|47|0|0|0|0|None|
 '1-7' - |047|1-7|48|1|0|0|1|Error 47|
 '6-2' - |048|name|49|1|0|0|1|Error 48|
 '3-4' - |049|3-4|50|1|0|0|1|Error 49|
 '1-6' - |050|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 50|
 '1-6' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-6' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-6' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-6' - |034|name|38|0|0|0|0|None|
 '1-6' - |035|1-6|43|0|0|0|0|None|
 '1-6' - |043|1-6|44|1|0|0|1|Error 43|
 '4-1' - |044|name,literal,4-1|98|0|45|0|1|Error 44|
 '4-1' - |098|name,literal|100|0|0|0|0|None|
 '4-1' - |099|4-1|101|0|0|0|1|Error 99|
 '4-1' - |101|4-1|102|1|0|0|1|Error 101|
 '5-2' - |102|literal|103|1|0|0|1|Error 102|
 '4-1' - |103|4-1|0|1|0|1|1|Error 103|
 '3-4' - |045|3-4|46|1|0|0|1|Error 45|
 '1-7' - |046|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 46|
 '1-7' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-7' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-7' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-7' - |034|name|38|0|0|0|0|None|
 '1-7' - |035|1-6|43|0|0|0|0|None|
 '1-7' - |036|1-7|47|0|0|0|0|None|
 '1-7' - |047|1-7|48|1|0|0|1|Error 47|
 '6-3' - |048|name|49|1|0|0|1|Error 48|
 '3-4' - |049|3-4|50|1|0|0|1|Error 49|
 '1-6' - |050|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 50|
 '1-6' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-6' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-6' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-6' - |034|name|38|0|0|0|0|None|
 '1-6' - |035|1-6|43|0|0|0|0|None|
 '1-6' - |043|1-6|44|1|0|0|1|Error 43|
 '4-1' - |044|name,literal,4-1|98|0|45|0|1|Error 44|
 '4-1' - |098|name,literal|100|0|0|0|0|None|
 '4-1' - |099|4-1|101|0|0|0|1|Error 99|

'4-1' - |101|4-1|102|1|0|0|1|Error 101|
 '5-3' - |102|literal|103|1|0|0|1|Error 102|
 '4-1' - |103|4-1|0|1|0|1|1|Error 103|
 '3-4' - |045|3-4|46|1|0|0|1|Error 45|
 '1-7' - |046|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 46|
 '1-7' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-7' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-7' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-7' - |034|name|38|0|0|0|0|None|
 '1-7' - |035|1-6|43|0|0|0|0|None|
 '1-7' - |036|1-7|47|0|0|0|0|None|
 '1-7' - |047|1-7|48|1|0|0|1|Error 47|
 '6-4' - |048|name|49|1|0|0|1|Error 48|
 '3-4' - |049|3-4|50|1|0|0|1|Error 49|
 '1-4' - |050|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 50|
 '1-4' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-4' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-4' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-4' - |034|name|38|0|0|0|0|None|
 '1-4' - |035|1-6|43|0|0|0|0|None|
 '1-4' - |036|1-7|47|0|0|0|0|None|
 '1-4' - |037|1-4|51|0|0|0|0|None|
 '1-4' - |051|1-4|52|1|0|0|1|Error 51|
 '3-1' - |052|3-1|53|1|0|0|1|Error 52|
 '6-4' - |053|name|104|0|54|0|1|Error 53|
 '6-4' - |104|name|105|0|0|0|1|Error 104|
 '6-4' - |105|name|106|1|0|0|1|Error 105|
 '2-10' - |106|2-6,2-5,2-8,2-7,2-9,2-10|82|0|107|0|1|Error 106|
 '2-10' - |082|2-6|88|0|0|0|0|None|
 '2-10' - |083|2-5|89|0|0|0|0|None|
 '2-10' - |084|2-8|90|0|0|0|0|None|
 '2-10' - |085|2-7|91|0|0|0|0|None|
 '2-10' - |086|2-9|92|0|0|0|0|None|
 '2-10' - |087|2-10|93|0|0|0|1|Error 87|
 '2-10' - |093|2-10|0|1|0|1|1|Error 93|
