一个小型编译器的设计与实现

目录

1 课程目的和要求

在该课程中，学生将综合利用汇编语言、高级程序设计语言、数据结构与算法、计算机原理、操作系统、软件工程、编译原理等课程中学到的原理、技术和方法，设计并实现一个将高级程序设计语言编写的源程序翻译为可执行文件的较为完整的小型编译器。该课程将培养学生设计复杂大型软件系统的能力；培养学生应用知识解决问题的能力；培养学生的实践动手能力；培养学生的系统分析与设计能力；培养学生的工程素质。

1.1 编译程序的基本功能

① 词法分析
② 语法分析
③ 语义分析和汇编代码生成
④ 符号表管理和存储分配

1.2、编译程序所处理的源程序的主要成分

① 声明语句，包括变量声明和函数声明；
② 赋值语句；
③ 布尔表达式；三种典型的控制流语句（顺序，循环、分支）；
④ 一维数组的声明和引用；
⑤ 输入、输出语句；
⑥ 算术表达式的运算包括加、减、乘、除；
⑦ 函数调用语句。

1.3、最终功能要求

编译程序首先能将源程序生成汇编代码，再经过汇编器的编译生成可执行程序，要求生成的可执行程序能够正确运行。

2 源语言的形式描述

2.1 词法规则的形式描述：

2.1.1 词法需要识别的单词

ID INT FLOAT DIGIT STR IF ELSE FOR RETURN + - * / > < = >= <= DOT SEM ( ) [ ] { } AND INC

2.1.2 形式化描述

ID → letter* ; DIGIT → digit* ; STR → “ (letter)* “ ; INT → i n t ; FOR → f o r ; FLOAT → f l o a t ; ELSE → e l s e ; RETURN → return ; + → + ; INC → ++ ; - → - ; * → * ; / → / ; > → > ; >= → > = ; < → < ; <= → <= ; = → = ; == → = = ; , → , ; → ; ; (→ (;) →) ; [→ [;] →] ; { → { ; } → } ; & → & ;

2.2 语法规则的形式化描述：

2.2.1 非终结符

p expression expression_list type declarator declarator_list declarator_e digit_list assignment_e add_e multi_e cast_e select_e iterator_e func_e bool_e inc_e

2.2.2 终结符

ID INT FLOAT DIGIT STR IF ELSE FOR RETURN + - * / > < = >= <= DOT SEM ( ) [ ] { } AND INC

2.2.3 文法产生式

p' => p;
p => INT ID ( ) { expression_list };
expression_list => expression | expression_list expression;
type => INT | FLOAT;
declarator => ID | ID [DIGIT] ;
declarator_list => declarator | declarator_list DOT declarator;
declarator_e => type declarator_list SEM | type declarator = { digit_list SEM;
digit_list => DIGIT | digit_list DOT DIGIT;
assignment_e => declarator = add_e SEM;
add_e => multi_e | add_e + multi_e | add_e - multi_e;
multi_e => cast_e | multi_e * cast_e | multi_e / cast_e;
cast_e => declarator | DIGIT | ID [ID];
inc_e => ID INC;
expression => declarator_e | assignment_e | select_e | iterator_e | func_e | RETURN DIGIT SEM;
bool_e => cast_e < cast_e | cast_e > cast_e | cast_e <= cast_e | cast_e >= cast_e;
select_e => IF (bool_e) { expression_list } ELSE { expression_list };
iterator_e => FOR (assignment_e bool_e SEM inc_e) { expression_list };
func_e => ID (STR DOT declarator_list) SEM | ID (STR) SEM | ID (STR DOT AND ID) SEM;

3 词法分析器的设计与实现

3.1 词法分析器的总体结构

迭代地从源文件中读取每一行，根据每一行调用词法识别程序。对该行的字符流进行单词识别。词法识别程序根据词法识别的状态转换图来设计实现程序根据当前的一个全局状态值 state，查看接下来到来的字符，选择识别该单词，或者进入下一个状态。如果没有任何一种状态可以转移，则调出出错处理程序，打印出错字符附近的行数。对于已经识别的单词，将其组成一个元组。左边是名子，右边是实际单词。存储在一个全局列表里面，供语法分析时候使用。

3.2 数据结构设计

主要使用了 python 中的列表和元组的数据结构。其中元组结构用于绑定符号表中的名字和十实际的值。列表结构用于存储识别后的单词序列。里面的每一个元素是一个元组。包含名字和识别出来的单词。

3.2.1 词法分析中的符号表

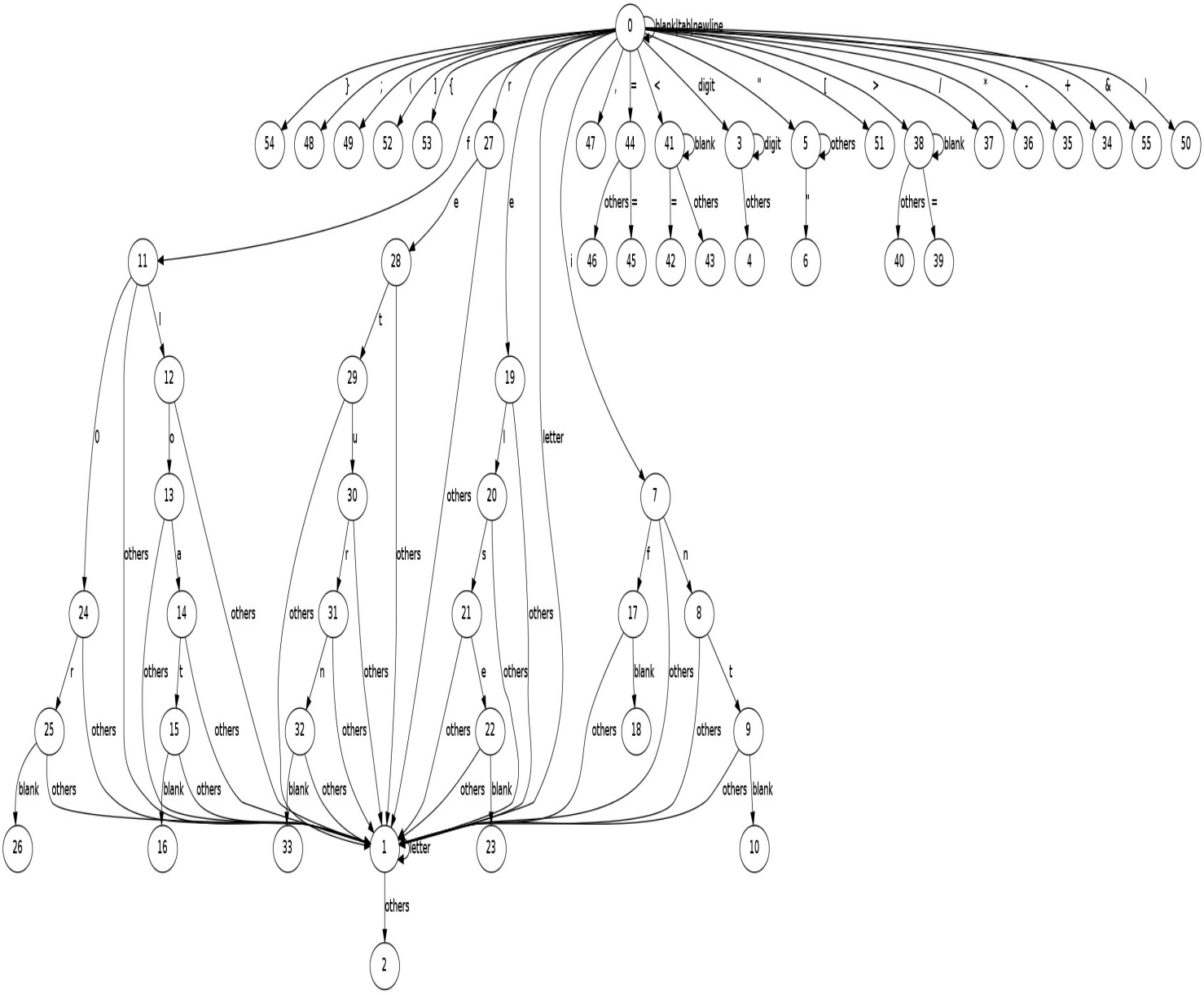
(DIGIT,整数)
(ID,标识符)
(STR,字符窜)
(INT,int)
(FLOAT,float)
(IF,if)
(FOR,for)
(ELSE,else)
(RETURN,return)
(+,+)
(-,-)
(*,*)
(/,/)
(>,>)
(>=,>=)
(<,<)
(<=,<=)
(==,==)
(INC,++)
(DOT,,)
(SEM,;)

(AND,&)
([,[)
(],])
({,{)
(},})
((,()
(),))

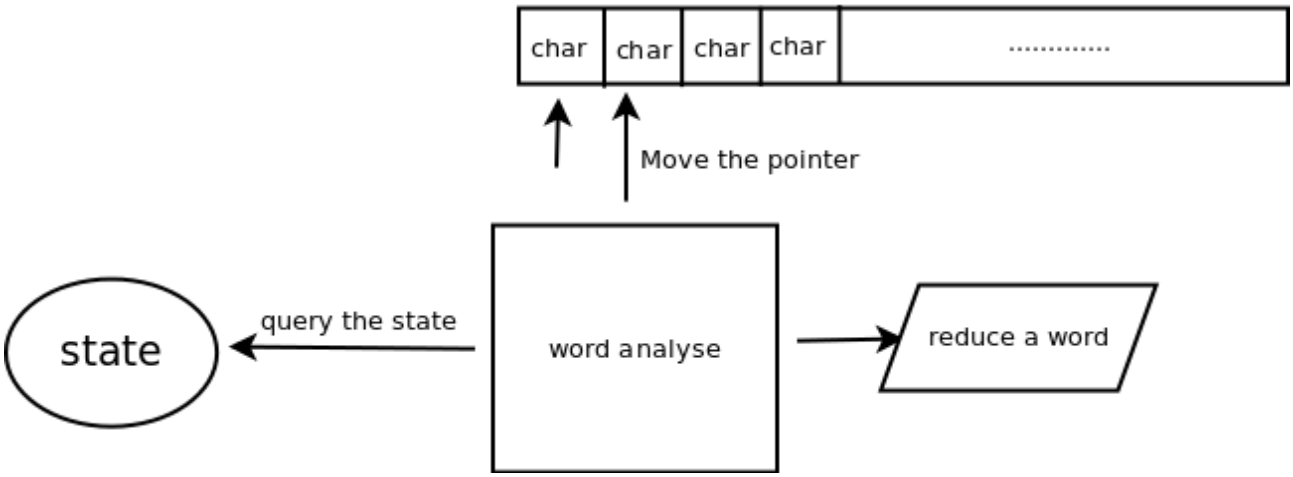
3.2.2 词法分析中的关键字

int，float，return，if，for，else。

3.3 词法分析状态转换图：



3.4 程序流程图：



4 语法分析器的设计与实现

4.1 语法分析器总体结构

读取词法分析完成之后产生的单词序列。构建两个栈，分别存入状态，和字典结构。按顺序读取单词序列，每当读取一个单词之后，根据状态栈的栈顶状态。查询 action 表，确定是归约产生式，或者是移进单词到语义栈中。这里的单词是一个字典，包括名字，单词本身，类型等信息。一旦查询 action 表得到需要归约到某个产生式，则从状态栈中弹出相应个数的状态。从语义栈中弹出相应个数的符号，并且压入归约产生式的左边部分，在这一部分，压入栈中的符号也是一个字典，需要调用相应的语义动作构造这个字典。然后根据当前状态栈的栈顶状态和语义栈中的栈定符号，查询 goto 表，得到表中的相应状态，将该状态压入状态栈中。然后继续读取下一个单词。直到最终得到 accept 标识符。如果中间查询 action 表和 goto 表得到空时，表示源文件语义表示不合法。调用出错处理程序，打印相应信息。结束程序。

4.2 数据结构设计

4.2.1 分析栈

在这里分析栈使用了两个栈，一个状态栈，一个语义栈。状态栈初始状态压入 0.语义栈为空。状态栈的符号由分析表得出。用于在语义分析时候，压入相应状态。语义栈里面存放的一个个字典结构。每个字典结构类型如下：

{'name': ,'value': ,'type': ,'size': }

根据表中查询的结果，对栈进行移进和归约操作。在归约的时候传递相应的语义信息，和执行相应的语义动作。

4.2.2 分析表

这里使用的分析表是 LALR 分析表。分为 action 表和 goto 表。action 作用：当读取一个单词时候，根据当前栈顶状态和该单词查询 action 表相应的项，得到移进或者归约操作。然后执行相应作用。
goto 表作用：当归约一个产生式之后，根据状态栈顶状态值和语义栈顶的结构体中的 name 字段的值。查询 goto 表相应的项。得到一个状态值，将其压入状态栈。

4.3 LALR 分析表构造

LALR 分析表构造，分析产生式的 first 集和 follow 集。然后画出项目状态集，最后构造出 LALR 分析表。因为构造表的过程比较机械，而且也不是编译的主要内容。在这里，我使用编译工作台这个工具来生成 LALR 分析表。通过输入文法产生式，终结符号，非终结符号。可以自动生成 LALR 分析表。

4.3.1 Action 表：

0 RETURN	error
0 >=	error
0 <=	error
0 DOT	error
0 $	error
0)	error
0 (	error
0 +	error
0 *	error
0 -	error
0 /	error
0 SEM	error
0 =	error
0 <	error
0 >	error
0 FOR	error
0 ELSE	error
0 STR	error
0 [	error
0]	error
0 ID	error

0 IF	error
0 AND	error
0 DIGIT	error
0 INT	shift 2
0 FLOAT	error
0 {	error
0 }	error
0 INC	error
.....	.....

完整 action 表见附录 2

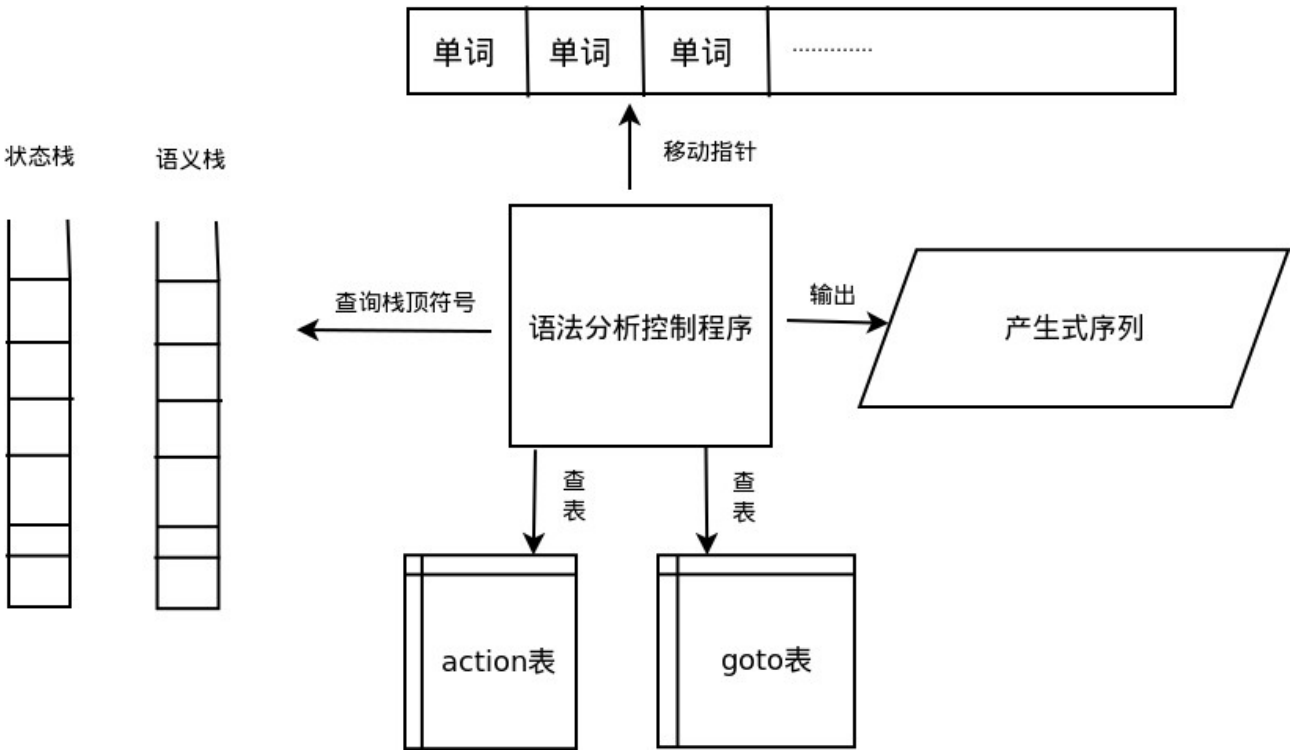
4.3.2 goto 表

0 inc_e	error
0 cast_e	error
0 add_e	error
0 declarator_e	error
0 select_e	error
0 type	error
0 expression_list	error
0 assignment_e	error
0 bool_e	error
0 iterator_e	error
0 p	1
0 digit_list	error
0 multi_e	error
0 declarator	error
0 func_e	error
0 expression	error

0 declarator_list	error
.....	.....

