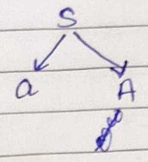
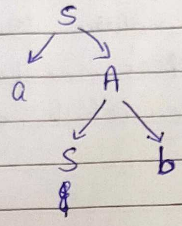
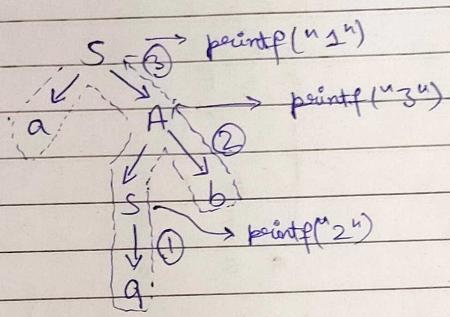
Date

Compiler Design

Q-1 (a)







Output for given input aab will be "231".

Stack	Input	Action
\$	aab\$	shift a
\$a	ab\$	shift a
\$aa	b\$	reduce $S \to a$ // print "2"
\$aS	b\$	shift b
\$aSb	\$	reduce $A \to Sb$ // print "3"
\$aA	\$	reduce $S \to aA$ // print "1"
\$S	\$	accept

Spiral

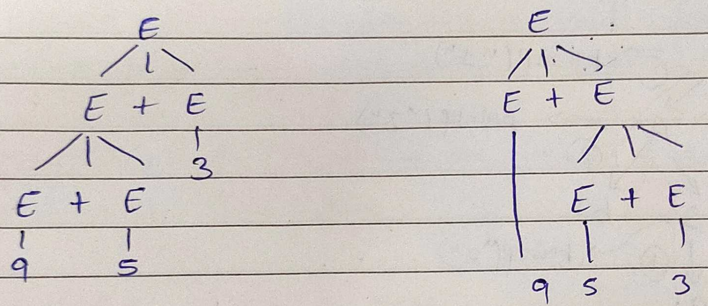
Date________

b. $E \to E+E$
$E \to E*E$
$E \to (E)$
$E \to id$

~~Ex: (9+5+3~~

→ Resolve conflicts using associativity : we know that '+' and '*' operators are left associative in nature. So, in case when there can be multiple parse ~~trees~~ trees are possible for a single expression, we choose the tree which grows towards the left direction.

For ex: 9+5+3



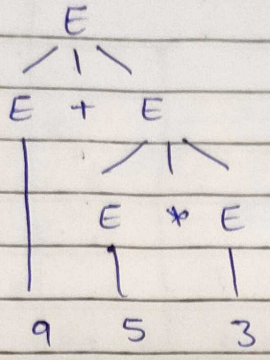
(a) (b)

Tree (a) is the correct representation for expression "9+5+3" as it grows towards the left direction.

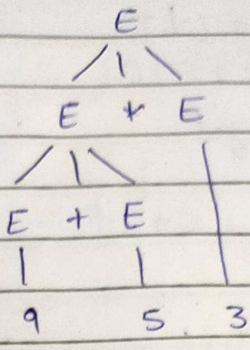
→ Resolve conflicts using precedence : We can apply rule of associativity to occurence of same operator. So, for multiple operators (e.g + and *) we can use relative precedence of operator.

For ex : 9+5*3

Date



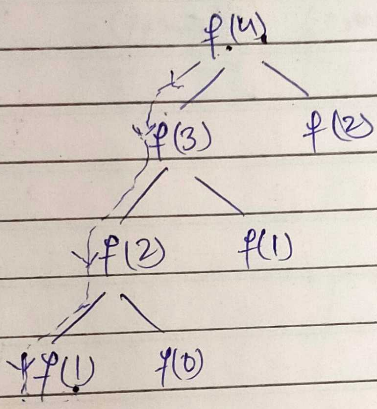
(a)



(b)

Here, tree (a) is correct as precedence of * is higher than the precedence of +, so, it takes it operands before + does.

(c)



Run-time stack

integer n
f(4)
int s, t | s = f(3) | t = f(2)

integer n
f(3)
int t, s | s = f(2) | t = f(1)

integer n
f(2)
int t, s | s = f(1) | t = f(0)

