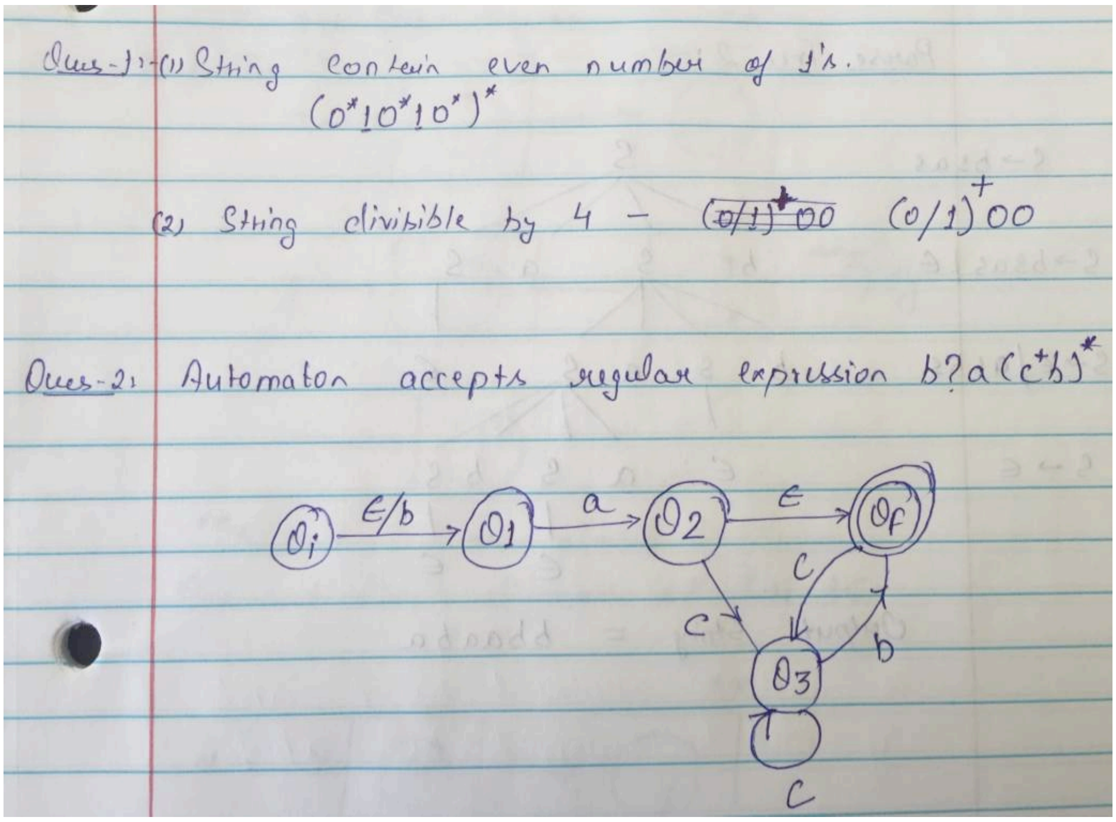
Shashi Upadhyay
BU ID: B00627613
Section: CS 571-01
Email: supadhy2@binghamton.edu

Hardik Bagdi
BU ID: B00576043
Section: CS 571-02
Email: hbagdi1@binghamton.edu

CS571, Programming Languages – Assignment 2

Answer 1 and 2:



Ques-1:- (1) String contain even number of 1's.