完整 goto 表见附录 3

4.4 程序流程图



5 语义子程序与汇编代码生成器的设计与实现

5.1 目标语言*(*汇编语言*)*的简单定义

目标程序采用 linux 下的 AT&T 汇编语言。

5.1.1 定义段空间

.section .data　　有初值数据段

.section .bss　　无初值数据段

.section .text　　代码段

5.1.2 程序开始

```
.section .data
a:      .int    4               定义变量 a 值为 4
.section .bss
        .lcomm     b,       2       定义变量 b 大小为 2 字节
.section .text
.globl              _start              程序的开始
_start:
```

5.1.3 程序结束

```
movl        $1,   %eax        程序的结束
movl        $0,   %ebx
int          $0x80
```

5.1.4 寄存器

32 位通用寄存器：%eax, %ebx, %ecx, %edx

16 位通用寄存器：%ax, %bx, %cx, %dx

8 位通用寄存器：　　%ah, %al, %bh, %bl, %ch, %cl, %dh, %dl

段寄存器：　cs, ds, ss, es, fs, gs（很少使用，不用修改）

段寄存器配对：

cs:eip(代码段指针) ds:esi(数据段), es:edi, ss:esp(栈顶指针), ss:ebp(栈底指针)

5.1.5 数据类型

```
.ascii              字符串
.short              短整型 16 位 2 字节
.int .long          长整型 32 位 4 字节
.byte               一个字节
.float              浮点单精度
.double             浮点双精度
.comm               名称,长度       长度为字节大小，定义一个为初始化的变量放在 bss 中
```

5.1.6 寻址方式

base_address(offset_address, index, size)

得到地址为 base_address + offset_address + index * size

例如 -4(%eax, %ebx, 2)　　得到地址为 -4 + %eax + %ebx * 2

其中 base_address 可以为变量或者常量，offset_address 和 index 必须为寄存器

5.1.7 常用指令

movb	b 表示一个字节
movw	w 表示 2 字节 short 型整数
movl	l 表示 4 字节 long 型整数

指令	指令格式	指令说明
movb, movw, movl	movl 源，目的	源地址数据移动到目的地址
addb, addw, addl	addl 源，目的	目的地址 = 目的地址 + 源地址
subb, subw, subl	subl 源，目的	目的地址 = 目的地址 - 源地址
mulb, mulw, mull	mull 源	8 位，AX = AL * 源 16 位，DX:AX = AX * 源 32 位，EDX:EAX = EAX * 源
divb, divw, divl	divl 源	8 位，AL = AX / 源　AH = AX % 源 16 位，AX = DX:AX / 源　DX = DX:AX % 源 32 位，EAX = EDX:EAX / 源 EDX = EDX:EAX % 源
cmp	cmp 源，目的	源和目的比较，后面接 jz、jnz、ja、jb、jae、jbe
jz	jz 目的代码	如果源和目的相等跳转
jnz	jnz 目的代码	如果源和目的不相等跳转
ja	ja 目的代码	目的 > 源
jb	jb 目的代码	目的 < 源
jae	jae 目的代码	目的 >= 源
jbe	jbe 目的代码	目的 <= 源
pushw, pushl	pushl 源	将源地址数据压栈
popw, popl	popl 目的	将栈顶元素弹出到目的地址

5.2 语义子程序的设计

5.2.1 属性设计

在语义栈里的每个元素都是一个字典结构，里面保存着该元素的所有信息。其中有一项字段名字为 type。保存该元素的属性。在进行归约的时候，将产生式右边的属性值传递到产生式左边的非终结符。然后压入语义栈中。

属性值包括：int，float，digit，digit_list，id，id_list，array。

不同的属性值决定了翻译时候使用不用的翻译方式：

int：在类型声明时候将该属性值传递到产生式的左边并且压入栈中

float:在类型声明时候将该属性值传递到产生式的左边并且压入栈中

digit:在语法制导翻译时，当作立即数进行翻译。

digit_list：在语法指导翻译时候，遍历整个 value 字段的列表长度，对于每个值当作立即数进行翻译。

id：在声明的时候传递信息，在语法制导翻译时候，调用存在符号表 f_table 中的相应键值指向的地址。通讯寻址方法翻译。

id_list：在声明语句中，遍历列表中的 id 值。分配相应的栈地址。

array：该属性的变量中。存放数组变量或者中间变量。翻译时候需要分别调用符号表中 f_table[‘addr’]中保存栈地址计数值和 f_table[‘tmp’]中保存的中间变量计数值。

5.2.2 存储分配子程序设计

在存储分配中主要使用两种方式：栈空间地址和中间变量。

栈空间地址需要在符号表中保存了一个栈地址计数值 f_table['addr']，该符号表项中保存了当前栈地址使用值。当需要申请新的栈存储空间时候，首先移动栈顶指针：

subl $num,%esp

其中 num 值为需要存储的变量的大小，单位为字节，然后在符号表中记录该变量对应的栈地址的值。这里的栈地址的值需要访问符号表的栈地址计数值 f_table['addr']。然后将 f_table['addr']的值加上相应的变量空间大小，便于下次给变量分配栈空间地址时使用。

中间变量用于保存算数表达式计算的中间结果。中间变量的分配管理使用符号表中的 f_table['tmp'] 项保存的值。所有中间变量都会以 tmp 开头，后面跟着一个数字。用数字的不同来区别不同的中间变量。当算式表达式计算的中间结果需要保存的时候就需要一个中间变量来存储。此时调用 符号表 f_table['tmp']中保存的值，分配一个新的中间变量'tmp' + 'f_table['tmp']'。同时将 f_table['tmp']的值进行加 1。便于下次申请中间变量时候使用。

函数调用部分主要涉及到 printf 和 scanf 两个函数的调用，调用方法如下：

pushl $LC0

push 占位符对应的变量地址

.....

call printf（scanf）

addl $num，%esp

调用时候，先将字符串型的语句根据预先定义的地址压入栈中，然后压入相应的占位符的地址。调用汇编中的相应指令。最后根据之前压入的值占用栈的空间的大小，移动栈顶指针。使得回复到原先状态。

5.3 汇编代码生成器的设计与实现

汇编代码生成器的实现在语法分析的时候完成。根据不同的产生式表达式，翻译成相应的汇编代码。

在语法分析阶段，每当归约出一个产生式的时候，对该产生式进行判断，如果是是对应的产生式，则调用相应的翻译子程序。这里需要借助符号表 f_table 来管理。使用 right_exp[]数组保存产生式右边的各项终结符和非终结符。

当产生式为一个声明语句:分配栈空间。移动栈指针，并且绑定变量名到栈地址值。

当产生式为一个 printf 或者 scanf 语句的时候，进行压栈操作，并且调用相应的汇编子程序。最后移动栈指针，恢复到调用前栈的状态。

当产生式为一个复制语句时候，先进行语义信息和属性值传递。把产生式右边的各项属性值传到产生式

左边。然后由符号表 f_table 找到产生式左边变量的栈地址或者是中间变量地址。执行 movl 操作，将数值移到变量对应的地址空间中。当产生式右边为中间算术表达式时，根据符号表 f_table['tmp']保存的值分配相应中间变量。保存中间计算结果的数值。压入语义栈中便于下一次调用。

当产生式为 return 0 时候，调用相应的汇编结束程序。完成对整个程序的翻译。

6 测试

6.1 测试样例

测试样例程序:

```
#include<stdio.h>
int main(){
   int score[6] = {76, 82, 90, 86, 79, 62};
   int credit[6] = {2, 2, 1, 2, 2, 3};
   int stu_number;
   float mean, sum;
   int temp;
   int i;

   printf("please input your student number:");
   scanf("%d",&stu_number);

   sum = 0;
   temp = 0;
   for( i = 0; i < 6; i++)
   {
      sum = sum + score[i] * credit[i];
      temp = temp + credit[i];
   }
   mean = sum / temp;

   if(mean >= 60){
      mean = mean - 60;
      printf("the score of student number %d is %f higher than 60.\n",
stu_number, mean);
   }
```

```
    else{
        mean = 60 - mean;
        printf("the score of student number %d is %f lower than
60.\n",stu_number, mean);
    }
    return 0;
}
```

6.2 测试结果与分析说明

使用该小型编译器编译后得到汇编代码如下：

```
.section .data
tmp0: .int 0
tmp1: .int 0
tmp2: .int 0
tmp3: .int 0
tmp4: .int 0
tmp5: .int 0
tmp6: .int 0
tmp7: .int 0
tmp8: .int 0
.section .text
.globl main
main:
pushl %ebp
movl %esp, %ebp
subl $24,%esp
movl $0,%ebx
movl $76, -24(%ebp,%ebx,4)
movl $1,%ebx
movl $82, -24(%ebp,%ebx,4)
movl $2,%ebx
movl $90, -24(%ebp,%ebx,4)
movl $3,%ebx
movl $86, -24(%ebp,%ebx,4)
movl $4,%ebx
movl $79, -24(%ebp,%ebx,4)
```

```
movl $5,%ebx
movl $62, -24(%ebp,%ebx,4)
subl $24,%esp
movl $0,%ebx
movl $2, -48(%ebp,%ebx,4)
movl $1,%ebx
movl $2, -48(%ebp,%ebx,4)
movl $2,%ebx
movl $1, -48(%ebp,%ebx,4)
movl $3,%ebx
movl $2, -48(%ebp,%ebx,4)
movl $4,%ebx
movl $2, -48(%ebp,%ebx,4)
movl $5,%ebx
movl $3, -48(%ebp,%ebx,4)
subl $4,%esp
subl $4,%esp
subl $4,%esp
subl $4,%esp
subl $4,%esp
pushl $LC0
call printf
addl $4,%esp
leal -52(%ebp),%eax
pushl %eax
pushl $LC1
call scanf
addl $8,%esp
movl $0,-60(%ebp)
movl $0,-64(%ebp)
movl $0,-68(%ebp)
start_for:
movl -68(%ebp), %eax
movl $6,%ebx
cmp %ebx,%eax
jae end_for
```

```
movl -68(%ebp),%edx
movl -24(%ebp,%edx,4), %eax
movl %eax, tmp0
movl -68(%ebp),%edx
movl -48(%ebp,%edx,4), %eax
movl %eax, tmp1
movl tmp0,%eax
movl tmp1,%ebx
mull %ebx
movl %eax,tmp2
movl -60(%ebp),%eax
movl tmp2,%ebx
addl %ebx,%eax
movl %eax,tmp3
movl tmp3,%eax
movl %eax,-60(%ebp)
movl -68(%ebp),%edx
movl -48(%ebp,%edx,4), %eax
movl %eax, tmp4
movl -64(%ebp),%eax
movl tmp4,%ebx
addl %ebx,%eax
movl %eax,tmp5
movl tmp5,%eax
movl %eax,-64(%ebp)
addl $1,-68(%ebp)
jmp start_for
end_for:
movl $0, %edx
movl -60(%ebp),%eax
movl -64(%ebp),%ebx
divl %ebx
movl %eax,tmp6
movl tmp6,%eax
movl %eax,-56(%ebp)
movl -56(%ebp),%eax
```

```
movl $60, %ebx
cmp %ebx, %eax
jb else
movl -56(%ebp),%eax
movl $60,%ebx
subl %ebx,%eax
movl %eax,tmp7
movl tmp7,%eax
movl %eax,-56(%ebp)
pushl -56(%ebp)
pushl -52(%ebp)
pushl $LC2
call printf
addl $12,%esp
jmp end
else:
movl $60,%eax
movl -56(%ebp),%ebx
subl %ebx,%eax
movl %eax,tmp8
movl tmp8,%eax
movl %eax,-56(%ebp)
pushl -56(%ebp)
pushl -52(%ebp)
pushl $LC3
call printf
addl $12,%esp
jmp end
end:
movl $1, %eax
movl $0, %ebx
int $0x80
LC0: .asciz "please input your student number:"
LC1: .asciz "%d"
LC2: .asciz "the score of student number %d is %d higher than 60.\n"
LC3: .asciz "the score of student number %d is %d lower than 60.\n"
```

汇编代码分析：

汇编代码中开头几行中的 tmp*变量是在语法制导翻译时动态增加的中间变量。在翻译的时候由符号表进行管理，在这里可以看出很多中间变量可以通过调整语句顺序而删掉的。但是由于在翻译的时候为了保证上下文语义的完整性，将每一次中间结果都进行了保存。代码中对栈的使用是动态开辟的，当需要使用多大空间的时候，动态的在栈中开辟相应的大小的空间。对于源程序的浮点类型，在这里为了翻译汇编时候的方便，统一使用了整型进行替换。避免了处理浮点寄存器的麻烦。对于读取标准输入和打印标准输出，这里直接调用了汇编里的相应指令。

程序执行结果分析：

将上述产生的汇编代码使用 gcc 汇编编译器编译成可执行文件。然后运行得到的可执行程序，运行结果如下：

```
[~/bianyi/test]-> ./a.out
please input your student number:78
the score of student number 78 is 16 higher than 60.
[~/bianyi/test]-> ./a.out
please input your student number:43
the score of student number 43 is 16 higher than 60.
```

与用 linux 标准 gcc 编译的可执行文件运行结果相比，除了浮点数和整型数之间的差异，其它运行结果均一致。说明该编译器可以在一定程度实现编译一个拥有一定文法规则的语言。

7 总结

本次软件设计与实践 4 课程基本是一个实践主要的课程。从编译器设计到最终实现都是自己亲自动手完成。在这个实践的过程中。对上一学期所学的编译原理理论课程有了更加深入和形象的了解。对编译器的总体结构也有了更加清晰的认识。尽管这是一个小型的编译器，可能支持的语法和逻辑结构也是非常有限，麻雀虽小，五脏俱全。这里面包括了所有一般性编译器所需要的结构和过程。词法分析，语法分析，语义分析和语法制导翻译，符号表管理，地址分配，到最终的汇编代码生成。同时对编程语言也有了新的认识。一门编程语言所有概念也是如此。为我以后学习新的编程语言奠定了基础。当然这样一个小型编译器的功能还是十分不完善的。对存储分配的优化，汇编代码优化部分做的工作比较少。这样让我认识到完成一个优秀的软件需要花费很大的精力以及广阔的知识面。总的来说，这次编译器实践课程也是对大学所学课程的一次很好的综合和总结，让我在即将毕业的大学生活收获非浅。

附录

附录 *1*

程序使用方法

该小型编译器使用 python 语言编写。大的部分包括词法分析，语法分析，语法分析和目标代码生成几个模块。可以使用如下方便编译相应的语言源文件：

执行 python xcc.py sourcefilename -S outfilename 会在指定目录下生成指定文件名的汇编代码文件。

其中 soucefilename，outfilename 可以是一个文件名，也可以是一个带目录的文件名。

然后调用 linux 下的 gcc 将汇编代码专成可执行文件；gcc filename -o outfilename。

均在 linux2.6 内核，64 位机器下测试成功。

附录 *2*

Action 表

24 RETURN	**error**
24 >=	**error**
24 <=	**error**
24 DOT	**reduce declarator_list -> declarator**
24 $	**error**
24)	**error**
24 (	**error**
24 +	**error**
24 *	**error**
24 -	**error**
24 /	**error**
24 SEM	**reduce declarator_list -> declarator**
24 =	**shift 33**
24 <	**error**
24 >	**error**
24 FOR	**error**
24 ELSE	**error**
24 STR	**error**
24 [	**error**
24]	**error**
24 ID	**error**
24 IF	**error**
24 AND	**error**
24 DIGIT	**error**
24 INT	**error**

24 FLOAT	**error**
24 {	**error**
24 }	**error**
24 INC	**error**
25 RETURN	**error**
25 >=	**error**
25 <=	**error**
25 DOT	**shift 34**
25 $	**error**
25)	**error**
25 (	**error**
25 +	**error**
25 *	**error**
25 -	**error**
25 /	**error**
25 SEM	**shift 35**
25 =	**error**
25 <	**error**
25 >	**error**
25 FOR	**error**
25 ELSE	**error**
25 STR	**error**
25 [	**error**
25]	**error**
25 ID	**error**
25 IF	**error**
25 AND	**error**
25 DIGIT	**error**

25 INT	**error**
25 FLOAT	**error**
25 {	**error**
25 }	**error**
25 INC	**error**
26 RETURN	**error**
26 >=	**error**
26 <=	**error**
26 DOT	**reduce declarator -> ID**
26 $	**error**
26)	**reduce declarator -> ID**
26 (	**error**
26 +	**error**
26 *	**error**
26 -	**error**
26 /	**error**
26 SEM	**reduce declarator -> ID**
26 =	**reduce declarator -> ID**
26 <	**error**
26 >	**error**
26 FOR	**error**
26 ELSE	**error**
26 STR	**error**
26 [	**shift 29**
26]	**error**
26 ID	**error**
26 IF	**error**
26 AND	**error**

26 DIGIT	**error**
26 INT	**error**
26 FLOAT	**error**
26 {	**error**
26 }	**error**
26 INC	**error**
27 RETURN	**error**
27 >=	**error**
27 <=	**error**
27 DOT	**error**
27 $	**error**
27)	**error**
27 (	**error**
27 +	**error**
27 *	**error**
27 -	**error**
27 /	**error**
27 SEM	**error**
27 =	**error**
27 <	**error**
27 >	**error**
27 FOR	**error**
27 ELSE	**error**
27 STR	**error**
27 [	**error**
27]	**error**
27 ID	**shift 40**
27 IF	**error**

27 AND	**error**
27 DIGIT	**shift 41**
27 INT	**error**
27 FLOAT	**error**
27 {	**error**
27 }	**error**
27 INC	**error**
20 RETURN	**error**
20 >=	**error**
20 <=	**error**
20 DOT	**error**
20 $	**error**
20)	**error**
20 (	**shift 31**
20 +	**error**
20 *	**error**
20 -	**error**
20 /	**error**
20 SEM	**error**
20 =	**error**
20 <	**error**
20 >	**error**
20 FOR	**error**
20 ELSE	**error**
20 STR	**error**
20 [	**error**
20]	**error**
20 ID	**error**

20 IF	**error**
20 AND	**error**
20 DIGIT	**error**
20 INT	**error**
20 FLOAT	**error**
20 {	**error**
20 }	**error**
20 INC	**error**
21 RETURN	**error**
21 >=	**error**
21 <=	**error**
21 DOT	**error**
21 $	**error**
21)	**error**
21 (	**error**
21 +	**error**
21 *	**error**
21 -	**error**
21 /	**error**
21 SEM	**error**
21 =	**error**
21 <	**error**
21 >	**error**
21 FOR	**error**
21 ELSE	**error**
21 STR	**error**
21 [	**error**
21]	**error**

21 ID	**error**
21 IF	**error**
21 AND	**error**
21 DIGIT	**shift 32**
21 INT	**error**
21 FLOAT	**error**
21 {	**error**
21 }	**error**
21 INC	**error**
22 RETURN	**reduce expression_list -> expression_list expression**
22 >=	**error**
22 <=	**error**
22 DOT	**error**
22 $	**error**
22)	**error**
22 (	**error**
22 +	**error**
22 *	**error**
22 -	**error**
22 /	**error**
22 SEM	**error**
22 =	**error**
22 <	**error**
22 >	**error**
22 FOR	**reduce expression_list -> expression_list expression**
22 ELSE	**error**
22 STR	**error**

22 [	**error**
22]	**error**
22 ID	**reduce expression_list -> expression_list expression**
22 IF	**reduce expression_list -> expression_list expression**
22 AND	**error**
22 DIGIT	**error**
22 INT	**reduce expression_list -> expression_list expression**
22 FLOAT	**reduce expression_list -> expression_list expression**
22 {	**error**
22 }	**reduce expression_list -> expression_list expression**
22 INC	**error**
23 RETURN	**error**
23 >=	**error**
23 <=	**error**
23 DOT	**error**
23 $	**reduce p -> INT ID ( ) { expression_list }**
23)	**error**
23 (	**error**
23 +	**error**
23 *	**error**
23 -	**error**
23 /	**error**
23 SEM	**error**
23 =	**error**
23 <	**error**

23 >	**error**
23 FOR	**error**
23 ELSE	**error**
23 STR	**error**
23 [	**error**
23]	**error**
23 ID	**error**
23 IF	**error**
23 AND	**error**
23 DIGIT	**error**
23 INT	**error**
23 FLOAT	**error**
23 {	**error**
23 }	**error**
23 INC	**error**
28 RETURN	**error**
28 >=	**error**
28 <=	**error**
28 DOT	**error**
28 $	**error**
28)	**error**
28 (	**error**
28 +	**error**
28 *	**error**
28 -	**error**
28 /	**error**
28 SEM	**error**
28 =	**error**

28 <	**error**
28 >	**error**
28 FOR	**error**
28 ELSE	**error**
28 STR	**shift 42**
28 [	**error**
28]	**error**
28 ID	**error**
28 IF	**error**
28 AND	**error**
28 DIGIT	**error**
28 INT	**error**
28 FLOAT	**error**
28 {	**error**
28 }	**error**
28 INC	**error**
29 RETURN	**error**
29 >=	**error**
29 <=	**error**
29 DOT	**error**
29 $	**error**
29)	**error**
29 (	**error**
29 +	**error**
29 *	**error**
29 -	**error**
29 /	**error**
29 SEM	**error**