 '5-4' - |107|name,literal|94|0|0|0|1|Error 107|
 '5-4' - |094|name|96|0|0|0|0|None|
 '5-4' - |095|literal|97|0|0|0|1|Error 95|
 '5-4' - |097|literal|0|1|0|1|1|Error 97|
 '3-2' - |054|3-3,E|108|0|55|0|1|Error 54|
 '3-2' - |108|3-3|110|0|0|0|0|Error 108|
 '3-2' - |109|E|112|0|0|0|1|Error 109|
 '3-2' - |112|E|0|0|0|1|1|Error 112|
 '3-2' - |055|3-2|56|1|0|0|1|Error 55|
 '6-5' - |056|1-8,1-9,name,1-6,1-7,1-4,E|12|0|57|0|1|Error 56|
 '6-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '6-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '6-5' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '6-5' - |034|name|38|0|0|0|0|None|
 '6-5' - |038|name|39|1|0|0|1|Error 38|
 '2-11' - |039|2-11|40|1|0|0|1|Error 39|
 '6-2' - |040|name,literal|59|0|41|0|1|Error 40|

'6-2' - |059|name|61|0|0|0|0|None|
 '6-2' - |061|name|62|1|0|0|1|Error 61|
 '2-3' - |062|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10,E|65|0|0|0|1|Error 62|
 '2-3' - |065|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10|67|0|0|0|0|None|
 '2-3' - |067|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10|70|0|68|0|1|Error 67|
 '2-3' - |070|2-1,2-2,2-3,2-4|72|0|0|0|0|None|
 '2-3' - |072|2-1,2-2,2-3,2-4|74|0|0|0|1|Error 72|
 '2-3' - |074|2-1|78|0|0|0|0|None|
 '2-3' - |075|2-2|79|0|0|0|0|None|
 '2-3' - |076|2-3|80|0|0|0|0|None|
 '2-3' - |080|2-3|0|1|0|1|1|Error 80|
 '6-3' - |068|name,literal|94|0|0|1|1|Error 68|
 '6-3' - |094|name|96|0|0|0|0|None|
 '6-3' - |096|name|0|1|0|1|1|Error 96|
 '3-4' - |041|3-4|42|1|0|0|1|Error 41|
 '1-5' - |042|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 42|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |057|1-5|58|1|0|0|1|Error 57|
 '1-4' - |058|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 58|
 '1-4' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-4' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-4' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-4' - |034|name|38|0|0|0|0|None|
 '1-4' - |035|1-6|43|0|0|0|0|None|
 '1-4' - |036|1-7|47|0|0|0|0|None|
 '1-4' - |037|1-4|51|0|0|0|0|None|
 '1-4' - |051|1-4|52|1|0|0|1|Error 51|
 '3-1' - |052|3-1|53|1|0|0|1|Error 52|
 '6-4' - |053|name|104|0|54|0|1|Error 53|
 '6-4' - |104|name|105|0|0|0|1|Error 104|
 '6-4' - |105|name|106|1|0|0|1|Error 105|
 '2-10' - |106|2-6,2-5,2-8,2-7,2-9,2-10|82|0|107|0|1|Error 106|
 '2-10' - |082|2-6|88|0|0|0|0|None|
 '2-10' - |083|2-5|89|0|0|0|0|None|
 '2-10' - |084|2-8|90|0|0|0|0|None|
 '2-10' - |085|2-7|91|0|0|0|0|None|
 '2-10' - |086|2-9|92|0|0|0|0|None|
 '2-10' - |087|2-10|93|0|0|0|1|Error 87|
 '2-10' - |093|2-10|0|1|0|1|1|Error 93|
 '5-5' - |107|name,literal|94|0|0|0|1|Error 107|
 '5-5' - |094|name|96|0|0|0|0|None|
 '5-5' - |095|literal|97|0|0|0|1|Error 95|
 '5-5' - |097|literal|0|1|0|1|1|Error 97|
 '3-2' - |054|3-3,E|108|0|55|0|1|Error 54|

'3-2' - |108|3-3|110|0|0|0|0|Error 108|
 '3-2' - |109|E|112|0|0|0|1|Error 109|
 '3-2' - |112|E|0|0|0|1|1|Error 112|
 '3-2' - |055|3-2|56|1|0|0|1|Error 55|
 '1-4' - |056|1-8,1-9,name,1-6,1-7,1-4,E|12|0|57|0|1|Error 56|
 '1-4' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-4' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-4' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-4' - |034|name|38|0|0|0|0|None|
 '1-4' - |035|1-6|43|0|0|0|0|None|
 '1-4' - |036|1-7|47|0|0|0|0|None|
 '1-4' - |037|1-4|51|0|0|0|0|None|
