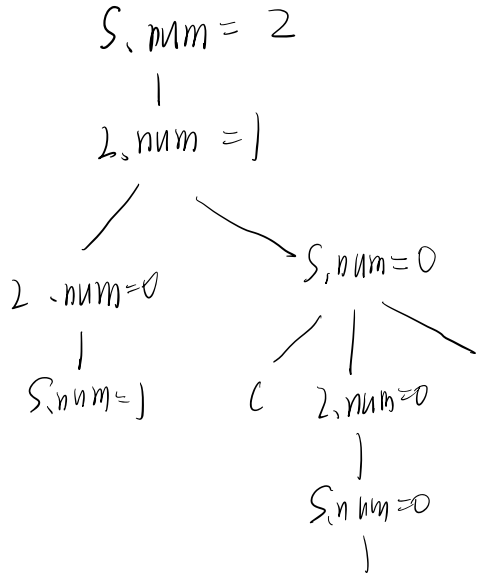
1. $S' \rightarrow S$ {print(S.val)}

$S \rightarrow S \vee T$ {S.val = S.val or T.val} | {S.val = T.val}

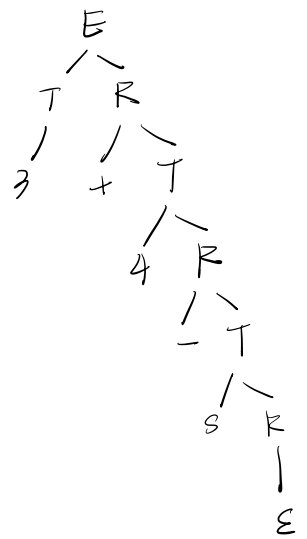
$T \rightarrow T \wedge F$ {T.val = T.val and F.val} | F {T.val = F.val}

$F \rightarrow \neg F$ {F.val = not F[].val} | false {F.val = False} | true {F.val = True}

2.



3.



E.val = 无

T.val = 3 R.in = 3

T.val = 4 R.in = 空

T.val = 5 R_2.in = R_2.val = 空.

状态栈	运算栈	符号栈	输入串栈
14:	02	--	#C	)#
15:	0	-	#	#

b、题目的文法G[E]如下：

E → TR

R → +T R1 | -T R1 | ε

T → num

(1)、计算FIRST 和Follow.

FIRST(E) = FIRST(T) = {num}

FIRST(R) = {+, -, ε}

Follow(T) = {num}

Follow(*E) = {#}

Follow(R) = Follow(E) = {#}

Follow(T) = FIRST(R) - {ε} = {+, -}

对所有 A → α | β，有 FIRST(α) ∩ FIRST(β) = ∅

对 ∀a ∈ FIRST(α) 和 b ∈ Follow(A)，有 a ≠ b.

则 G[E]是 LL(1)文法.

(2). 翻译程序如下：

```
token = None
#获取  def  getToken():
            global token

#匹配终结符
def match(t):
    global token
    if token == t:
        getToken()
    else:
        #报错.

#处理非终结符：
def E():
    global token
    t.val = T()
    r.in = t.val
    e.val = R(r.in)
    return e.val.

#处理非终结符R
def R(r.in):
    global token
    if token == '+'
        match('+')
        t.val = T()
        r1.in = r.in + t.val
        r.val = R(r1.in)
    elif token == '-':
        match('-')
        t.val = T()
        r1.in = r.in - t.val
        r.val = R(r1.in)
    else: r.val = r.in.
    return. r.val

#处理非终结符T
def T():
    global token
    if token == 'num':
        t.val = (eval)(token)
        match('num')
    else = #报错
    return t.val
def main():
        global token
        next Token()
        result = E()
        print(result)

main()
```

第8章.

1. 1. $E_1.val = E_2.val$ 仔 $E_3.val$
   2. $E_1.true = newlabel()$
   3. $E_1.false = E_3.fake$
   4. 生成：$E_2.true = newlabel().goto\ E_1.true$ 仔 $E_2.val == false$
   5. 生成：$goto\ E_3.true$ 仔 $E_3.val == false$
   6. 生成：$E_1.val = true$
   7. 生成：$label(E_1.true)$
   8. 生成：$E_1.val = false$

2. 1. $S.next = newlabel()$
   2. 生成：$label(S.next)$
   3. $E.true = newlabel()$
   4. $E.false = S.next$
   5. $S_1.next = E.gnad$
   6. 生成：$goto.E.true$

3. ①对于 $E_1 \to E_1$ or E_2
   1. $E_1.true = E.true$
   2. $E_1.false = newlabel()$
   3. $E_2.true = E.true.$
   4. $E_2.false = E.false.$
   5. 生成：$label(E_1.false)$

   ②对于 $S \to$ 仔 E then S_1 else S_2
   1. $E.true = newlabel()$
   2. $E.false = newlabel()$
   3. $S_1.next = S.next$
   4. $S_2.next = S.next$
   5. 生成：$label(E.true)$
   6. 生成：$goto\ S.next$
   7. 生成：$label(E.false)$