29 =	**error**
29 <	**error**
29 >	**error**
29 FOR	**error**
29 ELSE	**error**
29 STR	**error**
29 [	**error**
29]	**error**
29 ID	**error**
29 IF	**error**
29 AND	**error**
29 DIGIT	**shift 43**
29 INT	**error**
29 FLOAT	**error**
29 {	**error**
29 }	**error**
29 INC	**error**
0 RETURN	**error**
0 >=	**error**
0 <=	**error**
0 DOT	**error**
0 $	**error**
0)	**error**
0 (	**error**
0 +	**error**
0 *	**error**
0 -	**error**
0 /	**error**

0 SEM	**error**
0 =	**error**
0 <	**error**
0 >	**error**
0 FOR	**error**
0 ELSE	**error**
0 STR	**error**
0 [	**error**
0]	**error**
0 ID	**error**
0 IF	**error**
0 AND	**error**
0 DIGIT	**error**
0 INT	**shift 2**
0 FLOAT	**error**
0 {	**error**
0 }	**error**
0 INC	**error**
4 RETURN	**error**
4 >=	**error**
4 <=	**error**
4 DOT	**error**
4 $	**error**
4)	**shift 5**
4 (	**error**
4 +	**error**
4 *	**error**
4 -	**error**

4 /	**error**
4 SEM	**error**
4 =	**error**
4 <	**error**
4 >	**error**
4 FOR	**error**
4 ELSE	**error**
4 STR	**error**
4 [	**error**
4]	**error**
4 ID	**error**
4 IF	**error**
4 AND	**error**
4 DIGIT	**error**
4 INT	**error**
4 FLOAT	**error**
4 {	**error**
4 }	**error**
4 INC	**error**
8 RETURN	**shift 21**
8 >=	**error**
8 <=	**error**
8 DOT	**error**
8 $	**error**
8)	**error**
8 (	**error**
8 +	**error**
8 *	**error**

8 -	**error**
8 /	**error**
8 SEM	**error**
8 =	**error**
8 <	**error**
8 >	**error**
8 FOR	**shift 20**
8 ELSE	**error**
8 STR	**error**
8 [	**error**
8]	**error**
8 ID	**shift 16**
8 IF	**shift 19**
8 AND	**error**
8 DIGIT	**error**
8 INT	**shift 17**
8 FLOAT	**shift 18**
8 {	**error**
8 }	**shift 23**
8 INC	**error**
59 RETURN	**error**
59 >=	**error**
59 <=	**error**
59 DOT	**error**
59 $	**error**
59)	**error**
59 (	**error**
59 +	**error**

59 *	**error**
59 -	**error**
59 /	**error**
59 SEM	**error**
59 =	**error**
59 <	**error**
59 >	**error**
59 FOR	**error**
59 ELSE	**error**
59 STR	**error**
59 [	**error**
59]	**error**
59 ID	**shift 40**
59 IF	**error**
59 AND	**error**
59 DIGIT	**shift 41**
59 INT	**error**
59 FLOAT	**error**
59 {	**error**
59 }	**error**
59 INC	**error**
58 RETURN	**error**
58 >=	**reduce declarator -> ID [DIGIT]**
58 <=	**reduce declarator -> ID [DIGIT]**
58 DOT	**reduce declarator -> ID [DIGIT]**
58 $	**error**
58)	**reduce declarator -> ID [DIGIT]**
58 (	**error**

58 +	**reduce declarator -> ID [DIGIT]**
58 *	**reduce declarator -> ID [DIGIT]**
58 -	**reduce declarator -> ID [DIGIT]**
58 /	**reduce declarator -> ID [DIGIT]**
58 SEM	**reduce declarator -> ID [DIGIT]**
58 =	**reduce declarator -> ID [DIGIT]**
58 <	**reduce declarator -> ID [DIGIT]**
58 >	**reduce declarator -> ID [DIGIT]**
58 FOR	**error**
58 ELSE	**error**
58 STR	**error**
58 [	**error**
58]	**error**
58 ID	**error**
58 IF	**error**
58 AND	**error**
58 DIGIT	**error**
58 INT	**error**
58 FLOAT	**error**
58 {	**error**
58 }	**error**
58 INC	**error**
55 RETURN	**error**
55 >=	**error**
55 <=	**error**
55 DOT	**error**
55 $	**error**
55)	**error**

55 (	**error**
55 +	**error**
55 *	**error**
55 -	**error**
55 /	**error**
55 SEM	**error**
55 =	**error**
55 <	**error**
55 >	**error**
55 FOR	**error**
55 ELSE	**error**
55 STR	**error**
55 [	**error**
55]	**error**
55 ID	**shift 71**
55 IF	**error**
55 AND	**error**
55 DIGIT	**shift 43**
55 INT	**error**
55 FLOAT	**error**
55 {	**error**
55 }	**error**
55 INC	**error**
54 RETURN	**error**
54 >=	**error**
54 <=	**error**
54 DOT	**error**
54 $	**error**

54)	**error**
54 (	**error**
54 +	**error**
54 *	**error**
54 -	**error**
54 /	**error**
54 SEM	**error**
54 =	**error**
54 <	**error**
54 >	**error**
54 FOR	**error**
54 ELSE	**error**
54 STR	**error**
54 [	**error**
54]	**error**
54 ID	**shift 40**
54 IF	**error**
54 AND	**error**
54 DIGIT	**shift 41**
54 INT	**error**
54 FLOAT	**error**
54 {	**error**
54 }	**error**
54 INC	**error**
57 RETURN	**error**
57 >=	**error**
57 <=	**error**
57 DOT	**error**

57 $	**error**
57)	**error**
57 (	**error**
57 +	**error**
57 *	**error**
57 -	**error**
57 /	**error**
57 SEM	**shift 75**
57 =	**error**
57 <	**error**
57 >	**error**
57 FOR	**error**
57 ELSE	**error**
57 STR	**error**
57 [	**error**
57]	**error**
57 ID	**error**
57 IF	**error**
57 AND	**error**
57 DIGIT	**error**
57 INT	**error**
57 FLOAT	**error**
57 {	**error**
57 }	**error**
57 INC	**error**
56 RETURN	**error**
56 >=	**error**
56 <=	**error**

56 DOT	**error**
56 $	**error**
56)	**error**
56 (	**error**
56 +	**error**
56 *	**error**
56 -	**error**
56 /	**error**
56 SEM	**error**
56 =	**error**
56 <	**error**
56 >	**error**
56 FOR	**error**
56 ELSE	**error**
56 STR	**error**
56 [	**error**
56]	**error**
56 ID	**shift 26**
56 IF	**error**
56 AND	**shift 74**
56 DIGIT	**error**
56 INT	**error**
56 FLOAT	**error**
56 {	**error**
56 }	**error**
56 INC	**error**
51 RETURN	**error**
51 >=	**error**

51 <=	**error**
51 DOT	**error**
51 $	**error**
51)	**error**
51 (	**error**
51 +	**error**
51 *	**error**
51 -	**error**
51 /	**error**
51 SEM	**error**
51 =	**error**
51 <	**error**
51 >	**error**
51 FOR	**error**
51 ELSE	**error**
51 STR	**error**
51 [	**error**
51]	**error**
51 ID	**shift 40**
51 IF	**error**
51 AND	**error**
51 DIGIT	**shift 41**
51 INT	**error**
51 FLOAT	**error**
51 {	**error**
51 }	**error**
51 INC	**error**
50 RETURN	**error**

50 >=	**error**
50 <=	**error**
50 DOT	**error**
50 $	**error**
50)	**error**
50 (	**error**
50 +	**error**
50 *	**error**
50 -	**error**
50 /	**error**
50 SEM	**error**
50 =	**error**
50 <	**error**
50 >	**error**
50 FOR	**error**
50 ELSE	**error**
50 STR	**error**
50 [	**error**
50]	**error**
50 ID	**shift 40**
50 IF	**error**
50 AND	**error**
50 DIGIT	**shift 41**
50 INT	**error**
50 FLOAT	**error**
50 {	**error**
50 }	**error**
50 INC	**error**

53 RETURN	**error**
53 >=	**error**
53 <=	**error**
53 DOT	**error**
53 $	**error**
53)	**error**
53 (	**error**
53 +	**error**
53 *	**error**
53 -	**error**
53 /	**error**
53 SEM	**error**
53 =	**error**
53 <	**error**
53 >	**error**
53 FOR	**error**
53 ELSE	**error**
53 STR	**error**
53 [	**error**
53]	**error**
53 ID	**shift 40**
53 IF	**error**
53 AND	**error**
53 DIGIT	**shift 41**
53 INT	**error**
53 FLOAT	**error**
53 {	**error**
53 }	**error**

53 INC	**error**
52 RETURN	**reduce assignment_e -> declarator = add_e SEM**
52 >=	**error**
52 <=	**error**
52 DOT	**error**
52 $	**error**
52)	**error**
52 (	**error**
52 +	**error**
52 *	**error**
52 -	**error**
52 /	**error**
52 SEM	**error**
52 =	**error**
52 <	**error**
52 >	**error**
52 FOR	**reduce assignment_e -> declarator = add_e SEM**
52 ELSE	**error**
52 STR	**error**
52 [	**error**
52]	**error**
52 ID	**reduce assignment_e -> declarator = add_e SEM**
52 IF	**reduce assignment_e -> declarator = add_e SEM**
52 AND	**error**
52 DIGIT	**reduce assignment_e -> declarator = add_e SEM**

52 INT	**reduce assignment_e -> declarator = add_e SEM**
52 FLOAT	**reduce assignment_e -> declarator = add_e SEM**
52 {	**error**
52 }	**reduce assignment_e -> declarator = add_e SEM**
52 INC	**error**
88 RETURN	**error**
88 >=	**error**
88 <=	**error**
88 DOT	**error**
88 $	**error**
88)	**shift 95**
88 (	**error**
88 +	**error**
88 *	**error**
88 -	**error**
88 /	**error**
88 SEM	**error**
88 =	**error**
88 <	**error**
88 >	**error**
88 FOR	**error**
88 ELSE	**error**
88 STR	**error**
88 [	**error**
88]	**error**
88 ID	**error**

88 IF	**error**
88 AND	**error**
88 DIGIT	**error**
88 INT	**error**
88 FLOAT	**error**
88 {	**error**
88 }	**error**
88 INC	**error**
89 RETURN	**error**
89 >=	**error**
89 <=	**error**
89 DOT	**error**
89 $	**error**
89)	**error**
89 (	**error**
89 +	**error**
89 *	**error**
89 -	**error**
89 /	**error**
89 SEM	**error**
89 =	**error**
89 <	**error**
89 >	**error**
89 FOR	**error**
89 ELSE	**error**
89 STR	**error**
89 [	**error**
89]	**error**

89 ID	**error**
89 IF	**error**
89 AND	**error**
89 DIGIT	**error**
89 INT	**error**
89 FLOAT	**error**
89 {	**error**
89 }	**error**
89 INC	**shift 96**
82 RETURN	**error**
82 >=	**error**
82 <=	**error**
82 DOT	**error**
82 $	**error**
82)	**error**
82 (	**error**
82 +	**error**
82 *	**error**
82 -	**error**
82 /	**error**
82 SEM	**error**
82 =	**error**
82 <	**error**
82 >	**error**
82 FOR	**error**
82 ELSE	**error**
82 STR	**error**
82 [	**error**

82]	**error**
82 ID	**error**
82 IF	**error**
82 AND	**error**
82 DIGIT	**shift 90**
82 INT	**error**
82 FLOAT	**error**
82 {	**error**
82 }	**error**
82 INC	**error**
83 RETURN	**error**
83 >=	**error**
83 <=	**error**
83 DOT	**error**
83 $	**error**
83)	**error**
83 (	**error**
83 +	**error**
83 *	**error**
83 -	**error**
83 /	**error**
83 SEM	**shift 91**
83 =	**error**
83 <	**error**
83 >	**error**
83 FOR	**error**
83 ELSE	**error**
83 STR	**error**

83 [	**error**
83]	**error**
83 ID	**error**
83 IF	**error**
83 AND	**error**
83 DIGIT	**error**
83 INT	**error**
83 FLOAT	**error**
83 {	**error**
83 }	**error**
83 INC	**error**
80 RETURN	**shift 21**
80 >=	**error**
80 <=	**error**
80 DOT	**error**
80 $	**error**
80)	**error**
80 (	**error**
80 +	**error**
80 *	**error**
80 -	**error**
80 /	**error**
80 SEM	**error**
80 =	**error**
80 <	**error**
80 >	**error**
80 FOR	**shift 20**
80 ELSE	**error**

80 STR	**error**
80 [	**error**
80]	**error**
80 ID	**shift 16**
80 IF	**shift 19**
80 AND	**error**
80 DIGIT	**error**
80 INT	**shift 17**
80 FLOAT	**shift 18**
80 {	**error**
80 }	**error**
80 INC	**error**
81 RETURN	**error**
81 >=	**error**
81 <=	**error**
81 DOT	**error**
81 $	**error**
81)	**error**
81 (	**error**
81 +	**error**
81 *	**error**
81 -	**error**
81 /	**error**
81 SEM	**error**
81 =	**error**
81 <	**error**
81 >	**error**
81 FOR	**error**

81 ELSE	**error**
81 STR	**error**
81 [	**error**
81]	**error**
81 ID	**shift 89**
81 IF	**error**
81 AND	**error**
81 DIGIT	**error**
81 INT	**error**
81 FLOAT	**error**
81 {	**error**
81 }	**error**
81 INC	**error**
86 RETURN	**error**
86 >=	**error**
86 <=	**error**
86 DOT	**error**
86 $	**error**
86)	**shift 93**
86 (	**error**
86 +	**error**
86 *	**error**
86 -	**error**
86 /	**error**
86 SEM	**error**
86 =	**error**
86 <	**error**
86 >	**error**

86 FOR	**error**
86 ELSE	**error**
86 STR	**error**
86 [	**error**
86]	**error**
86 ID	**error**
86 IF	**error**
86 AND	**error**
86 DIGIT	**error**
86 INT	**error**
86 FLOAT	**error**
86 {	**error**
86 }	**error**
86 INC	**error**
87 RETURN	**shift 21**
87 >=	**error**
87 <=	**error**
87 DOT	**error**
87 $	**error**
87)	**error**
87 (	**error**
87 +	**error**
87 *	**error**
87 -	**error**
87 /	**error**
87 SEM	**error**
87 =	**error**
87 <	**error**

87 >	**error**
87 FOR	**shift 20**
87 ELSE	**error**
87 STR	**error**
87 [	**error**
87]	**error**
87 ID	**shift 16**
87 IF	**shift 19**
87 AND	**error**
87 DIGIT	**error**
87 INT	**shift 17**
87 FLOAT	**shift 18**
87 {	**error**
87 }	**shift 94**
87 INC	**error**
84 RETURN	**error**
84 >=	**reduce cast_e -> ID [ID]**
84 <=	**reduce cast_e -> ID [ID]**
84 DOT	**error**
84 $	**error**
84)	**reduce cast_e -> ID [ID]**
84 (	**error**
84 +	**reduce cast_e -> ID [ID]**
84 *	**reduce cast_e -> ID [ID]**
84 -	**reduce cast_e -> ID [ID]**
84 /	**reduce cast_e -> ID [ID]**
84 SEM	**reduce cast_e -> ID [ID]**
84 =	**error**

84 <	**reduce cast_e -> ID [ID]**
84 >	**reduce cast_e -> ID [ID]**
84 FOR	**error**
84 ELSE	**error**
84 STR	**error**
84 [	**error**
84]	**error**
84 ID	**error**
84 IF	**error**
84 AND	**error**
84 DIGIT	**error**
84 INT	**error**
84 FLOAT	**error**
84 {	**error**
84 }	**error**
84 INC	**error**
85 RETURN	**error**
85 >=	**error**
85 <=	**error**
85 DOT	**error**
85 $	**error**
85)	**error**
85 (	**error**
85 +	**error**
85 *	**error**
85 -	**error**
85 /	**error**
85 SEM	**shift 92**

85 =	**error**
85 <	**error**
85 >	**error**
85 FOR	**error**
85 ELSE	**error**
85 STR	**error**
85 [	**error**
85]	**error**
85 ID	**error**
85 IF	**error**
85 AND	**error**
85 DIGIT	**error**
85 INT	**error**
85 FLOAT	**error**
85 {	**error**
85 }	**error**
85 INC	**error**
3 RETURN	**error**
3 >=	**error**
3 <=	**error**
3 DOT	**error**
3 $	**error**
3)	**error**
3 (	**shift 4**
3 +	**error**
3 *	**error**
3 -	**error**
3 /	**error**

3 SEM	**error**
3 =	**error**
3 <	**error**
3 >	**error**
3 FOR	**error**
3 ELSE	**error**
3 STR	**error**
3 [	**error**
3]	**error**
3 ID	**error**
3 IF	**error**
3 AND	**error**
3 DIGIT	**error**
3 INT	**error**
3 FLOAT	**error**
3 {	**error**
3 }	**error**
3 INC	**error**
7 RETURN	**reduce expression_list -> expression**
7 >=	**error**
7 <=	**error**
7 DOT	**error**
7 $	**error**
7)	**error**
7 (	**error**
7 +	**error**
7 *	**error**
7 -	**error**

7 /	**error**
7 SEM	**error**
7 =	**error**
7 <	**error**
7 >	**error**
7 FOR	**reduce expression_list -> expression**
7 ELSE	**error**
7 STR	**error**
7 [	**error**
7]	**error**
7 ID	**reduce expression_list -> expression**
7 IF	**reduce expression_list -> expression**
7 AND	**error**
7 DIGIT	**error**
7 INT	**reduce expression_list -> expression**
7 FLOAT	**reduce expression_list -> expression**
7 {	**error**
7 }	**reduce expression_list -> expression**
7 INC	**error**
102 RETURN	**shift 21**
102 >=	**error**
102 <=	**error**
102 DOT	**error**
102 $	**error**

102)	**error**
102 (	**error**
102 +	**error**
102 *	**error**
102 -	**error**
102 /	**error**
102 SEM	**error**
102 =	**error**
102 <	**error**
102 >	**error**
102 FOR	**shift 20**
102 ELSE	**error**
102 STR	**error**
102 [	**error**
102]	**error**
102 ID	**shift 16**
102 IF	**shift 19**
102 AND	**error**
102 DIGIT	**error**
102 INT	**shift 17**
102 FLOAT	**shift 18**
102 {	**error**
102 }	**shift 104**
102 INC	**error**
103 RETURN	**reduce iterator_e -> FOR (assignment_e bool_e SEM inc_e) { expression_list }**
103 >=	**error**
103 <=	**error**