$(0^*10^*10^*)^*$

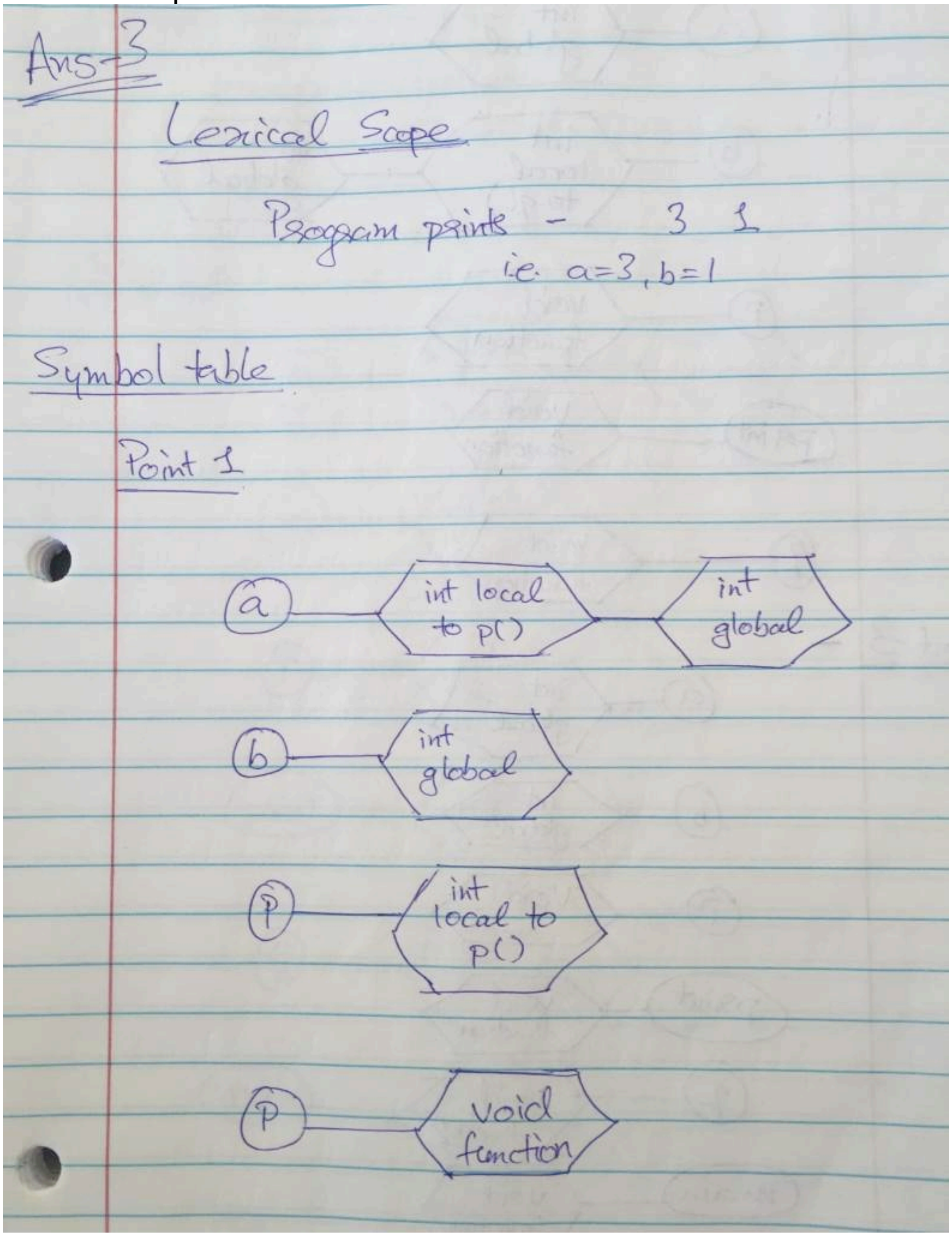
(2) String divisible by 4 - ~~$(0/1)^*00$~~ $(0/1)^+00$

Ques-2: Automaton accepts regular expression $b?a(c^+b)^*$

$$\begin{tikzcd}
Q_i \arrow[r, "\epsilon/b"] & Q_1 \arrow[r, "a"] & Q_2 \arrow[r, "\epsilon"] \arrow[dr, "c"'] & Q_f \arrow[d, bend right, "c"'] \\
 & & & Q_3 \arrow[u, bend right, "b"'] \arrow[loop below, "c"]
\end{tikzcd}$$

Question 3:

Lexical Scope:



Ans-3

Lexical Scope

Program prints - 3 1

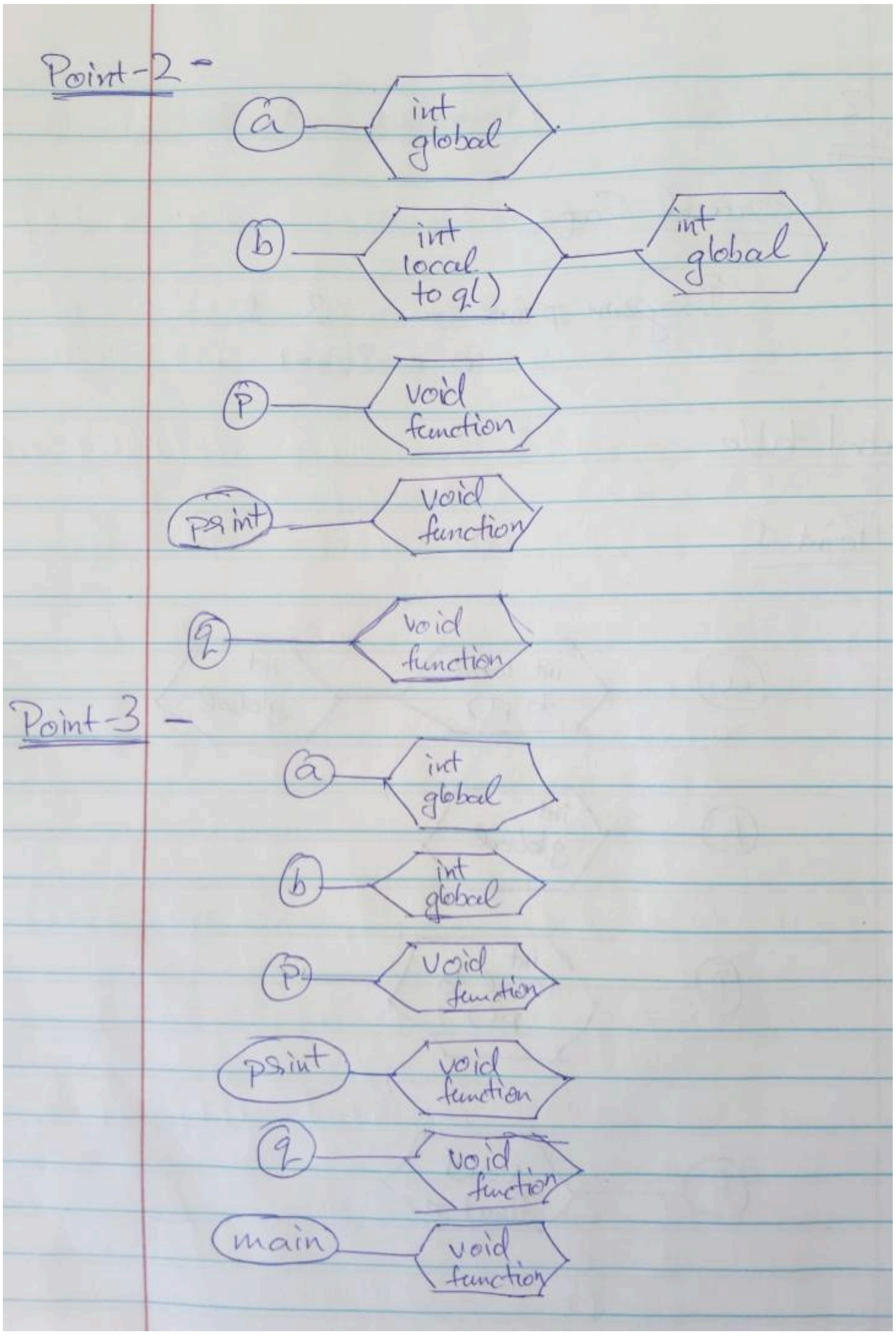
i.e. a=3, b=1

Symbol table

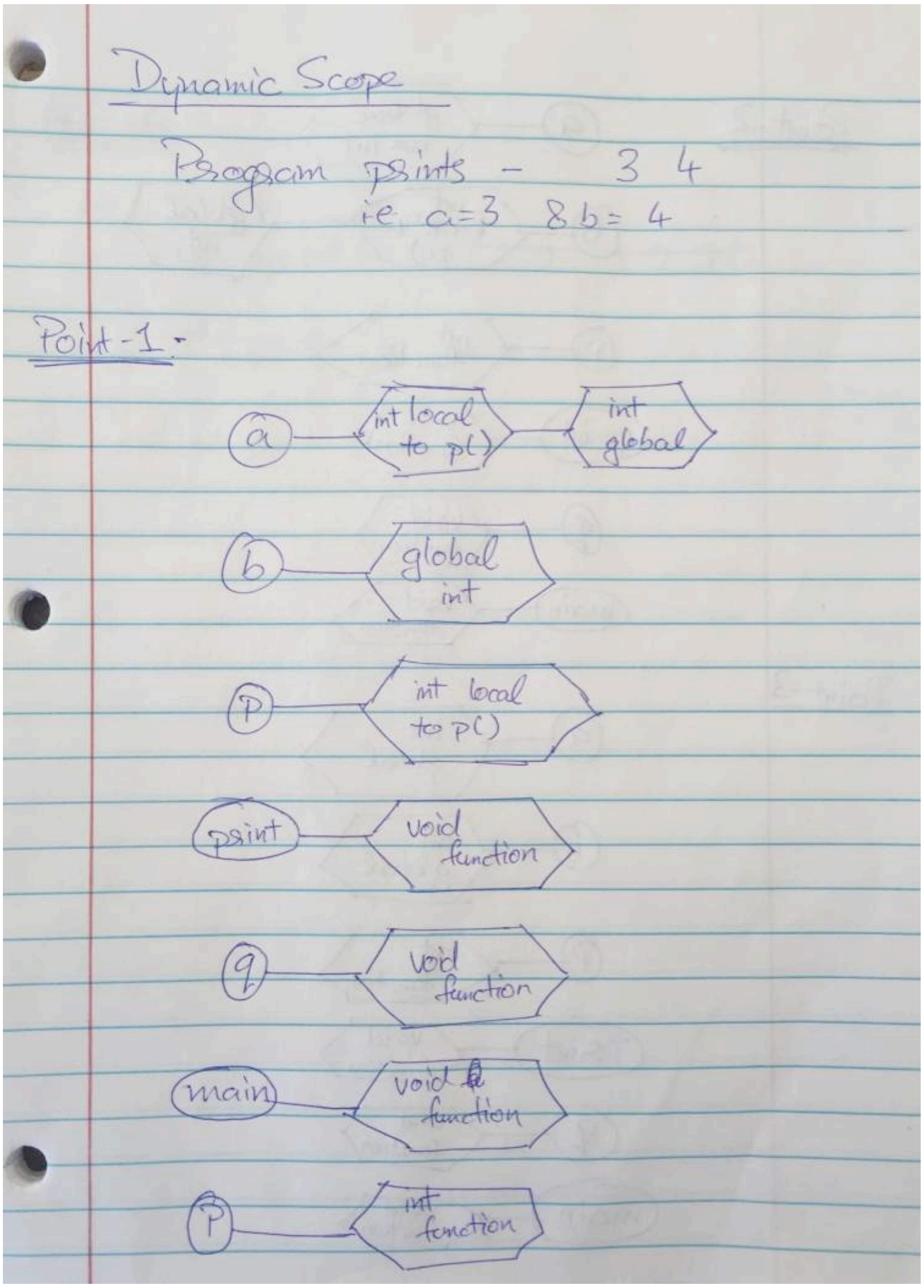
Point 1

$$\begin{tikzcd}
a \arrow[r, dash] & \text{int local to p()} \arrow[r, dash] & \text{int global} \\
b \arrow[r, dash] & \text{int global} & \\
p \arrow[r, dash] & \text{int local to p()} & \\
p \arrow[r, dash] & \text{void function} &
\end{tikzcd}$$

Lexical Scope Continues:



Dynamic Scope:

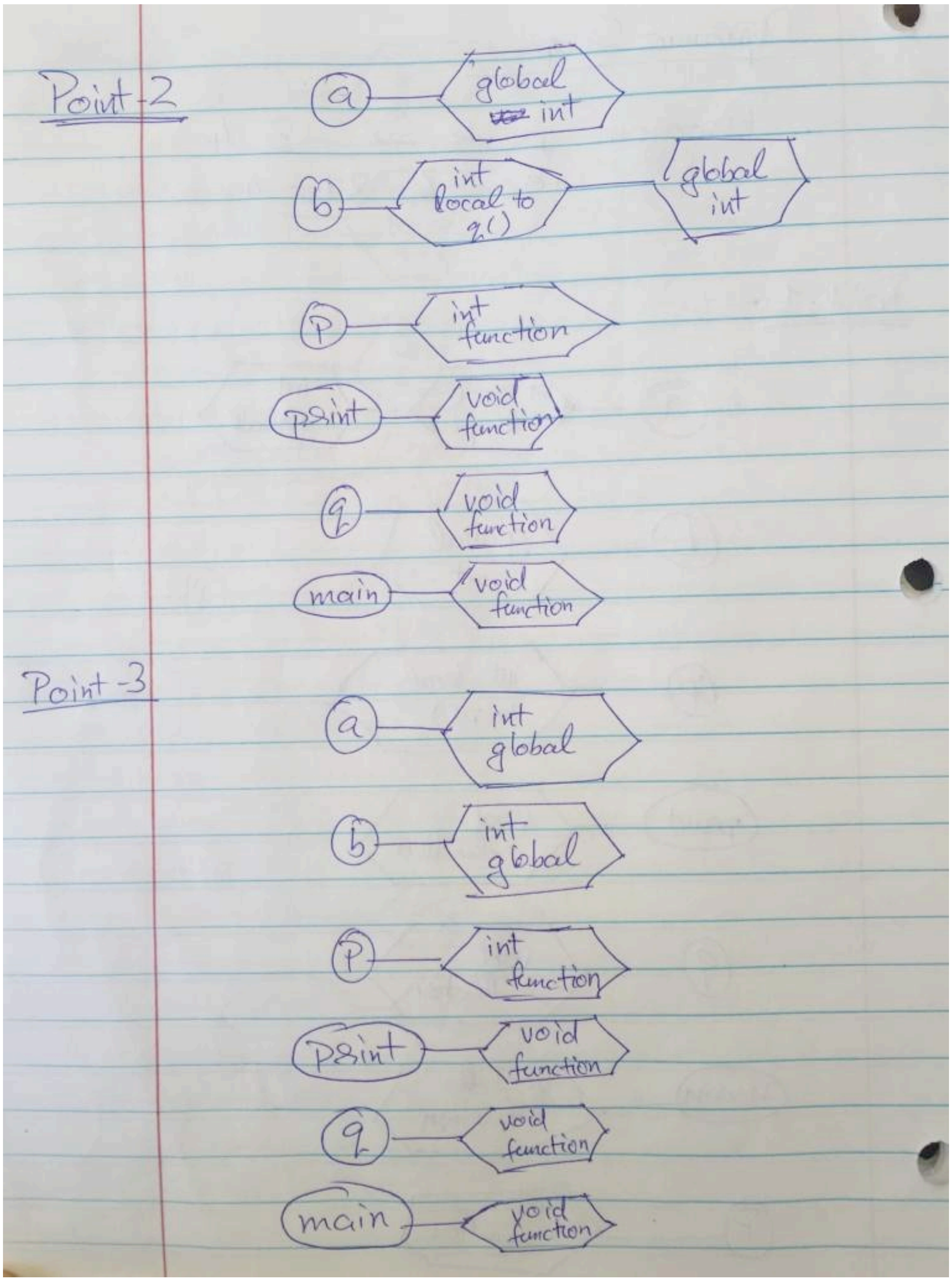


Dynamic Scope

Program prints - 3 4
ie a=3 & b= 4

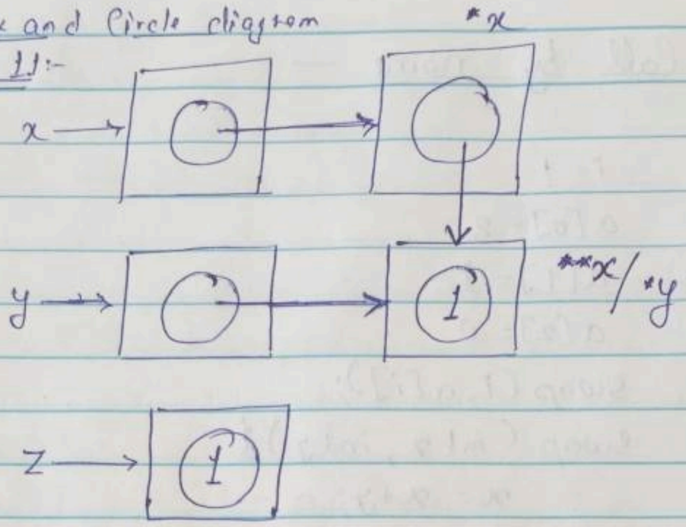
Point-1 -

Dynamic Scope Continues:



