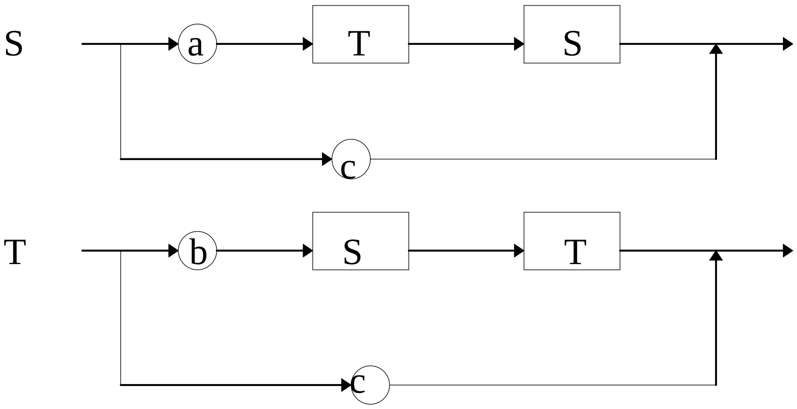
Problem 1. A syntax chart is given as follows:



(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!

Problem 2. The following PL/0 program divides a by b, and store the qoutient in q and the remainder in r. The symbol “%” is for “>=“.

```
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
 1 jmp 0 2
 2 inc 0 3
 3 lod 1 4
 4 lit 0 7
 5 opr 0 13
 6 jpc 0 16
 7 lod 1 4
 8 lit 0 7
 9 opr 0 3
10 sto 1 4
11 lod 1 3
12 lit 0 1
13 opr 0 2
14 sto 1 3
15 cal 1 2
16 opr 0 0
17 inc 0 5
18 lit 0 25
19 sto 0 4
20 lit 0 0
21 sto 0 3
22 cal 0 2
23 lod 0 3
24 wrt 0 0
25 lod 0 4
26 wrt 0 0
27 opr 0 0
```

(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.

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