103 DOT	**error**
103 $	**error**
103)	**error**
103 (	**error**
103 +	**error**
103 *	**error**
103 -	**error**
103 /	**error**
103 SEM	**error**
103 =	**error**
103 <	**error**
103 >	**error**
103 FOR	**reduce iterator_e -> FOR (assignment_e bool_e SEM inc_e) { expression_list }**
103 ELSE	**error**
103 STR	**error**
103 [	**error**
103]	**error**
103 ID	**reduce iterator_e -> FOR (assignment_e bool_e SEM inc_e) { expression_list }**
103 IF	**reduce iterator_e -> FOR (assignment_e bool_e SEM inc_e) { expression_list }**
103 AND	**error**
103 DIGIT	**error**
103 INT	**reduce iterator_e -> FOR (assignment_e bool_e SEM inc_e) { expression_list }**
103 FLOAT	**reduce iterator_e -> FOR (assignment_e bool_e SEM inc_e)**

	{ expression_list }
103 {	**error**
103 }	**reduce iterator_e -> FOR (assignment_e bool_e SEM inc_e) { expression_list }**
103 INC	**error**
100 RETURN	**shift 21**
100 >=	**error**
100 <=	**error**
100 DOT	**error**
100 $	**error**
100)	**error**
100 (	**error**
100 +	**error**
100 *	**error**
100 -	**error**
100 /	**error**
100 SEM	**error**
100 =	**error**
100 <	**error**
100 >	**error**
100 FOR	**shift 20**
100 ELSE	**error**
100 STR	**error**
100 [	**error**
100]	**error**
100 ID	**shift 16**
100 IF	**shift 19**
100 AND	**error**

100 DIGIT	**error**
100 INT	**shift 17**
100 FLOAT	**shift 18**
100 {	**error**
100 }	**error**
100 INC	**error**
101 RETURN	**shift 21**
101 >=	**error**
101 <=	**error**
101 DOT	**error**
101 $	**error**
101)	**error**
101 (	**error**
101 +	**error**
101 *	**error**
101 -	**error**
101 /	**error**
101 SEM	**error**
101 =	**error**
101 <	**error**
101 >	**error**
101 FOR	**shift 20**
101 ELSE	**error**
101 STR	**error**
101 [	**error**
101]	**error**
101 ID	**shift 16**
101 IF	**shift 19**

101 AND	**error**
101 DIGIT	**error**
101 INT	**shift 17**
101 FLOAT	**shift 18**
101 {	**error**
101 }	**shift 103**
101 INC	**error**
104 RETURN	**reduce select_e -> IF (bool_e) { expression_list } ELSE { expression_list }**
104 >=	**error**
104 <=	**error**
104 DOT	**error**
104 $	**error**
104)	**error**
104 (	**error**
104 +	**error**
104 *	**error**
104 -	**error**
104 /	**error**
104 SEM	**error**
104 =	**error**
104 <	**error**
104 >	**error**
104 FOR	**reduce select_e -> IF (bool_e) { expression_list } ELSE { expression_list }**
104 ELSE	**error**
104 STR	**error**
104 [	**error**

104]	**error**
104 ID	**reduce select_e -> IF (bool_e) { expression_list } ELSE { expression_list }**
104 IF	**reduce select_e -> IF (bool_e) { expression_list } ELSE { expression_list }**
104 AND	**error**
104 DIGIT	**error**
104 INT	**reduce select_e -> IF (bool_e) { expression_list } ELSE { expression_list }**
104 FLOAT	**reduce select_e -> IF (bool_e) { expression_list } ELSE { expression_list }**
104 {	**error**
104 }	**reduce select_e -> IF (bool_e) { expression_list } ELSE { expression_list }**
104 INC	**error**
39 RETURN	**error**
39 >=	**error**
39 <=	**error**
39 DOT	**error**
39 $	**error**
39)	**error**
39 (	**error**
39 +	**reduce multi_e -> cast_e**
39 *	**reduce multi_e -> cast_e**
39 -	**reduce multi_e -> cast_e**
39 /	**reduce multi_e -> cast_e**
39 SEM	**reduce multi_e -> cast_e**
39 =	**error**

39 <	**error**
39 >	**error**
39 FOR	**error**
39 ELSE	**error**
39 STR	**error**
39 [	**error**
39]	**error**
39 ID	**error**
39 IF	**error**
39 AND	**error**
39 DIGIT	**error**
39 INT	**error**
39 FLOAT	**error**
39 {	**error**
39 }	**error**
39 INC	**error**
38 RETURN	**error**
38 >=	**error**
38 <=	**error**
38 DOT	**error**
38 $	**error**
38)	**error**
38 (	**error**
38 +	**reduce add_e -> multi_e**
38 *	**shift 53**
38 -	**reduce add_e -> multi_e**
38 /	**shift 54**

38 SEM	**reduce add_e -> multi_e**
38 =	**error**
38 <	**error**
38 >	**error**
38 FOR	**error**
38 ELSE	**error**
38 STR	**error**
38 [	**error**
38]	**error**
38 ID	**error**
38 IF	**error**
38 AND	**error**
38 DIGIT	**error**
38 INT	**error**
38 FLOAT	**error**
38 {	**error**
38 }	**error**
38 INC	**error**
33 RETURN	**error**
33 >=	**error**
33 <=	**error**
33 DOT	**error**
33 $	**error**
33)	**error**
33 (	**error**
33 +	**error**
33 *	**error**
33 -	**error**

33 /	**error**
33 SEM	**error**
33 =	**error**
33 <	**error**
33 >	**error**
33 FOR	**error**
33 ELSE	**error**
33 STR	**error**
33 [	**error**
33]	**error**
33 ID	**error**
33 IF	**error**
33 AND	**error**
33 DIGIT	**error**
33 INT	**error**
33 FLOAT	**error**
33 {	**shift 48**
33 }	**error**
33 INC	**error**
32 RETURN	**error**
32 >=	**error**
32 <=	**error**
32 DOT	**error**
32 $	**error**
32)	**error**
32 (	**error**
32 +	**error**
32 *	**error**

32 -	**error**
32 /	**error**
32 SEM	**shift 47**
32 =	**error**
32 <	**error**
32 >	**error**
32 FOR	**error**
32 ELSE	**error**
32 STR	**error**
32 [	**error**
32]	**error**
32 ID	**error**
32 IF	**error**
32 AND	**error**
32 DIGIT	**error**
32 INT	**error**
32 FLOAT	**error**
32 {	**error**
32 }	**error**
32 INC	**error**
31 RETURN	**error**
31 >=	**error**
31 <=	**error**
31 DOT	**error**
31 $	**error**
31)	**error**
31 (	**error**
31 +	**error**

31 *	**error**
31 -	**error**
31 /	**error**
31 SEM	**error**
31 =	**error**
31 <	**error**
31 >	**error**
31 FOR	**error**
31 ELSE	**error**
31 STR	**error**
31 [	**error**
31]	**error**
31 ID	**shift 26**
31 IF	**error**
31 AND	**error**
31 DIGIT	**error**
31 INT	**error**
31 FLOAT	**error**
31 {	**error**
31 }	**error**
31 INC	**error**
30 RETURN	**error**
30 >=	**error**
30 <=	**error**
30 DOT	**error**
30 $	**error**
30)	**error**
30 (	**error**

30 +	**error**
30 *	**error**
30 -	**error**
30 /	**error**
30 SEM	**error**
30 =	**error**
30 <	**error**
30 >	**error**
30 FOR	**error**
30 ELSE	**error**
30 STR	**error**
30 [	**error**
30]	**error**
30 ID	**shift 40**
30 IF	**error**
30 AND	**error**
30 DIGIT	**shift 41**
30 INT	**error**
30 FLOAT	**error**
30 {	**error**
30 }	**error**
30 INC	**error**
37 RETURN	**error**
37 >=	**error**
37 <=	**error**
37 DOT	**error**
37 $	**error**
37)	**error**

37 (	**error**
37 +	**shift 50**
37 *	**error**
37 -	**shift 51**
37 /	**error**
37 SEM	**shift 52**
37 =	**error**
37 <	**error**
37 >	**error**
37 FOR	**error**
37 ELSE	**error**
37 STR	**error**
37 [	**error**
37]	**error**
37 ID	**error**
37 IF	**error**
37 AND	**error**
37 DIGIT	**error**
37 INT	**error**
37 FLOAT	**error**
37 {	**error**
37 }	**error**
37 INC	**error**
36 RETURN	**error**
36 >=	**reduce cast_e -> declarator**
36 <=	**reduce cast_e -> declarator**
36 DOT	**error**
36 $	**error**

36)	**reduce cast_e -> declarator**
36 (	**error**
36 +	**reduce cast_e -> declarator**
36 *	**reduce cast_e -> declarator**
36 -	**reduce cast_e -> declarator**
36 /	**reduce cast_e -> declarator**
36 SEM	**reduce cast_e -> declarator**
36 =	**error**
36 <	**reduce cast_e -> declarator**
36 >	**reduce cast_e -> declarator**
36 FOR	**error**
36 ELSE	**error**
36 STR	**error**
36 [	**error**
36]	**error**
36 ID	**error**
36 IF	**error**
36 AND	**error**
36 DIGIT	**error**
36 INT	**error**
36 FLOAT	**error**
36 {	**error**
36 }	**error**
36 INC	**error**
35 RETURN	**reduce declarator_e -> type declarator_list SEM**
35 >=	**error**
35 <=	**error**
35 DOT	**error**

35 $	**error**
35)	**error**
35 (	**error**
35 +	**error**
35 *	**error**
35 -	**error**
35 /	**error**
35 SEM	**error**
35 =	**error**
35 <	**error**
35 >	**error**
35 FOR	**reduce declarator_e -> type declarator_list SEM**
35 ELSE	**error**
35 STR	**error**
35 [	**error**
35]	**error**
35 ID	**reduce declarator_e -> type declarator_list SEM**
35 IF	**reduce declarator_e -> type declarator_list SEM**
35 AND	**error**
35 DIGIT	**error**
35 INT	**reduce declarator_e -> type declarator_list SEM**
35 FLOAT	**reduce declarator_e -> type declarator_list SEM**
35 {	**error**
35 }	**reduce declarator_e -> type declarator_list SEM**

35 INC	**error**
34 RETURN	**error**
34 >=	**error**
34 <=	**error**
34 DOT	**error**
34 $	**error**
34)	**error**
34 (	**error**
34 +	**error**
34 *	**error**
34 -	**error**
34 /	**error**
34 SEM	**error**
34 =	**error**
34 <	**error**
34 >	**error**
34 FOR	**error**
34 ELSE	**error**
34 STR	**error**
34 [	**error**
34]	**error**
34 ID	**shift 26**
34 IF	**error**
34 AND	**error**
34 DIGIT	**error**
34 INT	**error**
34 FLOAT	**error**
34 {	**error**

34 }	**error**
34 INC	**error**
60 RETURN	**error**
60 >=	**error**
60 <=	**error**
60 DOT	**error**
60 $	**error**
60)	**error**
60 (	**error**
60 +	**error**
60 *	**error**
60 -	**error**
60 /	**error**
60 SEM	**error**
60 =	**error**
60 <	**error**
60 >	**error**
60 FOR	**error**
60 ELSE	**error**
60 STR	**error**
60 [	**error**
60]	**error**
60 ID	**shift 40**
60 IF	**error**
60 AND	**error**
60 DIGIT	**shift 41**
60 INT	**error**
60 FLOAT	**error**

60 {	**error**
60 }	**error**
60 INC	**error**
61 RETURN	**error**
61 >=	**error**
61 <=	**error**
61 DOT	**error**
61 $	**error**
61)	**error**
61 (	**error**
61 +	**error**
61 *	**error**
61 -	**error**
61 /	**error**
61 SEM	**error**
61 =	**error**
61 <	**error**
61 >	**error**
61 FOR	**error**
61 ELSE	**error**
61 STR	**error**
61 [	**error**
61]	**error**
61 ID	**shift 40**
61 IF	**error**
61 AND	**error**
61 DIGIT	**shift 41**
61 INT	**error**

61 FLOAT	**error**
61 {	**error**
61 }	**error**
61 INC	**error**
62 RETURN	**error**
62 >=	**error**
62 <=	**error**
62 DOT	**error**
62 $	**error**
62)	**error**
62 (	**error**
62 +	**error**
62 *	**error**
62 -	**error**
62 /	**error**
62 SEM	**error**
62 =	**error**
62 <	**error**
62 >	**error**
62 FOR	**error**
62 ELSE	**error**
62 STR	**error**
62 [	**error**
62]	**error**
62 ID	**shift 40**
62 IF	**error**
62 AND	**error**
62 DIGIT	**shift 41**

62 INT	**error**
62 FLOAT	**error**
62 {	**error**
62 }	**error**
62 INC	**error**
63 RETURN	**error**
63 >=	**error**
63 <=	**error**
63 DOT	**error**
63 $	**error**
63)	**error**
63 (	**error**
63 +	**error**
63 *	**error**
63 -	**error**
63 /	**error**
63 SEM	**error**
63 =	**error**
63 <	**error**
63 >	**error**
63 FOR	**error**
63 ELSE	**error**
63 STR	**error**
63 [	**error**
63]	**error**
63 ID	**error**
63 IF	**error**
63 AND	**error**

63 DIGIT	**error**
63 INT	**error**
63 FLOAT	**error**
63 {	**shift 80**
63 }	**error**
63 INC	**error**
64 RETURN	**error**
64 >=	**error**
64 <=	**error**
64 DOT	**error**
64 $	**error**
64)	**error**
64 (	**error**
64 +	**error**
64 *	**error**
64 -	**error**
64 /	**error**
64 SEM	**shift 81**
64 =	**error**
64 <	**error**
64 >	**error**
64 FOR	**error**
64 ELSE	**error**
64 STR	**error**
64 [	**error**
64]	**error**
64 ID	**error**
64 IF	**error**

64 AND	**error**
64 DIGIT	**error**
64 INT	**error**
64 FLOAT	**error**
64 {	**error**
64 }	**error**
64 INC	**error**
65 RETURN	**error**
65 >=	**error**
65 <=	**error**
65 DOT	**shift 82**
65 $	**error**
65)	**error**
65 (	**error**
65 +	**error**
65 *	**error**
65 -	**error**
65 /	**error**
65 SEM	**error**
65 =	**error**
65 <	**error**
65 >	**error**
65 FOR	**error**
65 ELSE	**error**
65 STR	**error**
65 [	**error**
65]	**error**
65 ID	**error**

65 IF	**error**
65 AND	**error**
65 DIGIT	**error**
65 INT	**error**
65 FLOAT	**error**
65 {	**error**
65 }	**shift 83**
65 INC	**error**
66 RETURN	**error**
66 >=	**error**
66 <=	**error**
66 DOT	**reduce digit_list -> DIGIT**
66 $	**error**
66)	**error**
66 (	**error**
66 +	**error**
66 *	**error**
66 -	**error**
66 /	**error**
66 SEM	**error**
66 =	**error**
66 <	**error**
66 >	**error**
66 FOR	**error**
66 ELSE	**error**
66 STR	**error**
66 [	**error**
66]	**error**

66 ID	**error**
66 IF	**error**
66 AND	**error**
66 DIGIT	**error**
66 INT	**error**
66 FLOAT	**error**
66 {	**error**
66 }	**reduce digit_list -> DIGIT**
66 INC	**error**
67 RETURN	**error**
67 >=	**error**
67 <=	**error**
67 DOT	**error**
67 $	**error**
67)	**error**
67 (	**error**
67 +	**reduce add_e -> add_e + multi_e**
67 *	**shift 53**
67 -	**reduce add_e -> add_e + multi_e**
67 /	**shift 54**
67 SEM	**reduce add_e -> add_e + multi_e**
67 =	**error**
67 <	**error**
67 >	**error**
67 FOR	**error**
67 ELSE	**error**
67 STR	**error**
67 [	**error**

67]	**error**
67 ID	**error**
67 IF	**error**
67 AND	**error**
67 DIGIT	**error**
67 INT	**error**
67 FLOAT	**error**
67 {	**error**
67 }	**error**
67 INC	**error**
68 RETURN	**error**
68 >=	**error**
68 <=	**error**
68 DOT	**error**
68 $	**error**
68)	**error**
68 (	**error**
68 +	**reduce add_e -> add_e - multi_e**
68 *	**shift 53**
68 -	**reduce add_e -> add_e - multi_e**
68 /	**shift 54**
68 SEM	**reduce add_e -> add_e - multi_e**
68 =	**error**
68 <	**error**
68 >	**error**
68 FOR	**error**
68 ELSE	**error**
68 STR	**error**

68 [	**error**
68]	**error**
68 ID	**error**
68 IF	**error**
68 AND	**error**
68 DIGIT	**error**
68 INT	**error**
68 FLOAT	**error**
68 {	**error**
68 }	**error**
68 INC	**error**
69 RETURN	**error**
69 >=	**error**
69 <=	**error**
69 DOT	**error**
69 $	**error**
69)	**error**
69 (	**error**
69 +	**reduce multi_e -> multi_e * cast_e**
69 *	**reduce multi_e -> multi_e * cast_e**
69 -	**reduce multi_e -> multi_e * cast_e**
69 /	**reduce multi_e -> multi_e * cast_e**
69 SEM	**reduce multi_e -> multi_e * cast_e**
69 =	**error**
69 <	**error**
69 >	**error**
69 FOR	**error**
69 ELSE	**error**

69 STR	**error**
69 [	**error**
69]	**error**
69 ID	**error**
69 IF	**error**
69 AND	**error**
69 DIGIT	**error**
69 INT	**error**
69 FLOAT	**error**
69 {	**error**
69 }	**error**
69 INC	**error**
2 RETURN	**error**
2 >=	**error**
2 <=	**error**
2 DOT	**error**
2 $	**error**
2)	**error**
2 (	**error**
2 +	**error**
2 *	**error**
2 -	**error**
2 /	**error**
2 SEM	**error**
2 =	**error**
2 <	**error**
2 >	**error**
2 FOR	**error**

2 ELSE	**error**
2 STR	**error**
2 [	**error**
2]	**error**
2 ID	**shift 3**
2 IF	**error**
2 AND	**error**
2 DIGIT	**error**
2 INT	**error**
2 FLOAT	**error**
2 {	**error**
2 }	**error**
2 INC	**error**
6 RETURN	**shift 21**
6 >=	**error**
6 <=	**error**
6 DOT	**error**
6 $	**error**
6)	**error**
6 (	**error**
6 +	**error**
6 *	**error**
6 -	**error**
6 /	**error**
6 SEM	**error**
6 =	**error**
6 <	**error**
6 >	**error**

6 FOR	**shift 20**
6 ELSE	**error**
6 STR	**error**
6 [	**error**
6]	**error**
6 ID	**shift 16**
6 IF	**shift 19**
6 AND	**error**
6 DIGIT	**error**
6 INT	**shift 17**
6 FLOAT	**shift 18**
6 {	**error**
6 }	**error**
6 INC	**error**
99 RETURN	**shift 21**
99 >=	**error**
99 <=	**error**
99 DOT	**error**
99 $	**error**
99)	**error**
99 (	**error**
99 +	**error**
99 *	**error**
99 -	**error**
99 /	**error**
99 SEM	**error**
99 =	**error**
99 <	**error**