integer n
f(1)
return 1

Spiral

Date

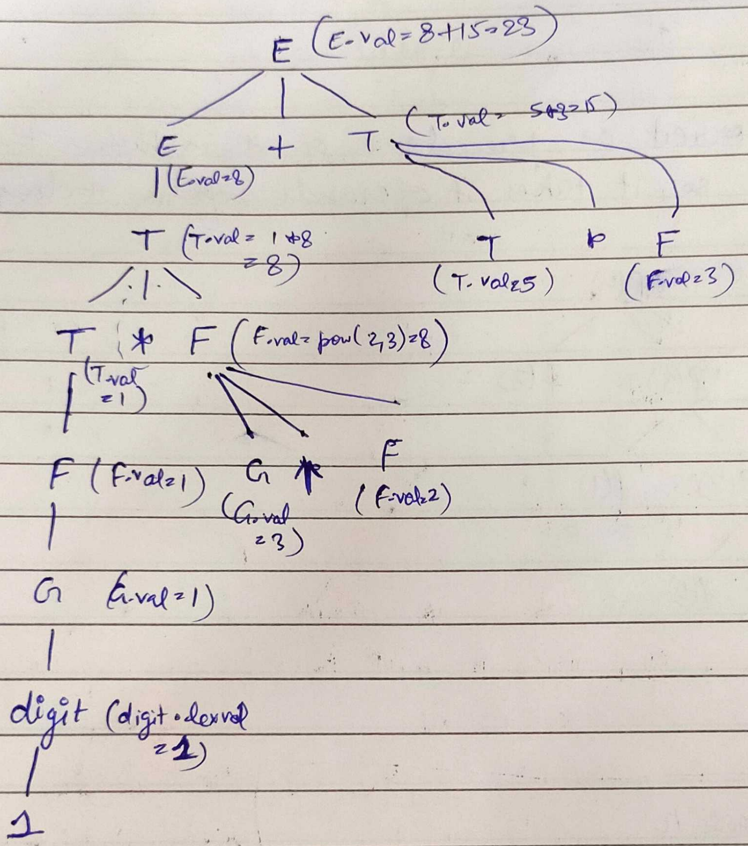
Q-2 $T \to T * F \mid F$

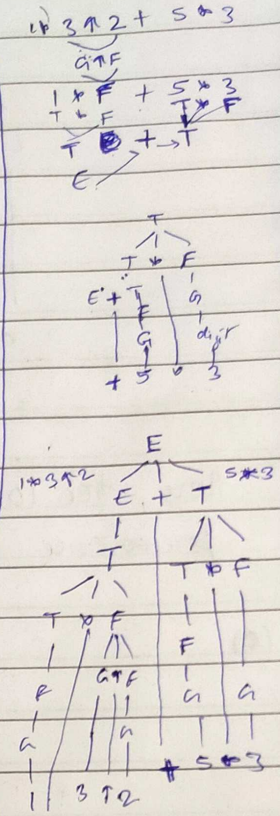
$E \to E + T \mid T$

$F \to G \uparrow F \mid G$

$G \to \text{digit}$

Expr: $1 * 3 \uparrow 2 + 5 * 3$





(b) $S \to Aa \mid bAc \mid Bc \mid bBa$

$A \to d$

$B \to d$

Augmented Grammar:

$S' \to .S,\ \$$

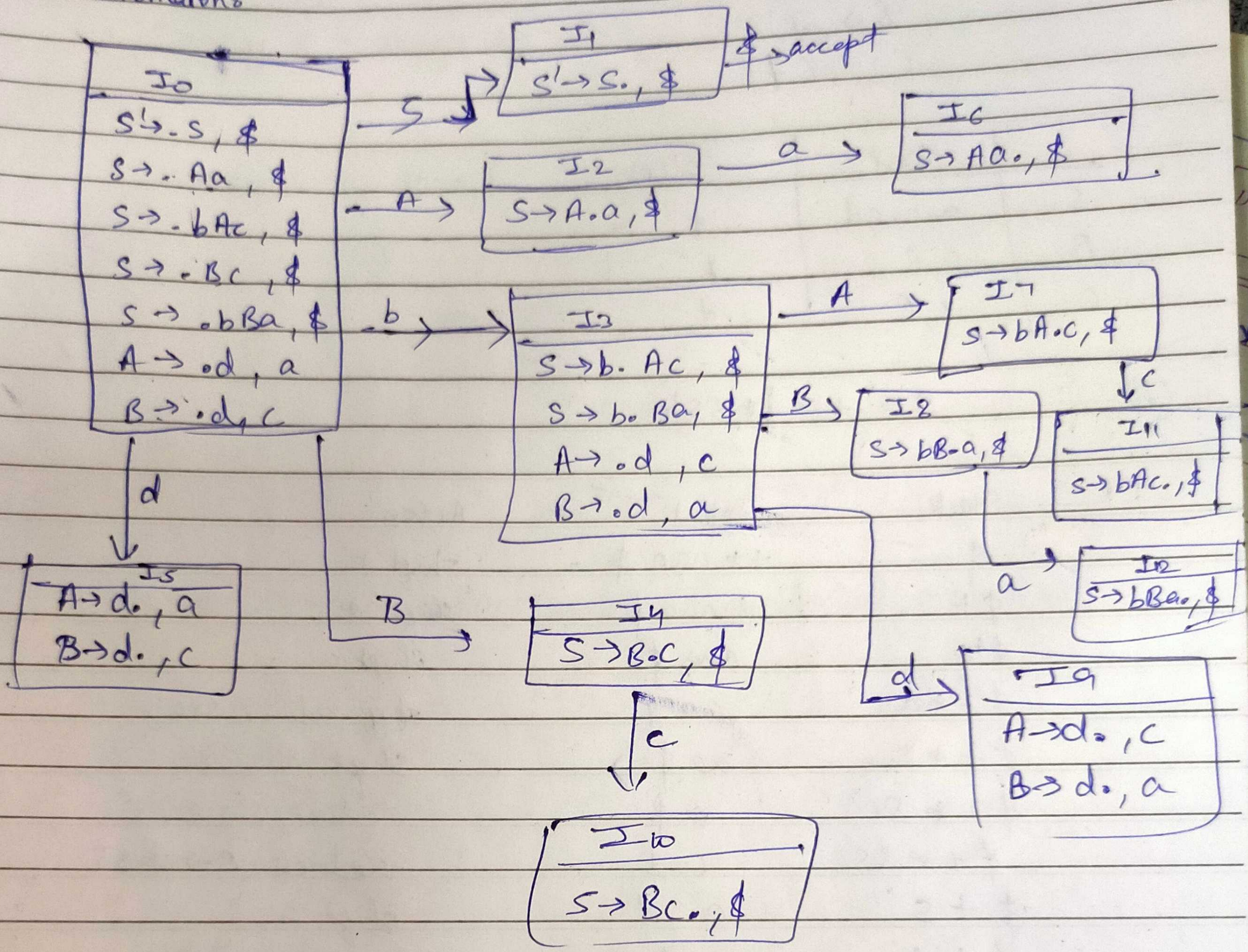
$S \to .Aa^{(1)} \mid .bAc^{(2)} \mid .Bc^{(3)} \mid .bBa^{(4)},\ \$$