 '1-4' - |051|1-4|52|1|0|0|1|Error 51|
 '3-1' - |052|3-1|53|1|0|0|1|Error 52|
 '6-3' - |053|name|104|0|54|0|1|Error 53|
 '6-3' - |104|name|105|0|0|0|1|Error 104|
 '6-3' - |105|name|106|1|0|0|1|Error 105|
 '2-9' - |106|2-6,2-5,2-8,2-7,2-9,2-10|82|0|107|0|1|Error 106|
 '2-9' - |082|2-6|88|0|0|0|0|None|
 '2-9' - |083|2-5|89|0|0|0|0|None|
 '2-9' - |084|2-8|90|0|0|0|0|None|
 '2-9' - |085|2-7|91|0|0|0|0|None|
 '2-9' - |086|2-9|92|0|0|0|0|None|
 '2-9' - |092|2-9|0|1|0|1|1|Error 92|
 '5-4' - |107|name,literal|94|0|0|0|1|Error 107|
 '5-4' - |094|name|96|0|0|0|0|None|
 '5-4' - |095|literal|97|0|0|0|1|Error 95|
 '5-4' - |097|literal|0|1|0|1|1|Error 97|
 '3-2' - |054|3-3,E|108|0|55|0|1|Error 54|
 '3-2' - |108|3-3|110|0|0|0|0|Error 108|
 '3-2' - |109|E|112|0|0|0|1|Error 109|
 '3-2' - |112|E|0|0|0|1|1|Error 112|
 '3-2' - |055|3-2|56|1|0|0|1|Error 55|
 '6-5' - |056|1-8,1-9,name,1-6,1-7,1-4,E|12|0|57|0|1|Error 56|
 '6-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '6-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '6-5' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '6-5' - |034|name|38|0|0|0|0|None|
 '6-5' - |038|name|39|1|0|0|1|Error 38|
 '2-11' - |039|2-11|40|1|0|0|1|Error 39|
 '6-2' - |040|name,literal|59|0|41|0|1|Error 40|
 '6-2' - |059|name|61|0|0|0|0|None|
 '6-2' - |061|name|62|1|0|0|1|Error 61|
 '2-4' - |062|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10,E|65|0|0|0|1|Error 62|
 '2-4' - |065|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10|67|0|0|0|0|None|
 '2-4' - |067|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10|70|0|68|0|1|Error 67|
 '2-4' - |070|2-1,2-2,2-3,2-4|72|0|0|0|0|None|
 '2-4' - |072|2-1,2-2,2-3,2-4|74|0|0|0|1|Error 72|
 '2-4' - |074|2-1|78|0|0|0|0|None|
 '2-4' - |075|2-2|79|0|0|0|0|None|
 '2-4' - |076|2-3|80|0|0|0|0|None|
 '2-4' - |077|2-4|81|0|0|0|1|Error 77|

'2-4' - |081|2-4|0|1|0|1|1|Error 81|
 '6-3' - |068|name,literal|94|0|0|1|1|Error 68|
 '6-3' - |094|name|96|0|0|0|0|None|
 '6-3' - |096|name|0|1|0|1|1|Error 96|
 '3-4' - |041|3-4|42|1|0|0|1|Error 41|
 '1-5' - |042|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 42|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |057|1-5|58|1|0|0|1|Error 57|
 '1-4' - |058|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 58|
 '1-4' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-4' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-4' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-4' - |034|name|38|0|0|0|0|None|
 '1-4' - |035|1-6|43|0|0|0|0|None|
 '1-4' - |036|1-7|47|0|0|0|0|None|
 '1-4' - |037|1-4|51|0|0|0|0|None|
 '1-4' - |051|1-4|52|1|0|0|1|Error 51|
 '3-1' - |052|3-1|53|1|0|0|1|Error 52|
 '6-3' - |053|name|104|0|54|0|1|Error 53|
 '6-3' - |104|name|105|0|0|0|1|Error 104|
 '6-3' - |105|name|106|1|0|0|1|Error 105|
 '2-10' - |106|2-6,2-5,2-8,2-7,2-9,2-10|82|0|107|0|1|Error 106|
 '2-10' - |082|2-6|88|0|0|0|0|None|
 '2-10' - |083|2-5|89|0|0|0|0|None|
 '2-10' - |084|2-8|90|0|0|0|0|None|
 '2-10' - |085|2-7|91|0|0|0|0|None|
 '2-10' - |086|2-9|92|0|0|0|0|None|
 '2-10' - |087|2-10|93|0|0|0|1|Error 87|
 '2-10' - |093|2-10|0|1|0|1|1|Error 93|
 '5-4' - |107|name,literal|94|0|0|0|1|Error 107|
 '5-4' - |094|name|96|0|0|0|0|None|
 '5-4' - |095|literal|97|0|0|0|1|Error 95|
 '5-4' - |097|literal|0|1|0|1|1|Error 97|