99 >	**error**
99 FOR	**shift 20**
99 ELSE	**error**
99 STR	**error**
99 [	**error**
99]	**error**
99 ID	**shift 16**
99 IF	**shift 19**
99 AND	**error**
99 DIGIT	**error**
99 INT	**shift 17**
99 FLOAT	**shift 18**
99 {	**error**
99 }	**error**
99 INC	**error**
98 RETURN	**error**
98 >=	**error**
98 <=	**error**
98 DOT	**error**
98 $	**error**
98)	**error**
98 (	**error**
98 +	**error**
98 *	**error**
98 -	**error**
98 /	**error**
98 SEM	**error**
98 =	**error**

98 <	**error**
98 >	**error**
98 FOR	**error**
98 ELSE	**error**
98 STR	**error**
98 [	**error**
98]	**error**
98 ID	**error**
98 IF	**error**
98 AND	**error**
98 DIGIT	**error**
98 INT	**error**
98 FLOAT	**error**
98 {	**shift 100**
98 }	**error**
98 INC	**error**
91 RETURN	**reduce declarator_e -> type declarator = { digit_list } SEM**
91 >=	**error**
91 <=	**error**
91 DOT	**error**
91 $	**error**
91)	**error**
91 (	**error**
91 +	**error**
91 *	**error**
91 -	**error**
91 /	**error**
91 SEM	**error**

91 =	**error**
91 <	**error**
91 >	**error**
91 FOR	**reduce declarator_e -> type declarator = { digit_list } SEM**
91 ELSE	**error**
91 STR	**error**
91 [	**error**
91]	**error**
91 ID	**reduce declarator_e -> type declarator = { digit_list } SEM**
91 IF	**reduce declarator_e -> type declarator = { digit_list } SEM**
91 AND	**error**
91 DIGIT	**error**
91 INT	**reduce declarator_e -> type declarator = { digit_list } SEM**
91 FLOAT	**reduce declarator_e -> type declarator = { digit_list } SEM**
91 {	**error**
91 }	**reduce declarator_e -> type declarator = { digit_list } SEM**
91 INC	**error**
90 RETURN	**error**
90 >=	**error**
90 <=	**error**
90 DOT	**reduce digit_list -> digit_list DOT DIGIT**
90 $	**error**
90)	**error**

90 (	**error**
90 +	**error**
90 *	**error**
90 -	**error**
90 /	**error**
90 SEM	**error**
90 =	**error**
90 <	**error**
90 >	**error**
90 FOR	**error**
90 ELSE	**error**
90 STR	**error**
90 [	**error**
90]	**error**
90 ID	**error**
90 IF	**error**
90 AND	**error**
90 DIGIT	**error**
90 INT	**error**
90 FLOAT	**error**
90 {	**error**
90 }	**reduce digit_list -> digit_list DOT DIGIT**
90 INC	**error**
93 RETURN	**error**
93 >=	**error**
93 <=	**error**
93 DOT	**error**
93 $	**error**

93)	**error**
93 (	**error**
93 +	**error**
93 *	**error**
93 -	**error**
93 /	**error**
93 SEM	**shift 97**
93 =	**error**
93 <	**error**
93 >	**error**
93 FOR	**error**
93 ELSE	**error**
93 STR	**error**
93 [	**error**
93]	**error**
93 ID	**error**
93 IF	**error**
93 AND	**error**
93 DIGIT	**error**
93 INT	**error**
93 FLOAT	**error**
93 {	**error**
93 }	**error**
93 INC	**error**
92 RETURN	**reduce func_e -> ID (STR DOT declarator_list) SEM**
92 >=	**error**
92 <=	**error**

92 DOT	**error**
92 $	**error**
92)	**error**
92 (	**error**
92 +	**error**
92 *	**error**
92 -	**error**
92 /	**error**
92 SEM	**error**
92 =	**error**
92 <	**error**
92 >	**error**
92 FOR	**reduce func_e -> ID (STR DOT declarator_list) SEM**
92 ELSE	**error**
92 STR	**error**
92 [	**error**
92]	**error**
92 ID	**reduce func_e -> ID (STR DOT declarator_list) SEM**
92 IF	**reduce func_e -> ID (STR DOT declarator_list) SEM**
92 AND	**error**
92 DIGIT	**error**
92 INT	**reduce func_e -> ID (STR DOT declarator_list) SEM**
92 FLOAT	**reduce func_e -> ID (STR DOT declarator_list) SEM**
92 {	**error**

92 }	**reduce func_e -> ID (STR DOT declarator_list) SEM**
92 INC	**error**
95 RETURN	**error**
95 >=	**error**
95 <=	**error**
95 DOT	**error**
95 $	**error**
95)	**error**
95 (	**error**
95 +	**error**
95 *	**error**
95 -	**error**
95 /	**error**
95 SEM	**error**
95 =	**error**
95 <	**error**
95 >	**error**
95 FOR	**error**
95 ELSE	**error**
95 STR	**error**
95 [	**error**
95]	**error**
95 ID	**error**
95 IF	**error**
95 AND	**error**
95 DIGIT	**error**
95 INT	**error**
95 FLOAT	**error**

95 {	**shift 99**
95 }	**error**
95 INC	**error**
94 RETURN	**error**
94 >=	**error**
94 <=	**error**
94 DOT	**error**
94 $	**error**
94)	**error**
94 (	**error**
94 +	**error**
94 *	**error**
94 -	**error**
94 /	**error**
94 SEM	**error**
94 =	**error**
94 <	**error**
94 >	**error**
94 FOR	**error**
94 ELSE	**shift 98**
94 STR	**error**
94 [	**error**
94]	**error**
94 ID	**error**
94 IF	**error**
94 AND	**error**
94 DIGIT	**error**

94 INT	**error**
94 FLOAT	**error**
94 {	**error**
94 }	**error**
94 INC	**error**
97 RETURN	**reduce func_e -> ID (STR DOT AND ID) SEM**
97 >=	**error**
97 <=	**error**
97 DOT	**error**
97 $	**error**
97)	**error**
97 (	**error**
97 +	**error**
97 *	**error**
97 -	**error**
97 /	**error**
97 SEM	**error**
97 =	**error**
97 <	**error**
97 >	**error**
97 FOR	**reduce func_e -> ID (STR DOT AND ID) SEM**
97 ELSE	**error**
97 STR	**error**
97 [	**error**
97]	**error**
97 ID	**reduce func_e -> ID (STR DOT AND ID) SEM**

97 IF	**reduce func_e -> ID (STR DOT AND ID) SEM**
97 AND	**error**
97 DIGIT	**error**
97 INT	**reduce func_e -> ID (STR DOT AND ID) SEM**
97 FLOAT	**reduce func_e -> ID (STR DOT AND ID) SEM**
97 {	**error**
97 }	**reduce func_e -> ID (STR DOT AND ID) SEM**
97 INC	**error**
96 RETURN	**error**
96 >=	**error**
96 <=	**error**
96 DOT	**error**
96 $	**error**
96)	**reduce inc_e -> ID INC**
96 (	**error**
96 +	**error**
96 *	**error**
96 -	**error**
96 /	**error**
96 SEM	**error**
96 =	**error**
96 <	**error**
96 >	**error**
96 FOR	**error**
96 ELSE	**error**
96 STR	**error**

96 [	**error**
96]	**error**
96 ID	**error**
96 IF	**error**
96 AND	**error**
96 DIGIT	**error**
96 INT	**error**
96 FLOAT	**error**
96 {	**error**
96 }	**error**
96 INC	**error**
11 RETURN	**reduce expression -> declarator_e**
11 >=	**error**
11 <=	**error**
11 DOT	**error**
11 $	**error**
11)	**error**
11 (	**error**
11 +	**error**
11 *	**error**
11 -	**error**
11 /	**error**
11 SEM	**error**
11 =	**error**
11 <	**error**
11 >	**error**
11 FOR	**reduce expression -> declarator_e**
11 ELSE	**error**

11 STR	**error**
11 [	**error**
11]	**error**
11 ID	**reduce expression -> declarator_e**
11 IF	**reduce expression -> declarator_e**
11 AND	**error**
11 DIGIT	**error**
11 INT	**reduce expression -> declarator_e**
11 FLOAT	**reduce expression -> declarator_e**
11 {	**error**
11 }	**reduce expression -> declarator_e**
11 INC	**error**
10 RETURN	**error**
10 >=	**error**
10 <=	**error**
10 DOT	**error**
10 $	**error**
10)	**error**
10 (	**error**
10 +	**error**
10 *	**error**
10 -	**error**
10 /	**error**
10 SEM	**error**
10 =	**shift 27**
10 <	**error**
10 >	**error**
10 FOR	**error**

10 ELSE	**error**
10 STR	**error**
10 [	**error**
10]	**error**
10 ID	**error**
10 IF	**error**
10 AND	**error**
10 DIGIT	**error**
10 INT	**error**
10 FLOAT	**error**
10 {	**error**
10 }	**error**
10 INC	**error**
13 RETURN	**reduce expression -> select_e**
13 >=	**error**
13 <=	**error**
13 DOT	**error**
13 $	**error**
13)	**error**
13 (	**error**
13 +	**error**
13 *	**error**
13 -	**error**
13 /	**error**
13 SEM	**error**
13 =	**error**
13 <	**error**
13 >	**error**

13 FOR	**reduce expression -> select_e**
13 ELSE	**error**
13 STR	**error**
13 [	**error**
13]	**error**
13 ID	**reduce expression -> select_e**
13 IF	**reduce expression -> select_e**
13 AND	**error**
13 DIGIT	**error**
13 INT	**reduce expression -> select_e**
13 FLOAT	**reduce expression -> select_e**
13 {	**error**
13 }	**reduce expression -> select_e**
13 INC	**error**
12 RETURN	**reduce expression -> assignment_e**
12 >=	**error**
12 <=	**error**
12 DOT	**error**
12 $	**error**
12)	**error**
12 (	**error**
12 +	**error**
12 *	**error**
12 -	**error**
12 /	**error**
12 SEM	**error**
12 =	**error**
12 <	**error**

12 >	**error**
12 FOR	**reduce expression -> assignment_e**
12 ELSE	**error**
12 STR	**error**
12 [	**error**
12]	**error**
12 ID	**reduce expression -> assignment_e**
12 IF	**reduce expression -> assignment_e**
12 AND	**error**
12 DIGIT	**error**
12 INT	**reduce expression -> assignment_e**
12 FLOAT	**reduce expression -> assignment_e**
12 {	**error**
12 }	**reduce expression -> assignment_e**
12 INC	**error**
15 RETURN	**reduce expression -> func_e**
15 >=	**error**
15 <=	**error**
15 DOT	**error**
15 $	**error**
15)	**error**
15 (	**error**
15 +	**error**
15 *	**error**

15 -	**error**
15 /	**error**
15 SEM	**error**
15 =	**error**
15 <	**error**
15 >	**error**
15 FOR	**reduce expression -> func_e**
15 ELSE	**error**
15 STR	**error**
15 [	**error**
15]	**error**
15 ID	**reduce expression -> func_e**
15 IF	**reduce expression -> func_e**
15 AND	**error**
15 DIGIT	**error**
15 INT	**reduce expression -> func_e**
15 FLOAT	**reduce expression -> func_e**
15 {	**error**
15 }	**reduce expression -> func_e**
15 INC	**error**
14 RETURN	**reduce expression -> iterator_e**
14 >=	**error**
14 <=	**error**
14 DOT	**error**
14 $	**error**
14)	**error**
14 (	**error**
14 +	**error**

14 *	**error**
14 -	**error**
14 /	**error**
14 SEM	**error**
14 =	**error**
14 <	**error**
14 >	**error**
14 FOR	**reduce expression -> iterator_e**
14 ELSE	**error**
14 STR	**error**
14 [	**error**
14]	**error**
14 ID	**reduce expression -> iterator_e**
14 IF	**reduce expression -> iterator_e**
14 AND	**error**
14 DIGIT	**error**
14 INT	**reduce expression -> iterator_e**
14 FLOAT	**reduce expression -> iterator_e**
14 {	**error**
14 }	**reduce expression -> iterator_e**
14 INC	**error**
17 RETURN	**error**
17 >=	**error**
17 <=	**error**
17 DOT	**error**
17 $	**error**
17)	**error**
17 (	**error**

17 +	**error**
17 *	**error**
17 -	**error**
17 /	**error**
17 SEM	**error**
17 =	**error**
17 <	**error**
17 >	**error**
17 FOR	**error**
17 ELSE	**error**
17 STR	**error**
17 [	**error**
17]	**error**
17 ID	**reduce type -> INT**
17 IF	**error**
17 AND	**error**
17 DIGIT	**error**
17 INT	**error**
17 FLOAT	**error**
17 {	**error**
17 }	**error**
17 INC	**error**
16 RETURN	**error**
16 >=	**error**
16 <=	**error**
16 DOT	**error**
16 $	**error**
16)	**error**

16 (	**shift 28**
16 +	**error**
16 *	**error**
16 -	**error**
16 /	**error**
16 SEM	**error**
16 =	**reduce declarator -> ID**
16 <	**error**
16 >	**error**
16 FOR	**error**
16 ELSE	**error**
16 STR	**error**
16 [	**shift 29**
16]	**error**
16 ID	**error**
16 IF	**error**
16 AND	**error**
16 DIGIT	**error**
16 INT	**error**
16 FLOAT	**error**
16 {	**error**
16 }	**error**
16 INC	**error**
19 RETURN	**error**
19 >=	**error**
19 <=	**error**
19 DOT	**error**
19 $	**error**

19)	**error**
19 (	**shift 30**
19 +	**error**
19 *	**error**
19 -	**error**
19 /	**error**
19 SEM	**error**
19 =	**error**
19 <	**error**
19 >	**error**
19 FOR	**error**
19 ELSE	**error**
19 STR	**error**
19 [	**error**
19]	**error**
19 ID	**error**
19 IF	**error**
19 AND	**error**
19 DIGIT	**error**
19 INT	**error**
19 FLOAT	**error**
19 {	**error**
19 }	**error**
19 INC	**error**
18 RETURN	**error**
18 >=	**error**
18 <=	**error**
18 DOT	**error**

18 $	**error**
18)	**error**
18 (	**error**
18 +	**error**
18 *	**error**
18 -	**error**
18 /	**error**
18 SEM	**error**
18 =	**error**
18 <	**error**
18 >	**error**
18 FOR	**error**
18 ELSE	**error**
18 STR	**error**
18 [	**error**
18]	**error**
18 ID	**reduce type -> FLOAT**
18 IF	**error**
18 AND	**error**
18 DIGIT	**error**
18 INT	**error**
18 FLOAT	**error**
18 {	**error**
18 }	**error**
18 INC	**error**
48 RETURN	**error**
48 >=	**error**
48 <=	**error**

48 DOT	**error**
48 $	**error**
48)	**error**
48 (	**error**
48 +	**error**
48 *	**error**
48 -	**error**
48 /	**error**
48 SEM	**error**
48 =	**error**
48 <	**error**
48 >	**error**
48 FOR	**error**
48 ELSE	**error**
48 STR	**error**
48 [	**error**
48]	**error**
48 ID	**error**
48 IF	**error**
48 AND	**error**
48 DIGIT	**shift 66**
48 INT	**error**
48 FLOAT	**error**
48 {	**error**
48 }	**error**
48 INC	**error**
49 RETURN	**error**
49 >=	**error**

49 <=	**error**
49 DOT	**reduce declarator_list -> declarator_list DOT declarator**
49 $	**error**
49)	**reduce declarator_list -> declarator_list DOT declarator**
49 (	**error**
49 +	**error**
49 *	**error**
49 -	**error**
49 /	**error**
49 SEM	**reduce declarator_list -> declarator_list DOT declarator**
49 =	**error**
49 <	**error**
49 >	**error**
49 FOR	**error**
49 ELSE	**error**
49 STR	**error**
49 [	**error**
49]	**error**
49 ID	**error**
49 IF	**error**
49 AND	**error**
49 DIGIT	**error**
49 INT	**error**
49 FLOAT	**error**
49 {	**error**
49 }	**error**

49 INC	**error**
46 RETURN	**error**
46 >=	**error**
46 <=	**error**
46 DOT	**error**
46 $	**error**
46)	**error**
46 (	**error**
46 +	**error**
46 *	**error**
46 -	**error**
46 /	**error**
46 SEM	**error**
46 =	**error**
46 <	**error**
46 >	**error**
46 FOR	**error**
46 ELSE	**error**
46 STR	**error**
46 [	**error**
46]	**error**
46 ID	**shift 40**
46 IF	**error**
46 AND	**error**
46 DIGIT	**shift 41**
46 INT	**error**
46 FLOAT	**error**
46 {	**error**

46 }	**error**
46 INC	**error**
47 RETURN	**reduce expression -> RETURN DIGIT SEM**
47 >=	**error**
47 <=	**error**
47 DOT	**error**
47 $	**error**
47)	**error**
47 (	**error**
47 +	**error**
47 *	**error**
47 -	**error**
47 /	**error**
47 SEM	**error**
47 =	**error**
47 <	**error**
47 >	**error**
47 FOR	**reduce expression -> RETURN DIGIT SEM**
47 ELSE	**error**
47 STR	**error**
47 [	**error**
47]	**error**
47 ID	**reduce expression -> RETURN DIGIT SEM**
47 IF	**reduce expression -> RETURN DIGIT SEM**
47 AND	**error**
47 DIGIT	**error**

47 INT	**reduce expression -> RETURN DIGIT SEM**
47 FLOAT	**reduce expression -> RETURN DIGIT SEM**
47 {	**error**
47 }	**reduce expression -> RETURN DIGIT SEM**
47 INC	**error**
44 RETURN	**error**
44 >=	**shift 61**
44 <=	**shift 62**
44 DOT	**error**
44 $	**error**
44)	**error**
44 (	**error**
44 +	**error**
44 *	**error**
44 -	**error**
44 /	**error**
44 SEM	**error**
44 =	**error**
44 <	**shift 60**
44 >	**shift 59**
44 FOR	**error**
44 ELSE	**error**
44 STR	**error**
44 [	**error**
44]	**error**
44 ID	**error**

44 IF	**error**
44 AND	**error**
44 DIGIT	**error**
44 INT	**error**
44 FLOAT	**error**
44 {	**error**
44 }	**error**
44 INC	**error**
45 RETURN	**error**
45 >=	**error**
45 <=	**error**
45 DOT	**error**
45 $	**error**
45)	**shift 63**
45 (	**error**
45 +	**error**
45 *	**error**
45 -	**error**
45 /	**error**
45 SEM	**error**
45 =	**error**
45 <	**error**
45 >	**error**
45 FOR	**error**
45 ELSE	**error**
45 STR	**error**
45 [	**error**
45]	**error**

45 ID	**error**
45 IF	**error**
45 AND	**error**
45 DIGIT	**error**
45 INT	**error**
45 FLOAT	**error**
45 {	**error**
45 }	**error**
45 INC	**error**
42 RETURN	**error**
42 >=	**error**
42 <=	**error**
42 DOT	**shift 56**
42 $	**error**
42)	**shift 57**
42 (	**error**
42 +	**error**
42 *	**error**
42 -	**error**
42 /	**error**
42 SEM	**error**
42 =	**error**
42 <	**error**
42 >	**error**
42 FOR	**error**
42 ELSE	**error**
42 STR	**error**
42 [	**error**