$A \to .d^{(5)},\ a$

$B \to .d^{(6)},\ c$

Date

Automaton:



Items/state	Action					GOTO		
	a	b	c	d	$	S	A	B
0		s3		s5		1	2	4
1					ac			
2	s6							
3				s9			7	8
4			s10					
5	r5		r6					
6					r1			
7			s11					
8	s12							
9	r6		r5		r3			
10					r2			
11					r4			
12								

Date

Ans (a) $S \to Bb \mid Cd$
$B \to aB \mid \epsilon$
$C \to cC \mid \epsilon$

	First	Follow
S	a, b, c, d	$
B	a, ε	b
C	c, ε	d

(b) $S \to +SS \mid *SS \mid a$

Stack	Input	Action
$	+*aaa$	shift '+'
$+	*aaa$	shift *
$+*	aaa$	shift a
$+*a	aa$	~~shift~~ reduce S→a
$+*S	aa$	shift a
$+*Sa	a$	reduce S→a
$+*SS	a$	reduce S→*SS
$+S	a$	shift a
$+Sa	$	reduce S→a
$+SS	$	reduce S→+SS
$S	$	accept

Viable prefix for given input are +, +*, +*a etc.

(c) a *= b;
<id, pointer to symbol table entry for a>
~~<id, 1><*><=><id, 2><semicolon>~~
<mult_op>
<assign_op>
<id, pointer to symbol table entry for b>
<semicolon>

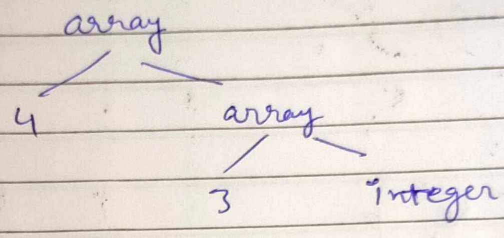
Spiral

Date____

Q-4(a)

"array of 4 arrays of 3 integers each"

array (4, array(3, integer))



(b) Error-recovery actions performed by lexical analyzers:

Panic-mode
Phrase-level
Error productions
global correction

Q-5 (a)

```
%{
    #include <stdio.h>
    int pos=0, neg=0, frac=0;
%}

Digit     [0-9]
Digits    Digit (Digit)*
%%

"+"? Digits "." Digits      { frac++; pos++; }
"+"? Digits                 { pos++; }
"-" Digits "." Digits       { frac++; neg++; }
"-" Digits                  { neg++; }
```

Spiral

Date

```
%%
int main()
{
    yylex();
    printf("

    return0;
}
```

Q6 (c)

```
LD   R0  p
LD   R1  q
SUB  R0  R0  R1
LD   R1  r
LD   R2  s
MUL  R1  R1  R2
DIV  R0  R0  R1
LD   R1  t
LD   R2  u
MUL  R1  R1  R2
ADD  R0  R0  R1
ST   a   R0
```

Spiral