

第 5 章语义分析中间代码生成习题答案

1. 写出下列源程序经过编译后所生成的虚拟机代码，并填写其符号表。

```
const pi=3;
var r,s;
procedure area;
  var r1;
  begin
    r1:=r*r;s:=pi*r1
  end;
begin
  read(r);
  if r>10 then call area;
  write(s)
end.
```

答案：

生成的虚拟机代码如下：

0	jmp	0	12
1	jmp	0	2
2	ini	0	4
3	lod	1	3
4	lod	1	3
5	opr	0	4
6	sto	0	3
7	lit	0	3
8	lod	0	3
9	opr	0	4
10	sto	1	4
11	opr	0	0
12	ini	0	5
13	opr	0	16
14	sto	0	3
15	lod	0	3
16	lit	0	10
17	opr	0	12
18	jpc	0	20
19	cal	0	2
20	lod	0	4
21	opr	0	14
22	opr	0	15
23	opr	0	0

符号表如下：

name	kind	val / level	adr	size
pi	const	3		
r	var	0	3	
s	var	0	4	
area	pro	0	2	4
r1	var	1	3	

2. 写出下列源程序经过编译后所生成的虚拟机代码，并填写其符号表。

```
var x,y,z;  
procedure gcd;  
  var f,g;  
  begin  
    f:=x;  
    g:=y;  
    while f#g do  
      begin  
        if f<g then  
          g:=g-f;  
        if g<f then  
          f:=f-g;  
        end;  
      z:=f  
    end;  
  begin  
    x:=84;y:=36;call gcd;write(z)  
  end.
```

答案：

生成的虚拟机代码如下：

0	jmp	0	31
1	jmp	0	2
2	ini	0	5
3	lod	1	3
4	sto	0	3
5	lod	1	4
6	sto	0	4
7	lod	0	3
8	lod	0	4
9	opr	0	9
10	jpc	0	28
11	lod	0	3
12	lod	0	4
13	opr	0	10
14	jpc	0	19
15	lod	0	4
16	lod	0	3
17	opr	0	3
18	sto	0	4
19	lod	0	4
20	lod	0	3
21	opr	0	10
22	jpc	0	27
23	lod	0	3
24	lod	0	4
25	opr	0	3
26	sto	0	3
27	jmp	0	7
28	lod	0	3
29	sto	1	5
30	opr	0	0
31	ini	0	6
32	lit	0	84
33	sto	0	3
34	lit	0	36
35	sto	0	4
36	cal	0	2
37	lod	0	5
38	opr	0	14
39	opr	0	15
40	opr	0	0

符号表如下：

name	kind	val / level	adr	size
x	var	0	3	
y	var	0	4	
z	var	0	5	
gcd	pro	0	2	5
f	var	1	3	
g	var	1	4	

3. 写出下列源程序经过编译后所生成的虚拟机代码，并填写其符号表。

```
const m=7,n=85;
var x,y,z,q,r;
procedure multiply;
  var a,b;
  begin
    a:=x;
    b:=y;
    z:=0;
    while b>0 do
      begin
        if odd b then
          z:=z+a;
          a:=2*a;
          b:=b/2;
        end
      end;
end;

procedure divide;
  var w;
  begin
    r:=x;
    q:=0;
    w:=y;
    while w<=r do
      w:=2*w;
    while w>y do
      begin
        q:=2*q;
        w:=w/2;
        if w<=r then
          begin
            r:=r-w;
            q:=q+1
          end
        end
      end;
end;

procedure gcd;
  var f,g;
  begin
    f:=x;
    g:=y;
```

生成的虚拟机代码如下：

(注意：0 1 30 73 四条指令生成时没有输出到文件中)

0	jmp	0	103
1	jmp	0	2
2	ini	0	5
3	lod	1	3
4	sto	0	3
5	lod	1	4
6	sto	0	4
7	lit	0	0
8	sto	1	5
9	lod	0	4
10	lit	0	0
11	opr	0	12
12	jpc	0	29
13	lod	0	4
14	opr	0	6
15	jpc	0	20
16	lod	1	5
17	lod	0	3
18	opr	0	2
19	sto	1	5
20	lit	0	2
21	lod	0	3
22	opr	0	4
23	sto	0	3
24	lod	0	4
25	lit	0	2
26	opr	0	5
27	sto	0	4
28	jmp	0	9
29	opr	0	0
30	jmp	0	31
31	ini	0	4
32	lod	1	3
33	sto	1	7
34	lit	0	0
35	sto	1	6
36	lod	1	4
37	sto	0	3
38	lod	0	3

```

        while f#g do
            begin
                if f<g then
                    g:=g-f;
                if g<f then
                    f:=f-g;
                end;
            z:=f
        end;

begin
    x:=m;y:=n;call multiply;write(z);
    x:=25;y:=3;call divide;write(q);
    x:=84;y:=36;call gcd;write(z)
end.

```

39	lod	1	7
40	opr	0	13
41	jpc	0	47
42	lit	0	2
43	lod	0	3
44	opr	0	4
45	sto	0	3
46	jmp	0	38
47	lod	0	3
48	lod	1	4
49	opr	0	12
50	jpc	0	72
51	lit	0	2
52	lod	1	6
53	opr	0	4
54	sto	1	6
55	lod	0	3
56	lit	0	2
57	opr	0	5
58	sto	0	3
59	lod	0	3
60	lod	1	7
61	opr	0	13
62	jpc	0	71
63	lod	1	7
64	lod	0	3
65	opr	0	3
66	sto	1	7
67	lod	1	6
68	lit	0	1
69	opr	0	2
70	sto	1	6
71	jmp	0	47
72	opr	0	0
73	jmp	0	74
74	ini	0	5
75	lod	1	3
76	sto	0	3
77	lod	1	4
78	sto	0	4
79	lod	0	3
80	lod	0	4

81	opr	0	9
82	jpc	0	100
83	lod	0	3
84	lod	0	4
85	opr	0	10
86	jpc	0	91
87	lod	0	4
88	lod	0	3
89	opr	0	3
90	sto	0	4
91	lod	0	4
92	lod	0	3
93	opr	0	10
94	jpc	0	99
95	lod	0	3
96	lod	0	4
97	opr	0	3
98	sto	0	3
99	jmp	0	79
100	lod	0	3
101	sto	1	5
102	opr	0	0
103	ini	0	8
104	lit	0	7
105	sto	0	3
106	lit	0	85
107	sto	0	4
108	cal	0	2
109	lod	0	5
110	opr	0	14
111	opr	0	15
112	lit	0	25
113	sto	0	3
114	lit	0	3
115	sto	0	4
116	cal	0	31
117	lod	0	6
118	opr	0	14
119	opr	0	15
120	lit	0	84
121	sto	0	3
122	lit	0	36

123	sto	0	4
124	cal	0	74
125	lod	0	5
126	opr	0	14
127	opr	0	15
128	opr	0	0

符号表如下：

name	kind	val / level	adr	size
m	const	7		
n	const	85		
x	var	0	3	
y	var	0	4	
z	var	0	5	
q	var	0	6	
r	var	0	7	
multiply	pro	0	2	5
a	var	1	3	
b	var	1	4	
divide	pro	0	31	4
w	var	1	3	
gcd	pro	0	74	5
f	var	1	3	
g	var	1	4	