42]	**error**
42 ID	**error**
42 IF	**error**
42 AND	**error**
42 DIGIT	**error**
42 INT	**error**
42 FLOAT	**error**
42 {	**error**
42 }	**error**
42 INC	**error**
43 RETURN	**error**
43 >=	**error**
43 <=	**error**
43 DOT	**error**
43 $	**error**
43)	**error**
43 (	**error**
43 +	**error**
43 *	**error**
43 -	**error**
43 /	**error**
43 SEM	**error**
43 =	**error**
43 <	**error**
43 >	**error**
43 FOR	**error**
43 ELSE	**error**
43 STR	**error**

43 [	**error**
43]	**shift 58**
43 ID	**error**
43 IF	**error**
43 AND	**error**
43 DIGIT	**error**
43 INT	**error**
43 FLOAT	**error**
43 {	**error**
43 }	**error**
43 INC	**error**
40 RETURN	**error**
40 >=	**reduce declarator -> ID**
40 <=	**reduce declarator -> ID**
40 DOT	**error**
40 $	**error**
40)	**reduce declarator -> ID**
40 (	**error**
40 +	**reduce declarator -> ID**
40 *	**reduce declarator -> ID**
40 -	**reduce declarator -> ID**
40 /	**reduce declarator -> ID**
40 SEM	**reduce declarator -> ID**
40 =	**error**
40 <	**reduce declarator -> ID**
40 >	**reduce declarator -> ID**
40 FOR	**error**
40 ELSE	**error**

40 STR	**error**
40 [	**shift 55**
40]	**error**
40 ID	**error**
40 IF	**error**
40 AND	**error**
40 DIGIT	**error**
40 INT	**error**
40 FLOAT	**error**
40 {	**error**
40 }	**error**
40 INC	**error**
41 RETURN	**error**
41 >=	**reduce cast_e -> DIGIT**
41 <=	**reduce cast_e -> DIGIT**
41 DOT	**error**
41 $	**error**
41)	**reduce cast_e -> DIGIT**
41 (	**error**
41 +	**reduce cast_e -> DIGIT**
41 *	**reduce cast_e -> DIGIT**
41 -	**reduce cast_e -> DIGIT**
41 /	**reduce cast_e -> DIGIT**
41 SEM	**reduce cast_e -> DIGIT**
41 =	**error**
41 <	**reduce cast_e -> DIGIT**
41 >	**reduce cast_e -> DIGIT**
41 FOR	**error**

41 ELSE	**error**
41 STR	**error**
41 [	**error**
41]	**error**
41 ID	**error**
41 IF	**error**
41 AND	**error**
41 DIGIT	**error**
41 INT	**error**
41 FLOAT	**error**
41 {	**error**
41 }	**error**
41 INC	**error**
1 RETURN	**error**
1 >=	**error**
1 <=	**error**
1 DOT	**error**
1 $	**accept**
1)	**error**
1 (	**error**
1 +	**error**
1 *	**error**
1 -	**error**
1 /	**error**
1 SEM	**error**
1 =	**error**
1 <	**error**
1 >	**error**

1 FOR	**error**
1 ELSE	**error**
1 STR	**error**
1 [	**error**
1]	**error**
1 ID	**error**
1 IF	**error**
1 AND	**error**
1 DIGIT	**error**
1 INT	**error**
1 FLOAT	**error**
1 {	**error**
1 }	**error**
1 INC	**error**
5 RETURN	**error**
5 >=	**error**
5 <=	**error**
5 DOT	**error**
5 $	**error**
5)	**error**
5 (	**error**
5 +	**error**
5 *	**error**
5 -	**error**
5 /	**error**
5 SEM	**error**
5 =	**error**
5 <	**error**

5 >	**error**
5 FOR	**error**
5 ELSE	**error**
5 STR	**error**
5 [	**error**
5]	**error**
5 ID	**error**
5 IF	**error**
5 AND	**error**
5 DIGIT	**error**
5 INT	**error**
5 FLOAT	**error**
5 {	**shift 6**
5 }	**error**
5 INC	**error**
9 RETURN	**error**
9 >=	**error**
9 <=	**error**
9 DOT	**error**
9 $	**error**
9)	**error**
9 (	**error**
9 +	**error**
9 *	**error**
9 -	**error**
9 /	**error**
9 SEM	**error**
9 =	**error**

9 <	**error**
9 >	**error**
9 FOR	**error**
9 ELSE	**error**
9 STR	**error**
9 [	**error**
9]	**error**
9 ID	**shift 26**
9 IF	**error**
9 AND	**error**
9 DIGIT	**error**
9 INT	**error**
9 FLOAT	**error**
9 {	**error**
9 }	**error**
9 INC	**error**
77 RETURN	**error**
77 >=	**error**
77 <=	**error**
77 DOT	**error**
77 $	**error**
77)	**reduce bool_e -> cast_e < cast_e**
77 (	**error**
77 +	**error**
77 *	**error**
77 -	**error**
77 /	**error**
77 SEM	**reduce bool_e -> cast_e < cast_e**

77 =	**error**
77 <	**error**
77 >	**error**
77 FOR	**error**
77 ELSE	**error**
77 STR	**error**
77 [	**error**
77]	**error**
77 ID	**error**
77 IF	**error**
77 AND	**error**
77 DIGIT	**error**
77 INT	**error**
77 FLOAT	**error**
77 {	**error**
77 }	**error**
77 INC	**error**
76 RETURN	**error**
76 >=	**error**
76 <=	**error**
76 DOT	**error**
76 $	**error**
76)	**reduce bool_e -> cast_e > cast_e**
76 (	**error**
76 +	**error**
76 *	**error**
76 -	**error**
76 /	**error**

76 SEM	**reduce bool_e -> cast_e > cast_e**
76 =	**error**
76 <	**error**
76 >	**error**
76 FOR	**error**
76 ELSE	**error**
76 STR	**error**
76 [	**error**
76]	**error**
76 ID	**error**
76 IF	**error**
76 AND	**error**
76 DIGIT	**error**
76 INT	**error**
76 FLOAT	**error**
76 {	**error**
76 }	**error**
76 INC	**error**
75 RETURN	**reduce func_e -> ID (STR) SEM**
75 >=	**error**
75 <=	**error**
75 DOT	**error**
75 $	**error**
75)	**error**
75 (	**error**
75 +	**error**
75 *	**error**
75 -	**error**

75 /	**error**
75 SEM	**error**
75 =	**error**
75 <	**error**
75 >	**error**
75 FOR	**reduce func_e -> ID (STR) SEM**
75 ELSE	**error**
75 STR	**error**
75 [	**error**
75]	**error**
75 ID	**reduce func_e -> ID (STR) SEM**
75 IF	**reduce func_e -> ID (STR) SEM**
75 AND	**error**
75 DIGIT	**error**
75 INT	**reduce func_e -> ID (STR) SEM**
75 FLOAT	**reduce func_e -> ID (STR) SEM**
75 {	**error**
75 }	**reduce func_e -> ID (STR) SEM**
75 INC	**error**
74 RETURN	**error**
74 >=	**error**
74 <=	**error**
74 DOT	**error**
74 $	**error**
74)	**error**
74 (	**error**
74 +	**error**
74 *	**error**

74 -	**error**
74 /	**error**
74 SEM	**error**
74 =	**error**
74 <	**error**
74 >	**error**
74 FOR	**error**
74 ELSE	**error**
74 STR	**error**
74 [	**error**
74]	**error**
74 ID	**shift 86**
74 IF	**error**
74 AND	**error**
74 DIGIT	**error**
74 INT	**error**
74 FLOAT	**error**
74 {	**error**
74 }	**error**
74 INC	**error**
73 RETURN	**error**
73 >=	**error**
73 <=	**error**
73 DOT	**shift 34**
73 $	**error**
73)	**shift 85**
73 (	**error**
73 +	**error**

73 *	**error**
73 -	**error**
73 /	**error**
73 SEM	**error**
73 =	**error**
73 <	**error**
73 >	**error**
73 FOR	**error**
73 ELSE	**error**
73 STR	**error**
73 [	**error**
73]	**error**
73 ID	**error**
73 IF	**error**
73 AND	**error**
73 DIGIT	**error**
73 INT	**error**
73 FLOAT	**error**
73 {	**error**
73 }	**error**
73 INC	**error**
72 RETURN	**error**
72 >=	**error**
72 <=	**error**
72 DOT	**reduce declarator_list -> declarator**
72 $	**error**
72)	**reduce declarator_list -> declarator**

72 (	**error**
72 +	**error**
72 *	**error**
72 -	**error**
72 /	**error**
72 SEM	**error**
72 =	**error**
72 <	**error**
72 >	**error**
72 FOR	**error**
72 ELSE	**error**
72 STR	**error**
72 [	**error**
72]	**error**
72 ID	**error**
72 IF	**error**
72 AND	**error**
72 DIGIT	**error**
72 INT	**error**
72 FLOAT	**error**
72 {	**error**
72 }	**error**
72 INC	**error**
71 RETURN	**error**
71 >=	**error**
71 <=	**error**
71 DOT	**error**
71 $	**error**

71)	**error**
71 (	**error**
71 +	**error**
71 *	**error**
71 -	**error**
71 /	**error**
71 SEM	**error**
71 =	**error**
71 <	**error**
71 >	**error**
71 FOR	**error**
71 ELSE	**error**
71 STR	**error**
71 [	**error**
71]	**shift 84**
71 ID	**error**
71 IF	**error**
71 AND	**error**
71 DIGIT	**error**
71 INT	**error**
71 FLOAT	**error**
71 {	**error**
71 }	**error**
71 INC	**error**
70 RETURN	**error**
70 >=	**error**
70 <=	**error**
70 DOT	**error**

70 $	**error**
70)	**error**
70 (	**error**
70 +	**reduce multi_e -> multi_e / cast_e**
70 *	**reduce multi_e -> multi_e / cast_e**
70 -	**reduce multi_e -> multi_e / cast_e**
70 /	**reduce multi_e -> multi_e / cast_e**
70 SEM	**reduce multi_e -> multi_e / cast_e**
70 =	**error**
70 <	**error**
70 >	**error**
70 FOR	**error**
70 ELSE	**error**
70 STR	**error**
70 [	**error**
70]	**error**
70 ID	**error**
70 IF	**error**
70 AND	**error**
70 DIGIT	**error**
70 INT	**error**
70 FLOAT	**error**
70 {	**error**
70 }	**error**
70 INC	**error**
79 RETURN	**error**
79 >=	**error**
79 <=	**error**

79 DOT	**error**
79 $	**error**
79)	**reduce bool_e -> cast_e <= cast_e**
79 (	**error**
79 +	**error**
79 *	**error**
79 -	**error**
79 /	**error**
79 SEM	**reduce bool_e -> cast_e <= cast_e**
79 =	**error**
79 <	**error**
79 >	**error**
79 FOR	**error**
79 ELSE	**error**
79 STR	**error**
79 [	**error**
79]	**error**
79 ID	**error**
79 IF	**error**
79 AND	**error**
79 DIGIT	**error**
79 INT	**error**
79 FLOAT	**error**
79 {	**error**
79 }	**error**
79 INC	**error**
78 RETURN	**error**
78 >=	**error**

78 <=	**error**
78 DOT	**error**
78 $	**error**
78)	**reduce bool_e -> cast_e >= cast_e**
78 (	**error**
78 +	**error**
78 *	**error**
78 -	**error**
78 /	**error**
78 SEM	**reduce bool_e -> cast_e >= cast_e**
78 =	**error**
78 <	**error**
78 >	**error**
78 FOR	**error**
78 ELSE	**error**
78 STR	**error**
78 [	**error**
78]	**error**
78 ID	**error**
78 IF	**error**
78 AND	**error**
78 DIGIT	**error**
78 INT	**error**
78 FLOAT	**error**
78 {	**error**
78 }	**error**
78 INC	**error**

附录 ***3***

Goto 表

24 inc_e	**error**
24 cast_e	**error**
24 add_e	**error**
24 declarator_e	**error**
24 select_e	**error**
24 type	**error**
24 expression_list	**error**
24 assignment_e	**error**
24 bool_e	**error**
24 iterator_e	**error**
24 p	**error**
24 digit_list	**error**
24 multi_e	**error**
24 declarator	**error**
24 func_e	**error**
24 expression	**error**
24 declarator_list	**error**
25 inc_e	**error**
25 cast_e	**error**
25 add_e	**error**
25 declarator_e	**error**
25 select_e	**error**
25 type	**error**
25 expression_list	**error**
25 assignment_e	**error**

25 bool_e	**error**
25 iterator_e	**error**
25 p	**error**
25 digit_list	**error**
25 multi_e	**error**
25 declarator	**error**
25 func_e	**error**
25 expression	**error**
25 declarator_list	**error**
26 inc_e	**error**
26 cast_e	**error**
26 add_e	**error**
26 declarator_e	**error**
26 select_e	**error**
26 type	**error**
26 expression_list	**error**
26 assignment_e	**error**
26 bool_e	**error**
26 iterator_e	**error**
26 p	**error**
26 digit_list	**error**
26 multi_e	**error**
26 declarator	**error**
26 func_e	**error**
26 expression	**error**
26 declarator_list	**error**
27 inc_e	**error**
27 cast_e	**39**

27 add_e	**37**
27 declarator_e	**error**
27 select_e	**error**
27 type	**error**
27 expression_list	**error**
27 assignment_e	**error**
27 bool_e	**error**
27 iterator_e	**error**
27 p	**error**
27 digit_list	**error**
27 multi_e	**38**
27 declarator	**36**
27 func_e	**error**
27 expression	**error**
27 declarator_list	**error**
20 inc_e	**error**
20 cast_e	**error**
20 add_e	**error**
20 declarator_e	**error**
20 select_e	**error**
20 type	**error**
20 expression_list	**error**
20 assignment_e	**error**
20 bool_e	**error**
20 iterator_e	**error**
20 p	**error**
20 digit_list	**error**
20 multi_e	**error**

20 declarator	**error**
20 func_e	**error**
20 expression	**error**
20 declarator_list	**error**
21 inc_e	**error**
21 cast_e	**error**
21 add_e	**error**
21 declarator_e	**error**
21 select_e	**error**
21 type	**error**
21 expression_list	**error**
21 assignment_e	**error**
21 bool_e	**error**
21 iterator_e	**error**
21 p	**error**
21 digit_list	**error**
21 multi_e	**error**
21 declarator	**error**
21 func_e	**error**
21 expression	**error**
21 declarator_list	**error**
22 inc_e	**error**
22 cast_e	**error**
22 add_e	**error**
22 declarator_e	**error**
22 select_e	**error**
22 type	**error**
22 expression_list	**error**

22 assignment_e	**error**
22 bool_e	**error**
22 iterator_e	**error**
22 p	**error**
22 digit_list	**error**
22 multi_e	**error**
22 declarator	**error**
22 func_e	**error**
22 expression	**error**
22 declarator_list	**error**
23 inc_e	**error**
23 cast_e	**error**
23 add_e	**error**
23 declarator_e	**error**
23 select_e	**error**
23 type	**error**
23 expression_list	**error**
23 assignment_e	**error**
23 bool_e	**error**
23 iterator_e	**error**
23 p	**error**
23 digit_list	**error**
23 multi_e	**error**
23 declarator	**error**
23 func_e	**error**
23 expression	**error**
23 declarator_list	**error**
28 inc_e	**error**

28 cast_e	**error**
28 add_e	**error**
28 declarator_e	**error**
28 select_e	**error**
28 type	**error**
28 expression_list	**error**
28 assignment_e	**error**
28 bool_e	**error**
28 iterator_e	**error**
28 p	**error**
28 digit_list	**error**
28 multi_e	**error**
28 declarator	**error**
28 func_e	**error**
28 expression	**error**
28 declarator_list	**error**
29 inc_e	**error**
29 cast_e	**error**
29 add_e	**error**
29 declarator_e	**error**
29 select_e	**error**
29 type	**error**
29 expression_list	**error**
29 assignment_e	**error**
29 bool_e	**error**
29 iterator_e	**error**
29 p	**error**
29 digit_list	**error**

29 multi_e	**error**
29 declarator	**error**
29 func_e	**error**
29 expression	**error**
29 declarator_list	**error**
0 inc_e	**error**
0 cast_e	**error**
0 add_e	**error**
0 declarator_e	**error**
0 select_e	**error**
0 type	**error**
0 expression_list	**error**
0 assignment_e	**error**
0 bool_e	**error**
0 iterator_e	**error**
0 p	**1**
0 digit_list	**error**
0 multi_e	**error**
0 declarator	**error**
0 func_e	**error**
0 expression	**error**
0 declarator_list	**error**
4 inc_e	**error**
4 cast_e	**error**
4 add_e	**error**
4 declarator_e	**error**
4 select_e	**error**
4 type	**error**

4 expression_list	**error**
4 assignment_e	**error**
4 bool_e	**error**
4 iterator_e	**error**
4 p	**error**
4 digit_list	**error**
4 multi_e	**error**
4 declarator	**error**
4 func_e	**error**
4 expression	**error**
4 declarator_list	**error**
8 inc_e	**error**
8 cast_e	**error**
8 add_e	**error**
8 declarator_e	**11**
8 select_e	**13**
8 type	**9**
8 expression_list	**error**
8 assignment_e	**12**
8 bool_e	**error**
8 iterator_e	**14**
8 p	**error**
8 digit_list	**error**
8 multi_e	**error**
8 declarator	**10**
8 func_e	**15**
8 expression	**22**
8 declarator_list	**error**

59 inc_e	**error**
59 cast_e	**76**
59 add_e	**error**
59 declarator_e	**error**
59 select_e	**error**
59 type	**error**
59 expression_list	**error**
59 assignment_e	**error**
59 bool_e	**error**
59 iterator_e	**error**
59 p	**error**
59 digit_list	**error**
59 multi_e	**error**
59 declarator	**36**
59 func_e	**error**
59 expression	**error**
59 declarator_list	**error**
58 inc_e	**error**
58 cast_e	**error**
58 add_e	**error**
58 declarator_e	**error**
58 select_e	**error**
58 type	**error**
58 expression_list	**error**
58 assignment_e	**error**
58 bool_e	**error**
58 iterator_e	**error**
58 p	**error**

58 digit_list	**error**
58 multi_e	**error**
58 declarator	**error**
58 func_e	**error**
58 expression	**error**
58 declarator_list	**error**
55 inc_e	**error**
55 cast_e	**error**
55 add_e	**error**
55 declarator_e	**error**
55 select_e	**error**
55 type	**error**
55 expression_list	**error**
55 assignment_e	**error**
55 bool_e	**error**
55 iterator_e	**error**
55 p	**error**
55 digit_list	**error**
55 multi_e	**error**
55 declarator	**error**
55 func_e	**error**
55 expression	**error**
55 declarator_list	**error**
54 inc_e	**error**
54 cast_e	**70**
54 add_e	**error**
54 declarator_e	**error**
54 select_e	**error**

54 type	**error**
54 expression_list	**error**
54 assignment_e	**error**
54 bool_e	**error**
54 iterator_e	**error**
54 p	**error**
54 digit_list	**error**
54 multi_e	**error**
54 declarator	**36**
54 func_e	**error**
54 expression	**error**
54 declarator_list	**error**
57 inc_e	**error**
57 cast_e	**error**
57 add_e	**error**
57 declarator_e	**error**
57 select_e	**error**
57 type	**error**
57 expression_list	**error**
57 assignment_e	**error**
57 bool_e	**error**
57 iterator_e	**error**
57 p	**error**
57 digit_list	**error**
57 multi_e	**error**
57 declarator	**error**
57 func_e	**error**
57 expression	**error**