 '3-2' - |054|3-3,E|108|0|55|0|1|Error 54|
 '3-2' - |108|3-3|110|0|0|0|0|Error 108|
 '3-2' - |109|E|112|0|0|0|1|Error 109|
 '3-2' - |112|E|0|0|0|1|1|Error 112|
 '3-2' - |055|3-2|56|1|0|0|1|Error 55|
 '1-6' - |056|1-8,1-9,name,1-6,1-7,1-4,E|12|0|57|0|1|Error 56|
 '1-6' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-6' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-6' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-6' - |034|name|38|0|0|0|0|None|
 '1-6' - |035|1-6|43|0|0|0|0|None|

'1-6' - |043|1-6|44|1|0|0|1|Error 43|
 '4-1' - |044|name,literal,4-1|98|0|45|0|1|Error 44|
 '4-1' - |098|name,literal|100|0|0|0|0|None|
 '4-1' - |099|4-1|101|0|0|0|1|Error 99|
 '4-1' - |101|4-1|102|1|0|0|1|Error 101|
 '5-6' - |102|literal|103|1|0|0|1|Error 102|
 '4-1' - |103|4-1|0|1|0|1|1|Error 103|
 '3-4' - |045|3-4|46|1|0|0|1|Error 45|
 '1-5' - |046|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 46|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |057|1-5|58|1|0|0|1|Error 57|
 '1-5' - |058|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 58|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |057|1-5|58|1|0|0|1|Error 57|
 '1-4' - |058|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 58|
 '1-4' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-4' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-4' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-4' - |034|name|38|0|0|0|0|None|
 '1-4' - |035|1-6|43|0|0|0|0|None|
 '1-4' - |036|1-7|47|0|0|0|0|None|
 '1-4' - |037|1-4|51|0|0|0|0|None|
 '1-4' - |051|1-4|52|1|0|0|1|Error 51|
 '3-1' - |052|3-1|53|1|0|0|1|Error 52|
 '6-4' - |053|name|104|0|54|0|1|Error 53|
 '6-4' - |104|name|105|0|0|0|1|Error 104|
 '6-4' - |105|name|106|1|0|0|1|Error 105|
 '2-10' - |106|2-6,2-5,2-8,2-7,2-9,2-10|82|0|107|0|1|Error 106|
 '2-10' - |082|2-6|88|0|0|0|0|None|
 '2-10' - |083|2-5|89|0|0|0|0|None|
 '2-10' - |084|2-8|90|0|0|0|0|None|

'2-10' - |085|2-7|91|0|0|0|None|
 '2-10' - |086|2-9|92|0|0|0|None|
 '2-10' - |087|2-10|93|0|0|0|1|Error 87|
 '2-10' - |093|2-10|0|1|0|1|1|Error 93|
 '5-7' - |107|name,literal|94|0|0|0|1|Error 107|
 '5-7' - |094|name|96|0|0|0|0|None|
 '5-7' - |095|literal|97|0|0|0|1|Error 95|
 '5-7' - |097|literal|0|1|0|1|1|Error 97|
 '3-2' - |054|3-3,E|108|0|55|0|1|Error 54|
 '3-2' - |108|3-3|110|0|0|0|0|Error 108|
 '3-2' - |109|E|112|0|0|0|1|Error 109|
 '3-2' - |112|E|0|0|0|1|1|Error 112|
 '3-2' - |055|3-2|56|1|0|0|1|Error 55|
 '6-5' - |056|1-8,1-9,name,1-6,1-7,1-4,E|12|0|57|0|1|Error 56|
 '6-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '6-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '6-5' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '6-5' - |034|name|38|0|0|0|0|None|
 '6-5' - |038|name|39|1|0|0|1|Error 38|
 '2-11' - |039|2-11|40|1|0|0|1|Error 39|
 '6-2' - |040|name,literal|59|0|41|0|1|Error 40|
 '6-2' - |059|name|61|0|0|0|0|None|
 '6-2' - |061|name|62|1|0|0|1|Error 61|
 '2-1' - |062|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10,E|65|0|0|0|1|Error 62|
 '2-1' - |065|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10|67|0|0|0|0|None|
 '2-1' - |067|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10|70|0|68|0|1|Error 67|
 '2-1' - |070|2-1,2-2,2-3,2-4|72|0|0|0|0|None|
 '2-1' - |072|2-1,2-2,2-3,2-4|74|0|0|0|1|Error 72|
 '2-1' - |074|2-1|78|0|0|0|0|None|
