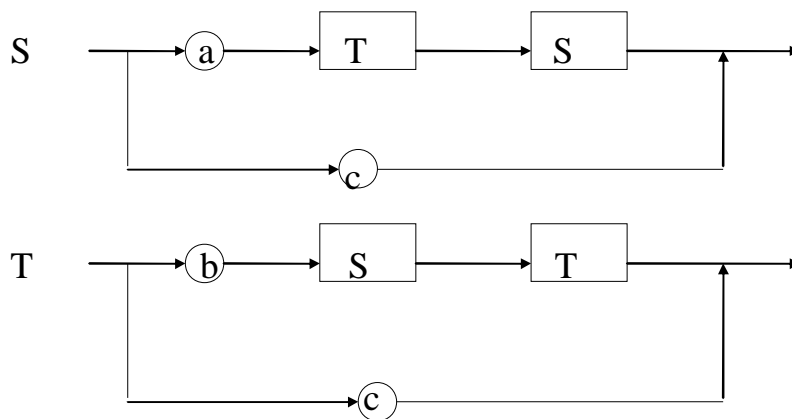


Problem 1. A syntax chart is given as follows:



(1) Write a recursive descent parser for this chart in C, Pascal, or Java. Pseudo code notation is allowed, similar to the notes.

(2) The trace for the string acc! where ! is an end marker is given as follows:

History	Input string	Comment
S	acc!	Consume a
S	cc!	Enter T
ST	cc!	Consume c
ST	c!	Exit from T
S	c!	Enter S
SS	c!	Consume c
SS	!	Exit from S
S	!	Exit from S
empty	!	Accept

Following this example, trace your parser with the string abccacc!

History	Input string	Comment
S	abccacc!	Consume a
S	bccacc!	Enter T
ST	bccacc!	Consume b
ST	ccacc!	Enter S
STS	ccacc!	Consume c
STS	cacc!	Exit from S
ST	cacc!	Enter T
STT	cacc!	Consume c
STT	acc!	Exit from T
ST	acc!	Exit from T
S	acc!	Enter S
SS	acc!	Consume a
SS	cc!	Enter T
SST	cc!	Consume c
SST	c!	Exit from T
SS	c!	Enter S
SSS	c!	Consume c
SSS	!	Exit from S
SS	!	Exit from S
S	!	Exit from S
Empty	!	Accept

Problem 2. The following PL/0 program divides a by b, and store the quotient in q and the remainder in r. The symbol  $\bullet \% \in$  is for  $\bullet \geq \bullet$ .

```

const a=25, b=7;
var q,r;
procedure P;
begin
  if r%b then begin
    r:=r-b; q:=q+1; call P;
  end;
end;
begin
  r:=a; q:=0; call P;
  write(q);
  write(r);
end.

```

The corresponding object code is given as follows:

0	jmp 0 17	jump to 17 (starting point of main)
1	jmp 0 2	jump to 2 (starting point of procedure P)
2	inc 0 3	increment stack (pointer) register by 3
3	lod 1 4	load r
4	lit 0 7	load literal 7 (=b)
5	opr 0 13	test $\bullet \geq \bullet$
6	jpc 0 16	jump to 16 if false
7	lod 1 4	load r
8	lit 0 7	load literal 7
9	opr 0 3	r-b
10	sto 1 4	store in r
11	lod 1 3	load q
12	lit 0 1	load literal 1
13	opr 0 2	q+1
14	sto 1 3	store in q
15	cal 1 2	call procedure P
16	opr 0 0	return to the address immediately after the calling point
17	inc 0 5	increment the stack register by 5 (3 for overhead plus 2 for q and r)
18	lit 0 25	load literal 25
19	sto 0 4	store in r
20	lit 0 0	load literal 0
21	sto 0 3	store in q
22	cal 0 2	call procedure P
23	lod 0 3	load q
24	wrt 0 0	write(q)
25	lod 0 4	load r
26	wrt 0 0	write•
27	opr 0 0	return to operating system

(1) Copy this code onto the answer sheet and give a comment to each instruction describing the meaning in the source program.

(2) After procedure P is entered twice and the machine instruction at address 12 is executed the stack becomes as follows:

0 0 0 1 11 1 1 23 1 6 16 1 1

Explain the meaning of each element.

0	0	0	1	11	1	1	23	1	6	16	1	1
SL	DL	RA	q	r	SL	DL	RA	SL	DL	RA	v(q)	lit 1

SL : static link, DL : dynamic link, RA : return address

v(q) is the value of q

Lit 1 means 1 is loaded by load literal operation

(3) Show the contents of the stack after you execute instruction 12 for the third time.

0 0 0 2 4 1 1 23 1 6 16 1 9 16 2 1