57 declarator_list	**error**
56 inc_e	**error**
56 cast_e	**error**
56 add_e	**error**
56 declarator_e	**error**
56 select_e	**error**
56 type	**error**
56 expression_list	**error**
56 assignment_e	**error**
56 bool_e	**error**
56 iterator_e	**error**
56 p	**error**
56 digit_list	**error**
56 multi_e	**error**
56 declarator	**72**
56 func_e	**error**
56 expression	**error**
56 declarator_list	**73**
51 inc_e	**error**
51 cast_e	**39**
51 add_e	**error**
51 declarator_e	**error**
51 select_e	**error**
51 type	**error**
51 expression_list	**error**
51 assignment_e	**error**
51 bool_e	**error**
51 iterator_e	**error**

51 p	**error**
51 digit_list	**error**
51 multi_e	**68**
51 declarator	**36**
51 func_e	**error**
51 expression	**error**
51 declarator_list	**error**
50 inc_e	**error**
50 cast_e	**39**
50 add_e	**error**
50 declarator_e	**error**
50 select_e	**error**
50 type	**error**
50 expression_list	**error**
50 assignment_e	**error**
50 bool_e	**error**
50 iterator_e	**error**
50 p	**error**
50 digit_list	**error**
50 multi_e	**67**
50 declarator	**36**
50 func_e	**error**
50 expression	**error**
50 declarator_list	**error**
53 inc_e	**error**
53 cast_e	**69**
53 add_e	**error**
53 declarator_e	**error**

53 select_e	**error**
53 type	**error**
53 expression_list	**error**
53 assignment_e	**error**
53 bool_e	**error**
53 iterator_e	**error**
53 p	**error**
53 digit_list	**error**
53 multi_e	**error**
53 declarator	**36**
53 func_e	**error**
53 expression	**error**
53 declarator_list	**error**
52 inc_e	**error**
52 cast_e	**error**
52 add_e	**error**
52 declarator_e	**error**
52 select_e	**error**
52 type	**error**
52 expression_list	**error**
52 assignment_e	**error**
52 bool_e	**error**
52 iterator_e	**error**
52 p	**error**
52 digit_list	**error**
52 multi_e	**error**
52 declarator	**error**
52 func_e	**error**

52 expression	**error**
52 declarator_list	**error**
88 inc_e	**error**
88 cast_e	**error**
88 add_e	**error**
88 declarator_e	**error**
88 select_e	**error**
88 type	**error**
88 expression_list	**error**
88 assignment_e	**error**
88 bool_e	**error**
88 iterator_e	**error**
88 p	**error**
88 digit_list	**error**
88 multi_e	**error**
88 declarator	**error**
88 func_e	**error**
88 expression	**error**
88 declarator_list	**error**
89 inc_e	**error**
89 cast_e	**error**
89 add_e	**error**
89 declarator_e	**error**
89 select_e	**error**
89 type	**error**
89 expression_list	**error**
89 assignment_e	**error**
89 bool_e	**error**

89 iterator_e	**error**
89 p	**error**
89 digit_list	**error**
89 multi_e	**error**
89 declarator	**error**
89 func_e	**error**
89 expression	**error**
89 declarator_list	**error**
82 inc_e	**error**
82 cast_e	**error**
82 add_e	**error**
82 declarator_e	**error**
82 select_e	**error**
82 type	**error**
82 expression_list	**error**
82 assignment_e	**error**
82 bool_e	**error**
82 iterator_e	**error**
82 p	**error**
82 digit_list	**error**
82 multi_e	**error**
82 declarator	**error**
82 func_e	**error**
82 expression	**error**
82 declarator_list	**error**
83 inc_e	**error**
83 cast_e	**error**
83 add_e	**error**

83 declarator_e	**error**
83 select_e	**error**
83 type	**error**
83 expression_list	**error**
83 assignment_e	**error**
83 bool_e	**error**
83 iterator_e	**error**
83 p	**error**
83 digit_list	**error**
83 multi_e	**error**
83 declarator	**error**
83 func_e	**error**
83 expression	**error**
83 declarator_list	**error**
80 inc_e	**error**
80 cast_e	**error**
80 add_e	**error**
80 declarator_e	**11**
80 select_e	**13**
80 type	**9**
80 expression_list	**87**
80 assignment_e	**12**
80 bool_e	**error**
80 iterator_e	**14**
80 p	**error**
80 digit_list	**error**
80 multi_e	**error**
80 declarator	**10**

80 func_e	**15**
80 expression	**7**
80 declarator_list	**error**
81 inc_e	**88**
81 cast_e	**error**
81 add_e	**error**
81 declarator_e	**error**
81 select_e	**error**
81 type	**error**
81 expression_list	**error**
81 assignment_e	**error**
81 bool_e	**error**
81 iterator_e	**error**
81 p	**error**
81 digit_list	**error**
81 multi_e	**error**
81 declarator	**error**
81 func_e	**error**
81 expression	**error**
81 declarator_list	**error**
86 inc_e	**error**
86 cast_e	**error**
86 add_e	**error**
86 declarator_e	**error**
86 select_e	**error**
86 type	**error**
86 expression_list	**error**
86 assignment_e	**error**

86 bool_e	**error**
86 iterator_e	**error**
86 p	**error**
86 digit_list	**error**
86 multi_e	**error**
86 declarator	**error**
86 func_e	**error**
86 expression	**error**
86 declarator_list	**error**
87 inc_e	**error**
87 cast_e	**error**
87 add_e	**error**
87 declarator_e	**11**
87 select_e	**13**
87 type	**9**
87 expression_list	**error**
87 assignment_e	**12**
87 bool_e	**error**
87 iterator_e	**14**
87 p	**error**
87 digit_list	**error**
87 multi_e	**error**
87 declarator	**10**
87 func_e	**15**
87 expression	**22**
87 declarator_list	**error**
84 inc_e	**error**
84 cast_e	**error**

84 add_e	**error**
84 declarator_e	**error**
84 select_e	**error**
84 type	**error**
84 expression_list	**error**
84 assignment_e	**error**
84 bool_e	**error**
84 iterator_e	**error**
84 p	**error**
84 digit_list	**error**
84 multi_e	**error**
84 declarator	**error**
84 func_e	**error**
84 expression	**error**
84 declarator_list	**error**
85 inc_e	**error**
85 cast_e	**error**
85 add_e	**error**
85 declarator_e	**error**
85 select_e	**error**
85 type	**error**
85 expression_list	**error**
85 assignment_e	**error**
85 bool_e	**error**
85 iterator_e	**error**
85 p	**error**
85 digit_list	**error**
85 multi_e	**error**

85 declarator	**error**
85 func_e	**error**
85 expression	**error**
85 declarator_list	**error**
3 inc_e	**error**
3 cast_e	**error**
3 add_e	**error**
3 declarator_e	**error**
3 select_e	**error**
3 type	**error**
3 expression_list	**error**
3 assignment_e	**error**
3 bool_e	**error**
3 iterator_e	**error**
3 p	**error**
3 digit_list	**error**
3 multi_e	**error**
3 declarator	**error**
3 func_e	**error**
3 expression	**error**
3 declarator_list	**error**
7 inc_e	**error**
7 cast_e	**error**
7 add_e	**error**
7 declarator_e	**error**
7 select_e	**error**
7 type	**error**
7 expression_list	**error**

7 assignment_e	**error**
7 bool_e	**error**
7 iterator_e	**error**
7 p	**error**
7 digit_list	**error**
7 multi_e	**error**
7 declarator	**error**
7 func_e	**error**
7 expression	**error**
7 declarator_list	**error**
102 inc_e	**error**
102 cast_e	**error**
102 add_e	**error**
102 declarator_e	**11**
102 select_e	**13**
102 type	**9**
102 expression_list	**error**
102 assignment_e	**12**
102 bool_e	**error**
102 iterator_e	**14**
102 p	**error**
102 digit_list	**error**
102 multi_e	**error**
102 declarator	**10**
102 func_e	**15**
102 expression	**22**
102 declarator_list	**error**
103 inc_e	**error**

103 cast_e	**error**
103 add_e	**error**
103 declarator_e	**error**
103 select_e	**error**
103 type	**error**
103 expression_list	**error**
103 assignment_e	**error**
103 bool_e	**error**
103 iterator_e	**error**
103 p	**error**
103 digit_list	**error**
103 multi_e	**error**
103 declarator	**error**
103 func_e	**error**
103 expression	**error**
103 declarator_list	**error**
100 inc_e	**error**
100 cast_e	**error**
100 add_e	**error**
100 declarator_e	**11**
100 select_e	**13**
100 type	**9**
100 expression_list	**102**
100 assignment_e	**12**
100 bool_e	**error**
100 iterator_e	**14**
100 p	**error**
100 digit_list	**error**

100 multi_e	**error**
100 declarator	**10**
100 func_e	**15**
100 expression	**7**
100 declarator_list	**error**
101 inc_e	**error**
101 cast_e	**error**
101 add_e	**error**
101 declarator_e	**11**
101 select_e	**13**
101 type	**9**
101 expression_list	**error**
101 assignment_e	**12**
101 bool_e	**error**
101 iterator_e	**14**
101 p	**error**
101 digit_list	**error**
101 multi_e	**error**
101 declarator	**10**
101 func_e	**15**
101 expression	**22**
101 declarator_list	**error**
104 inc_e	**error**
104 cast_e	**error**
104 add_e	**error**
104 declarator_e	**error**
104 select_e	**error**
104 type	**error**

104 expression_list	**error**
104 assignment_e	**error**
104 bool_e	**error**
104 iterator_e	**error**
104 p	**error**
104 digit_list	**error**
104 multi_e	**error**
104 declarator	**error**
104 func_e	**error**
104 expression	**error**
104 declarator_list	**error**
39 inc_e	**error**
39 cast_e	**error**
39 add_e	**error**
39 declarator_e	**error**
39 select_e	**error**
39 type	**error**
39 expression_list	**error**
39 assignment_e	**error**
39 bool_e	**error**
39 iterator_e	**error**
39 p	**error**
39 digit_list	**error**
39 multi_e	**error**
39 declarator	**error**
39 func_e	**error**
39 expression	**error**
39 declarator_list	**error**

38 inc_e	**error**
38 cast_e	**error**
38 add_e	**error**
38 declarator_e	**error**
38 select_e	**error**
38 type	**error**
38 expression_list	**error**
38 assignment_e	**error**
38 bool_e	**error**
38 iterator_e	**error**
38 p	**error**
38 digit_list	**error**
38 multi_e	**error**
38 declarator	**error**
38 func_e	**error**
38 expression	**error**
38 declarator_list	**error**
33 inc_e	**error**
33 cast_e	**error**
33 add_e	**error**
33 declarator_e	**error**
33 select_e	**error**
33 type	**error**
33 expression_list	**error**
33 assignment_e	**error**
33 bool_e	**error**
33 iterator_e	**error**
33 p	**error**

33 digit_list	**error**
33 multi_e	**error**
33 declarator	**error**
33 func_e	**error**
33 expression	**error**
33 declarator_list	**error**
32 inc_e	**error**
32 cast_e	**error**
32 add_e	**error**
32 declarator_e	**error**
32 select_e	**error**
32 type	**error**
32 expression_list	**error**
32 assignment_e	**error**
32 bool_e	**error**
32 iterator_e	**error**
32 p	**error**
32 digit_list	**error**
32 multi_e	**error**
32 declarator	**error**
32 func_e	**error**
32 expression	**error**
32 declarator_list	**error**
31 inc_e	**error**
31 cast_e	**error**
31 add_e	**error**
31 declarator_e	**error**
31 select_e	**error**

31 type	**error**
31 expression_list	**error**
31 assignment_e	**46**
31 bool_e	**error**
31 iterator_e	**error**
31 p	**error**
31 digit_list	**error**
31 multi_e	**error**
31 declarator	**10**
31 func_e	**error**
31 expression	**error**
31 declarator_list	**error**
30 inc_e	**error**
30 cast_e	**44**
30 add_e	**error**
30 declarator_e	**error**
30 select_e	**error**
30 type	**error**
30 expression_list	**error**
30 assignment_e	**error**
30 bool_e	**45**
30 iterator_e	**error**
30 p	**error**
30 digit_list	**error**
30 multi_e	**error**
30 declarator	**36**
30 func_e	**error**
30 expression	**error**

30 declarator_list	**error**
37 inc_e	**error**
37 cast_e	**error**
37 add_e	**error**
37 declarator_e	**error**
37 select_e	**error**
37 type	**error**
37 expression_list	**error**
37 assignment_e	**error**
37 bool_e	**error**
37 iterator_e	**error**
37 p	**error**
37 digit_list	**error**
37 multi_e	**error**
37 declarator	**error**
37 func_e	**error**
37 expression	**error**
37 declarator_list	**error**
36 inc_e	**error**
36 cast_e	**error**
36 add_e	**error**
36 declarator_e	**error**
36 select_e	**error**
36 type	**error**
36 expression_list	**error**
36 assignment_e	**error**
36 bool_e	**error**
36 iterator_e	**error**

36 p	**error**
36 digit_list	**error**
36 multi_e	**error**
36 declarator	**error**
36 func_e	**error**
36 expression	**error**
36 declarator_list	**error**
35 inc_e	**error**
35 cast_e	**error**
35 add_e	**error**
35 declarator_e	**error**
35 select_e	**error**
35 type	**error**
35 expression_list	**error**
35 assignment_e	**error**
35 bool_e	**error**
35 iterator_e	**error**
35 p	**error**
35 digit_list	**error**
35 multi_e	**error**
35 declarator	**error**
35 func_e	**error**
35 expression	**error**
35 declarator_list	**error**
34 inc_e	**error**
34 cast_e	**error**
34 add_e	**error**
34 declarator_e	**error**

34 select_e	**error**
34 type	**error**
34 expression_list	**error**
34 assignment_e	**error**
34 bool_e	**error**
34 iterator_e	**error**
34 p	**error**
34 digit_list	**error**
34 multi_e	**error**
34 declarator	**49**
34 func_e	**error**
34 expression	**error**
34 declarator_list	**error**
60 inc_e	**error**
60 cast_e	**77**
60 add_e	**error**
60 declarator_e	**error**
60 select_e	**error**
60 type	**error**
60 expression_list	**error**
60 assignment_e	**error**
60 bool_e	**error**
60 iterator_e	**error**
60 p	**error**
60 digit_list	**error**
60 multi_e	**error**
60 declarator	**36**
60 func_e	**error**

60 expression	**error**
60 declarator_list	**error**
61 inc_e	**error**
61 cast_e	**78**
61 add_e	**error**
61 declarator_e	**error**
61 select_e	**error**
61 type	**error**
61 expression_list	**error**
61 assignment_e	**error**
61 bool_e	**error**
61 iterator_e	**error**
61 p	**error**
61 digit_list	**error**
61 multi_e	**error**
61 declarator	**36**
61 func_e	**error**
61 expression	**error**
61 declarator_list	**error**
62 inc_e	**error**
62 cast_e	**79**
62 add_e	**error**
62 declarator_e	**error**
62 select_e	**error**
62 type	**error**
62 expression_list	**error**
62 assignment_e	**error**
62 bool_e	**error**

62 iterator_e	**error**
62 p	**error**
62 digit_list	**error**
62 multi_e	**error**
62 declarator	**36**
62 func_e	**error**
62 expression	**error**
62 declarator_list	**error**
63 inc_e	**error**
63 cast_e	**error**
63 add_e	**error**
63 declarator_e	**error**
63 select_e	**error**
63 type	**error**
63 expression_list	**error**
63 assignment_e	**error**
63 bool_e	**error**
63 iterator_e	**error**
63 p	**error**
63 digit_list	**error**
63 multi_e	**error**
63 declarator	**error**
63 func_e	**error**
63 expression	**error**
63 declarator_list	**error**
64 inc_e	**error**
64 cast_e	**error**
64 add_e	**error**

64 declarator_e	**error**
64 select_e	**error**
64 type	**error**
64 expression_list	**error**
64 assignment_e	**error**
64 bool_e	**error**
64 iterator_e	**error**
64 p	**error**
64 digit_list	**error**
64 multi_e	**error**
64 declarator	**error**
64 func_e	**error**
64 expression	**error**
64 declarator_list	**error**
65 inc_e	**error**
65 cast_e	**error**
65 add_e	**error**
65 declarator_e	**error**
65 select_e	**error**
65 type	**error**
65 expression_list	**error**
65 assignment_e	**error**
65 bool_e	**error**
65 iterator_e	**error**
65 p	**error**
65 digit_list	**error**
65 multi_e	**error**
65 declarator	**error**

65 func_e	**error**
65 expression	**error**
65 declarator_list	**error**
66 inc_e	**error**
66 cast_e	**error**
66 add_e	**error**
66 declarator_e	**error**
66 select_e	**error**
66 type	**error**
66 expression_list	**error**
66 assignment_e	**error**
66 bool_e	**error**
66 iterator_e	**error**
66 p	**error**
66 digit_list	**error**
66 multi_e	**error**
66 declarator	**error**
66 func_e	**error**
66 expression	**error**
66 declarator_list	**error**
67 inc_e	**error**
67 cast_e	**error**
67 add_e	**error**
67 declarator_e	**error**
67 select_e	**error**
67 type	**error**
67 expression_list	**error**
67 assignment_e	**error**

67 bool_e	**error**
67 iterator_e	**error**
67 p	**error**
67 digit_list	**error**
67 multi_e	**error**
67 declarator	**error**
67 func_e	**error**
67 expression	**error**
67 declarator_list	**error**
68 inc_e	**error**
68 cast_e	**error**
68 add_e	**error**
68 declarator_e	**error**
68 select_e	**error**
68 type	**error**
68 expression_list	**error**
68 assignment_e	**error**
68 bool_e	**error**
68 iterator_e	**error**
68 p	**error**
68 digit_list	**error**
68 multi_e	**error**
68 declarator	**error**
68 func_e	**error**
68 expression	**error**
68 declarator_list	**error**
69 inc_e	**error**
69 cast_e	**error**

69 add_e	**error**
69 declarator_e	**error**
69 select_e	**error**
69 type	**error**
69 expression_list	**error**
69 assignment_e	**error**
69 bool_e	**error**
69 iterator_e	**error**
69 p	**error**
69 digit_list	**error**
69 multi_e	**error**
69 declarator	**error**
69 func_e	**error**
69 expression	**error**
69 declarator_list	**error**
2 inc_e	**error**
2 cast_e	**error**
2 add_e	**error**
2 declarator_e	**error**
2 select_e	**error**
2 type	**error**
2 expression_list	**error**
2 assignment_e	**error**
2 bool_e	**error**
2 iterator_e	**error**
2 p	**error**
2 digit_list	**error**
2 multi_e	**error**