 '2-1' - |078|2-1|0|1|0|1|1|Error 78|
 '6-3' - |068|name,literal|94|0|0|1|1|Error 68|
 '6-3' - |094|name|96|0|0|0|0|None|
 '6-3' - |096|name|0|1|0|1|1|Error 96|
 '3-4' - |041|3-4|42|1|0|0|1|Error 41|
 '1-5' - |042|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 42|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |057|1-5|58|1|0|0|1|Error 57|
 '1-4' - |058|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 58|
 '1-4' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-4' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-4' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-4' - |034|name|38|0|0|0|0|None|
 '1-4' - |035|1-6|43|0|0|0|0|None|
 '1-4' - |036|1-7|47|0|0|0|0|None|

'1-4' - |037|1-4|51|0|0|0|None|
 '1-4' - |051|1-4|52|1|0|0|1|Error 51|
 '3-1' - |052|3-1|53|1|0|0|1|Error 52|
 '6-4' - |053|name|104|0|54|0|1|Error 53|
 '6-4' - |104|name|105|0|0|0|1|Error 104|
 '6-4' - |105|name|106|1|0|0|1|Error 105|
 '2-10' - |106|2-6,2-5,2-8,2-7,2-9,2-10|82|0|107|0|1|Error 106|
 '2-10' - |082|2-6|88|0|0|0|0|None|
 '2-10' - |083|2-5|89|0|0|0|0|None|
 '2-10' - |084|2-8|90|0|0|0|0|None|
 '2-10' - |085|2-7|91|0|0|0|0|None|
 '2-10' - |086|2-9|92|0|0|0|0|None|
 '2-10' - |087|2-10|93|0|0|0|1|Error 87|
 '2-10' - |093|2-10|0|1|0|1|1|Error 93|
 '5-8' - |107|name,literal|94|0|0|0|1|Error 107|
 '5-8' - |094|name|96|0|0|0|0|None|
 '5-8' - |095|literal|97|0|0|0|1|Error 95|
 '5-8' - |097|literal|0|1|0|1|1|Error 97|
 '3-2' - |054|3-3,E|108|0|55|0|1|Error 54|
 '3-2' - |108|3-3|110|0|0|0|0|Error 108|
 '3-2' - |109|E|112|0|0|0|1|Error 109|
 '3-2' - |112|E|0|0|0|1|1|Error 112|
 '3-2' - |055|3-2|56|1|0|0|1|Error 55|
 '6-5' - |056|1-8,1-9,name,1-6,1-7,1-4,E|12|0|57|0|1|Error 56|
 '6-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '6-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '6-5' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '6-5' - |034|name|38|0|0|0|0|None|
 '6-5' - |038|name|39|1|0|0|1|Error 38|
 '2-11' - |039|2-11|40|1|0|0|1|Error 39|
 '6-2' - |040|name,literal|59|0|41|0|1|Error 40|
 '6-2' - |059|name|61|0|0|0|0|None|
 '6-2' - |061|name|62|1|0|0|1|Error 61|
 '2-2' - |062|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10,E|65|0|0|0|1|Error 62|
 '2-2' - |065|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10|67|0|0|0|0|None|
 '2-2' - |067|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10|70|0|68|0|1|Error 67|
 '2-2' - |070|2-1,2-2,2-3,2-4|72|0|0|0|0|None|
 '2-2' - |072|2-1,2-2,2-3,2-4|74|0|0|0|1|Error 72|
 '2-2' - |074|2-1|78|0|0|0|0|None|
 '2-2' - |075|2-2|79|0|0|0|0|None|
 '2-2' - |079|2-2|0|1|0|1|1|Error 79|
 '6-3' - |068|name,literal|94|0|0|1|1|Error 68|
 '6-3' - |094|name|96|0|0|0|0|None|
 '6-3' - |096|name|0|1|0|1|1|Error 96|
 '3-4' - |041|3-4|42|1|0|0|1|Error 41|
 '1-5' - |042|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 42|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|

'1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |057|1-5|58|1|0|0|1|Error 57|
 '1-4' - |058|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 58|
 '1-4' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-4' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-4' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-4' - |034|name|38|0|0|0|0|None|
 '1-4' - |035|1-6|43|0|0|0|0|None|
 '1-4' - |036|1-7|47|0|0|0|0|None|
 '1-4' - |037|1-4|51|0|0|0|0|None|
 '1-4' - |051|1-4|52|1|0|0|1|Error 51|
 '3-1' - |052|3-1|53|1|0|0|1|Error 52|
 '6-4' - |053|name|104|0|54|0|1|Error 53|
 '6-4' - |104|name|105|0|0|0|1|Error 104|
 '6-4' - |105|name|106|1|0|0|1|Error 105|
 '2-5' - |106|2-6,2-5,2-8,2-7,2-9,2-10|82|0|107|0|1|Error 106|
 '2-5' - |082|2-6|88|0|0|0|0|None|
 '2-5' - |083|2-5|89|0|0|0|0|None|
 '2-5' - |089|2-5|0|1|0|1|1|Error 89|
 '5-4' - |107|name,literal|94|0|0|0|1|Error 107|
 '5-4' - |094|name|96|0|0|0|0|None|
 '5-4' - |095|literal|97|0|0|0|1|Error 95|
 '5-4' - |097|literal|0|1|0|1|1|Error 97|
 '3-3' - |054|3-3,E|108|0|55|0|1|Error 54|
 '3-3' - |108|3-3|110|0|0|0|0|Error 108|
 '3-3' - |110|3-3|111|1|0|0|1|Error 110|
 '6-4' - |111|name|104|0|0|0|1|Error 111|
 '6-4' - |104|name|105|0|0|0|1|Error 104|
 '6-4' - |105|name|106|1|0|0|1|Error 105|
 '2-6' - |106|2-6,2-5,2-8,2-7,2-9,2-10|82|0|107|0|1|Error 106|
 '2-6' - |082|2-6|88|0|0|0|0|None|
 '2-6' - |088|2-6|0|1|0|1|1|Error 88|
 '5-8' - |107|name,literal|94|0|0|0|1|Error 107|
 '5-8' - |094|name|96|0|0|0|0|None|
 '5-8' - |095|literal|97|0|0|0|1|Error 95|
 '5-8' - |097|literal|0|1|0|1|1|Error 97|
 '3-2' - |055|3-2|56|1|0|0|1|Error 55|
 '1-6' - |056|1-8,1-9,name,1-6,1-7,1-4,E|12|0|57|0|1|Error 56|
 '1-6' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-6' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-6' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-6' - |034|name|38|0|0|0|0|None|
 '1-6' - |035|1-6|43|0|0|0|0|None|
 '1-6' - |043|1-6|44|1|0|0|1|Error 43|
 '4-1' - |044|name,literal,4-1|98|0|45|0|1|Error 44|
 '4-1' - |098|name,literal|100|0|0|0|0|None|
 '4-1' - |099|4-1|101|0|0|0|1|Error 99|
 '4-1' - |101|4-1|102|1|0|0|1|Error 101|
 '5-9' - |102|literal|103|1|0|0|1|Error 102|
 '4-1' - |103|4-1|0|1|0|1|1|Error 103|

'3-4' - |045|3-4|46|1|0|0|1|Error 45|
 '1-5' - |046|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 46|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-5' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-5' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-5' - |014|E|21|0|0|0|1|Error 14|
 '1-5' - |021|E|0|0|0|1|1|Error 21|
 '1-5' - |057|1-5|58|1|0|0|1|Error 57|
 '1-6' - |058|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 58|
 '1-6' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-6' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-6' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-6' - |034|name|38|0|0|0|0|None|
 '1-6' - |035|1-6|43|0|0|0|0|None|
 '1-6' - |043|1-6|44|1|0|0|1|Error 43|
 '4-1' - |044|name,literal,4-1|98|0|45|0|1|Error 44|
 '4-1' - |098|name,literal|100|0|0|0|0|None|
 '4-1' - |099|4-1|101|0|0|0|1|Error 99|
 '4-1' - |101|4-1|102|1|0|0|1|Error 101|
 '5-10' - |102|literal|103|1|0|0|1|Error 102|
 '4-1' - |103|4-1|0|1|0|1|1|Error 103|
 '3-4' - |045|3-4|46|1|0|0|1|Error 45|
 '1-6' - |046|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 46|
 '1-6' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-6' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-6' - |019|name,1-6,1-7,1-4|34|0|20|0|1|Error 19|
 '1-6' - |034|name|38|0|0|0|0|None|
 '1-6' - |035|1-6|43|0|0|0|0|None|
 '1-6' - |043|1-6|44|1|0|0|1|Error 43|
 '6-5' - |044|name,literal,4-1|98|0|45|0|1|Error 44|
 '6-5' - |098|name,literal|100|0|0|0|0|None|