2 declarator	**error**
2 func_e	**error**
2 expression	**error**
2 declarator_list	**error**
6 inc_e	**error**
6 cast_e	**error**
6 add_e	**error**
6 declarator_e	**11**
6 select_e	**13**
6 type	**9**
6 expression_list	**8**
6 assignment_e	**12**
6 bool_e	**error**
6 iterator_e	**14**
6 p	**error**
6 digit_list	**error**
6 multi_e	**error**
6 declarator	**10**
6 func_e	**15**
6 expression	**7**
6 declarator_list	**error**
99 inc_e	**error**
99 cast_e	**error**
99 add_e	**error**
99 declarator_e	**11**
99 select_e	**13**
99 type	**9**
99 expression_list	**101**

99 assignment_e	**12**
99 bool_e	**error**
99 iterator_e	**14**
99 p	**error**
99 digit_list	**error**
99 multi_e	**error**
99 declarator	**10**
99 func_e	**15**
99 expression	**7**
99 declarator_list	**error**
98 inc_e	**error**
98 cast_e	**error**
98 add_e	**error**
98 declarator_e	**error**
98 select_e	**error**
98 type	**error**
98 expression_list	**error**
98 assignment_e	**error**
98 bool_e	**error**
98 iterator_e	**error**
98 p	**error**
98 digit_list	**error**
98 multi_e	**error**
98 declarator	**error**
98 func_e	**error**
98 expression	**error**
98 declarator_list	**error**
91 inc_e	**error**

91 cast_e	**error**
91 add_e	**error**
91 declarator_e	**error**
91 select_e	**error**
91 type	**error**
91 expression_list	**error**
91 assignment_e	**error**
91 bool_e	**error**
91 iterator_e	**error**
91 p	**error**
91 digit_list	**error**
91 multi_e	**error**
91 declarator	**error**
91 func_e	**error**
91 expression	**error**
91 declarator_list	**error**
90 inc_e	**error**
90 cast_e	**error**
90 add_e	**error**
90 declarator_e	**error**
90 select_e	**error**
90 type	**error**
90 expression_list	**error**
90 assignment_e	**error**
90 bool_e	**error**
90 iterator_e	**error**
90 p	**error**
90 digit_list	**error**

90 multi_e	**error**
90 declarator	**error**
90 func_e	**error**
90 expression	**error**
90 declarator_list	**error**
93 inc_e	**error**
93 cast_e	**error**
93 add_e	**error**
93 declarator_e	**error**
93 select_e	**error**
93 type	**error**
93 expression_list	**error**
93 assignment_e	**error**
93 bool_e	**error**
93 iterator_e	**error**
93 p	**error**
93 digit_list	**error**
93 multi_e	**error**
93 declarator	**error**
93 func_e	**error**
93 expression	**error**
93 declarator_list	**error**
92 inc_e	**error**
92 cast_e	**error**
92 add_e	**error**
92 declarator_e	**error**
92 select_e	**error**
92 type	**error**

92 expression_list	**error**
92 assignment_e	**error**
92 bool_e	**error**
92 iterator_e	**error**
92 p	**error**
92 digit_list	**error**
92 multi_e	**error**
92 declarator	**error**
92 func_e	**error**
92 expression	**error**
92 declarator_list	**error**
95 inc_e	**error**
95 cast_e	**error**
95 add_e	**error**
95 declarator_e	**error**
95 select_e	**error**
95 type	**error**
95 expression_list	**error**
95 assignment_e	**error**
95 bool_e	**error**
95 iterator_e	**error**
95 p	**error**
95 digit_list	**error**
95 multi_e	**error**
95 declarator	**error**
95 func_e	**error**
95 expression	**error**
95 declarator_list	**error**

94 inc_e	**error**
94 cast_e	**error**
94 add_e	**error**
94 declarator_e	**error**
94 select_e	**error**
94 type	**error**
94 expression_list	**error**
94 assignment_e	**error**
94 bool_e	**error**
94 iterator_e	**error**
94 p	**error**
94 digit_list	**error**
94 multi_e	**error**
94 declarator	**error**
94 func_e	**error**
94 expression	**error**
94 declarator_list	**error**
97 inc_e	**error**
97 cast_e	**error**
97 add_e	**error**
97 declarator_e	**error**
97 select_e	**error**
97 type	**error**
97 expression_list	**error**
97 assignment_e	**error**
97 bool_e	**error**
97 iterator_e	**error**
97 p	**error**

97 digit_list	**error**
97 multi_e	**error**
97 declarator	**error**
97 func_e	**error**
97 expression	**error**
97 declarator_list	**error**
96 inc_e	**error**
96 cast_e	**error**
96 add_e	**error**
96 declarator_e	**error**
96 select_e	**error**
96 type	**error**
96 expression_list	**error**
96 assignment_e	**error**
96 bool_e	**error**
96 iterator_e	**error**
96 p	**error**
96 digit_list	**error**
96 multi_e	**error**
96 declarator	**error**
96 func_e	**error**
96 expression	**error**
96 declarator_list	**error**
11 inc_e	**error**
11 cast_e	**error**
11 add_e	**error**
11 declarator_e	**error**
11 select_e	**error**

11 type	**error**
11 expression_list	**error**
11 assignment_e	**error**
11 bool_e	**error**
11 iterator_e	**error**
11 p	**error**
11 digit_list	**error**
11 multi_e	**error**
11 declarator	**error**
11 func_e	**error**
11 expression	**error**
11 declarator_list	**error**
10 inc_e	**error**
10 cast_e	**error**
10 add_e	**error**
10 declarator_e	**error**
10 select_e	**error**
10 type	**error**
10 expression_list	**error**
10 assignment_e	**error**
10 bool_e	**error**
10 iterator_e	**error**
10 p	**error**
10 digit_list	**error**
10 multi_e	**error**
10 declarator	**error**
10 func_e	**error**
10 expression	**error**

10 declarator_list	**error**
13 inc_e	**error**
13 cast_e	**error**
13 add_e	**error**
13 declarator_e	**error**
13 select_e	**error**
13 type	**error**
13 expression_list	**error**
13 assignment_e	**error**
13 bool_e	**error**
13 iterator_e	**error**
13 p	**error**
13 digit_list	**error**
13 multi_e	**error**
13 declarator	**error**
13 func_e	**error**
13 expression	**error**
13 declarator_list	**error**
12 inc_e	**error**
12 cast_e	**error**
12 add_e	**error**
12 declarator_e	**error**
12 select_e	**error**
12 type	**error**
12 expression_list	**error**
12 assignment_e	**error**
12 bool_e	**error**
12 iterator_e	**error**

12 p	**error**
12 digit_list	**error**
12 multi_e	**error**
12 declarator	**error**
12 func_e	**error**
12 expression	**error**
12 declarator_list	**error**
15 inc_e	**error**
15 cast_e	**error**
15 add_e	**error**
15 declarator_e	**error**
15 select_e	**error**
15 type	**error**
15 expression_list	**error**
15 assignment_e	**error**
15 bool_e	**error**
15 iterator_e	**error**
15 p	**error**
15 digit_list	**error**
15 multi_e	**error**
15 declarator	**error**
15 func_e	**error**
15 expression	**error**
15 declarator_list	**error**
14 inc_e	**error**
14 cast_e	**error**
14 add_e	**error**
14 declarator_e	**error**

14 select_e	**error**
14 type	**error**
14 expression_list	**error**
14 assignment_e	**error**
14 bool_e	**error**
14 iterator_e	**error**
14 p	**error**
14 digit_list	**error**
14 multi_e	**error**
14 declarator	**error**
14 func_e	**error**
14 expression	**error**
14 declarator_list	**error**
17 inc_e	**error**
17 cast_e	**error**
17 add_e	**error**
17 declarator_e	**error**
17 select_e	**error**
17 type	**error**
17 expression_list	**error**
17 assignment_e	**error**
17 bool_e	**error**
17 iterator_e	**error**
17 p	**error**
17 digit_list	**error**
17 multi_e	**error**
17 declarator	**error**
17 func_e	**error**

17 expression	**error**
17 declarator_list	**error**
16 inc_e	**error**
16 cast_e	**error**
16 add_e	**error**
16 declarator_e	**error**
16 select_e	**error**
16 type	**error**
16 expression_list	**error**
16 assignment_e	**error**
16 bool_e	**error**
16 iterator_e	**error**
16 p	**error**
16 digit_list	**error**
16 multi_e	**error**
16 declarator	**error**
16 func_e	**error**
16 expression	**error**
16 declarator_list	**error**
19 inc_e	**error**
19 cast_e	**error**
19 add_e	**error**
19 declarator_e	**error**
19 select_e	**error**
19 type	**error**
19 expression_list	**error**
19 assignment_e	**error**
19 bool_e	**error**

19 iterator_e	**error**
19 p	**error**
19 digit_list	**error**
19 multi_e	**error**
19 declarator	**error**
19 func_e	**error**
19 expression	**error**
19 declarator_list	**error**
18 inc_e	**error**
18 cast_e	**error**
18 add_e	**error**
18 declarator_e	**error**
18 select_e	**error**
18 type	**error**
18 expression_list	**error**
18 assignment_e	**error**
18 bool_e	**error**
18 iterator_e	**error**
18 p	**error**
18 digit_list	**error**
18 multi_e	**error**
18 declarator	**error**
18 func_e	**error**
18 expression	**error**
18 declarator_list	**error**
48 inc_e	**error**
48 cast_e	**error**
48 add_e	**error**

48 declarator_e	**error**
48 select_e	**error**
48 type	**error**
48 expression_list	**error**
48 assignment_e	**error**
48 bool_e	**error**
48 iterator_e	**error**
48 p	**error**
48 digit_list	**65**
48 multi_e	**error**
48 declarator	**error**
48 func_e	**error**
48 expression	**error**
48 declarator_list	**error**
49 inc_e	**error**
49 cast_e	**error**
49 add_e	**error**
49 declarator_e	**error**
49 select_e	**error**
49 type	**error**
49 expression_list	**error**
49 assignment_e	**error**
49 bool_e	**error**
49 iterator_e	**error**
49 p	**error**
49 digit_list	**error**
49 multi_e	**error**
49 declarator	**error**

49 func_e	**error**
49 expression	**error**
49 declarator_list	**error**
46 inc_e	**error**
46 cast_e	**44**
46 add_e	**error**
46 declarator_e	**error**
46 select_e	**error**
46 type	**error**
46 expression_list	**error**
46 assignment_e	**error**
46 bool_e	**64**
46 iterator_e	**error**
46 p	**error**
46 digit_list	**error**
46 multi_e	**error**
46 declarator	**36**
46 func_e	**error**
46 expression	**error**
46 declarator_list	**error**
47 inc_e	**error**
47 cast_e	**error**
47 add_e	**error**
47 declarator_e	**error**
47 select_e	**error**
47 type	**error**
47 expression_list	**error**
47 assignment_e	**error**

47 bool_e	**error**
47 iterator_e	**error**
47 p	**error**
47 digit_list	**error**
47 multi_e	**error**
47 declarator	**error**
47 func_e	**error**
47 expression	**error**
47 declarator_list	**error**
44 inc_e	**error**
44 cast_e	**error**
44 add_e	**error**
44 declarator_e	**error**
44 select_e	**error**
44 type	**error**
44 expression_list	**error**
44 assignment_e	**error**
44 bool_e	**error**
44 iterator_e	**error**
44 p	**error**
44 digit_list	**error**
44 multi_e	**error**
44 declarator	**error**
44 func_e	**error**
44 expression	**error**
44 declarator_list	**error**
45 inc_e	**error**
45 cast_e	**error**

45 add_e	**error**
45 declarator_e	**error**
45 select_e	**error**
45 type	**error**
45 expression_list	**error**
45 assignment_e	**error**
45 bool_e	**error**
45 iterator_e	**error**
45 p	**error**
45 digit_list	**error**
45 multi_e	**error**
45 declarator	**error**
45 func_e	**error**
45 expression	**error**
45 declarator_list	**error**
42 inc_e	**error**
42 cast_e	**error**
42 add_e	**error**
42 declarator_e	**error**
42 select_e	**error**
42 type	**error**
42 expression_list	**error**
42 assignment_e	**error**
42 bool_e	**error**
42 iterator_e	**error**
42 p	**error**
42 digit_list	**error**
42 multi_e	**error**

42 declarator	**error**
42 func_e	**error**
42 expression	**error**
42 declarator_list	**error**
43 inc_e	**error**
43 cast_e	**error**
43 add_e	**error**
43 declarator_e	**error**
43 select_e	**error**
43 type	**error**
43 expression_list	**error**
43 assignment_e	**error**
43 bool_e	**error**
43 iterator_e	**error**
43 p	**error**
43 digit_list	**error**
43 multi_e	**error**
43 declarator	**error**
43 func_e	**error**
43 expression	**error**
43 declarator_list	**error**
40 inc_e	**error**
40 cast_e	**error**
40 add_e	**error**
40 declarator_e	**error**
40 select_e	**error**
40 type	**error**
40 expression_list	**error**

40 assignment_e	**error**
40 bool_e	**error**
40 iterator_e	**error**
40 p	**error**
40 digit_list	**error**
40 multi_e	**error**
40 declarator	**error**
40 func_e	**error**
40 expression	**error**
40 declarator_list	**error**
41 inc_e	**error**
41 cast_e	**error**
41 add_e	**error**
41 declarator_e	**error**
41 select_e	**error**
41 type	**error**
41 expression_list	**error**
41 assignment_e	**error**
41 bool_e	**error**
41 iterator_e	**error**
41 p	**error**
41 digit_list	**error**
41 multi_e	**error**
41 declarator	**error**
41 func_e	**error**
41 expression	**error**
41 declarator_list	**error**
1 inc_e	**error**

1 cast_e	**error**
1 add_e	**error**
1 declarator_e	**error**
1 select_e	**error**
1 type	**error**
1 expression_list	**error**
1 assignment_e	**error**
1 bool_e	**error**
1 iterator_e	**error**
1 p	**error**
1 digit_list	**error**
1 multi_e	**error**
1 declarator	**error**
1 func_e	**error**
1 expression	**error**
1 declarator_list	**error**
5 inc_e	**error**
5 cast_e	**error**
5 add_e	**error**
5 declarator_e	**error**
5 select_e	**error**
5 type	**error**
5 expression_list	**error**
5 assignment_e	**error**
5 bool_e	**error**
5 iterator_e	**error**
5 p	**error**
5 digit_list	**error**

5 multi_e	**error**
5 declarator	**error**
5 func_e	**error**
5 expression	**error**
5 declarator_list	**error**
9 inc_e	**error**
9 cast_e	**error**
9 add_e	**error**
9 declarator_e	**error**
9 select_e	**error**
9 type	**error**
9 expression_list	**error**
9 assignment_e	**error**
9 bool_e	**error**
9 iterator_e	**error**
9 p	**error**
9 digit_list	**error**
9 multi_e	**error**
9 declarator	**24**
9 func_e	**error**
9 expression	**error**
9 declarator_list	**25**
77 inc_e	**error**
77 cast_e	**error**
77 add_e	**error**
77 declarator_e	**error**
77 select_e	**error**
77 type	**error**

77 expression_list	**error**
77 assignment_e	**error**
77 bool_e	**error**
77 iterator_e	**error**
77 p	**error**
77 digit_list	**error**
77 multi_e	**error**
77 declarator	**error**
77 func_e	**error**
77 expression	**error**
77 declarator_list	**error**
76 inc_e	**error**
76 cast_e	**error**
76 add_e	**error**
76 declarator_e	**error**
76 select_e	**error**
76 type	**error**
76 expression_list	**error**
76 assignment_e	**error**
76 bool_e	**error**
76 iterator_e	**error**
76 p	**error**
76 digit_list	**error**
76 multi_e	**error**
76 declarator	**error**
76 func_e	**error**
76 expression	**error**
76 declarator_list	**error**

75 inc_e	**error**
75 cast_e	**error**
75 add_e	**error**
75 declarator_e	**error**
75 select_e	**error**
75 type	**error**
75 expression_list	**error**
75 assignment_e	**error**
75 bool_e	**error**
75 iterator_e	**error**
75 p	**error**
75 digit_list	**error**
75 multi_e	**error**
75 declarator	**error**
75 func_e	**error**
75 expression	**error**
75 declarator_list	**error**
74 inc_e	**error**
74 cast_e	**error**
74 add_e	**error**
74 declarator_e	**error**
74 select_e	**error**
74 type	**error**
74 expression_list	**error**
74 assignment_e	**error**
74 bool_e	**error**
74 iterator_e	**error**
74 p	**error**

74 digit_list	**error**
74 multi_e	**error**
74 declarator	**error**
74 func_e	**error**
74 expression	**error**
74 declarator_list	**error**
73 inc_e	**error**
73 cast_e	**error**
73 add_e	**error**
73 declarator_e	**error**
73 select_e	**error**
73 type	**error**
73 expression_list	**error**
73 assignment_e	**error**
73 bool_e	**error**
73 iterator_e	**error**
73 p	**error**
73 digit_list	**error**
73 multi_e	**error**
73 declarator	**error**
73 func_e	**error**
73 expression	**error**
73 declarator_list	**error**
72 inc_e	**error**
72 cast_e	**error**
72 add_e	**error**
72 declarator_e	**error**
72 select_e	**error**

72 type	**error**
72 expression_list	**error**
72 assignment_e	**error**
72 bool_e	**error**
72 iterator_e	**error**
72 p	**error**
72 digit_list	**error**
72 multi_e	**error**
72 declarator	**error**
72 func_e	**error**
72 expression	**error**
72 declarator_list	**error**
71 inc_e	**error**
71 cast_e	**error**
71 add_e	**error**
71 declarator_e	**error**
71 select_e	**error**
71 type	**error**
71 expression_list	**error**
71 assignment_e	**error**
71 bool_e	**error**
71 iterator_e	**error**
71 p	**error**
71 digit_list	**error**
71 multi_e	**error**
71 declarator	**error**
71 func_e	**error**
71 expression	**error**

71 declarator_list	**error**
70 inc_e	**error**
70 cast_e	**error**
70 add_e	**error**
70 declarator_e	**error**
70 select_e	**error**
70 type	**error**
70 expression_list	**error**
70 assignment_e	**error**
70 bool_e	**error**
70 iterator_e	**error**
70 p	**error**
70 digit_list	**error**
70 multi_e	**error**
70 declarator	**error**
70 func_e	**error**
70 expression	**error**
70 declarator_list	**error**
79 inc_e	**error**
79 cast_e	**error**
79 add_e	**error**
79 declarator_e	**error**
79 select_e	**error**
79 type	**error**
79 expression_list	**error**
79 assignment_e	**error**
79 bool_e	**error**
79 iterator_e	**error**

79 p	**error**
79 digit_list	**error**
79 multi_e	**error**
79 declarator	**error**
79 func_e	**error**
79 expression	**error**
79 declarator_list	**error**
78 inc_e	**error**
78 cast_e	**error**
78 add_e	**error**
78 declarator_e	**error**
78 select_e	**error**
78 type	**error**
78 expression_list	**error**
78 assignment_e	**error**
78 bool_e	**error**
78 iterator_e	**error**
78 p	**error**
78 digit_list	**error**
78 multi_e	**error**
78 declarator	**error**
78 func_e	**error**
78 expression	**error**
78 declarator_list	**error**