 '6-5' - |100|name,literal|59|0|0|0|1|Error 100|
 '6-5' - |059|name|61|0|0|0|0|None|
 '6-5' - |061|name|62|1|0|0|1|Error 61|
 '3-4' - |062|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10,E|65|0|0|0|1|Error 62|
 '3-4' - |065|2-1,2-2,2-3,2-4,2-6,2-5,2-8,2-7,2-9,2-10|67|0|0|0|0|None|
 '3-4' - |066|E|69|0|0|0|1|Error 66|
 '3-4' - |069|E|0|0|0|1|1|Error 69|
 '3-4' - |045|3-4|46|1|0|0|1|Error 45|
 '1-3' - |046|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 46|
 '1-3' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-3' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-3' - |014|E|21|0|0|0|1|Error 14|
 '1-3' - |021|E|0|0|0|1|1|Error 21|
 '1-3' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-3' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-3' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-3' - |014|E|21|0|0|0|1|Error 14|

'1-3' - |021|E|0|0|0|1|1|Error 21|
 '1-3' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-3' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-3' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-3' - |014|E|21|0|0|0|1|Error 14|
 '1-3' - |021|E|0|0|0|1|1|Error 21|
 '1-3' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-3' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-3' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-3' - |014|E|21|0|0|0|1|Error 14|
 '1-3' - |021|E|0|0|0|1|1|Error 21|
 '1-3' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-3' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-3' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-3' - |014|E|21|0|0|0|1|Error 14|
 '1-3' - |021|E|0|0|0|1|1|Error 21|
 '1-3' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-3' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-3' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-3' - |014|E|21|0|0|0|1|Error 14|
 '1-3' - |021|E|0|0|0|1|1|Error 21|
 '1-3' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-3' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-3' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-3' - |014|E|21|0|0|0|1|Error 14|
 '1-3' - |021|E|0|0|0|1|1|Error 21|
 '1-3' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-3' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-3' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-3' - |014|E|21|0|0|0|1|Error 14|
 '1-3' - |021|E|0|0|0|1|1|Error 21|
 '1-3' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-3' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-3' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-3' - |014|E|21|0|0|0|1|Error 14|
 '1-3' - |021|E|0|0|0|1|1|Error 21|
 '1-3' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
 '1-3' - |012|1-8,1-9|15|0|0|0|0|None|
 '1-3' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
 '1-3' - |014|E|21|0|0|0|1|Error 14|
 '1-3' - |021|E|0|0|0|1|1|Error 21|
 '1-3' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|

'1-3' - |012|1-8,1-9|15|0|0|0|0|None|
'1-3' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
'1-3' - |014|E|21|0|0|0|1|Error 14|
'1-3' - |021|E|0|0|0|1|1|Error 21|
'1-3' - |020|1-8,1-9,name,1-6,1-7,1-4,E|12|0|0|0|1|Error 20|
'1-3' - |012|1-8,1-9|15|0|0|0|0|None|
'1-3' - |013|name,1-6,1-7,1-4|19|0|0|0|0|Error 13|
'1-3' - |014|E|21|0|0|0|1|Error 14|
'1-3' - |021|E|0|0|0|1|1|Error 21|
'1-3' - |006|1-3|7|1|0|0|1|Error 6|
No Errors