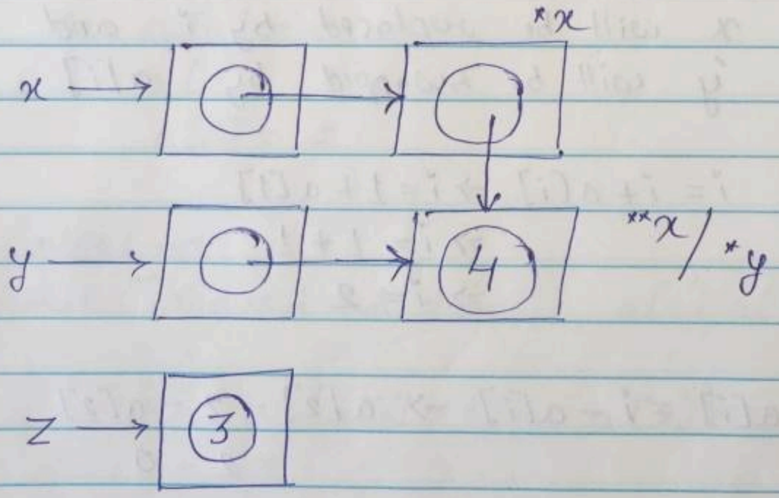
Question 4:

Ques 4 :- Box and Circle diagram
Line 11:-



**x and *y are aliases.

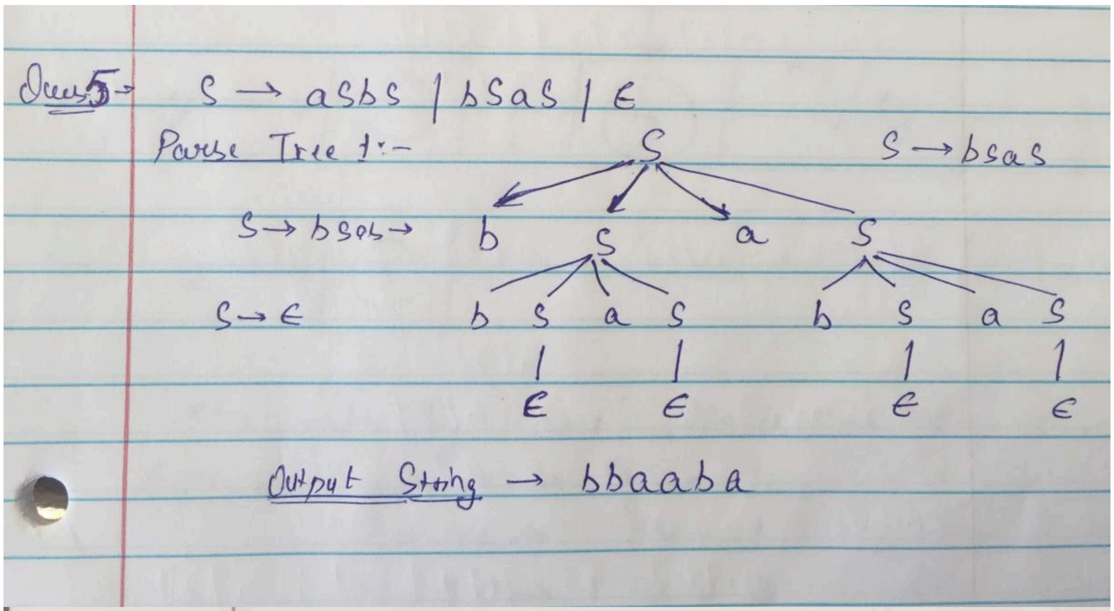
Box and Circle diagram at line 15:



**x and *y are aliases at line 15

Output → Line 12 → *y = 1
Line 14 → *y = 1
Line 16 → z = 3

Question 5:

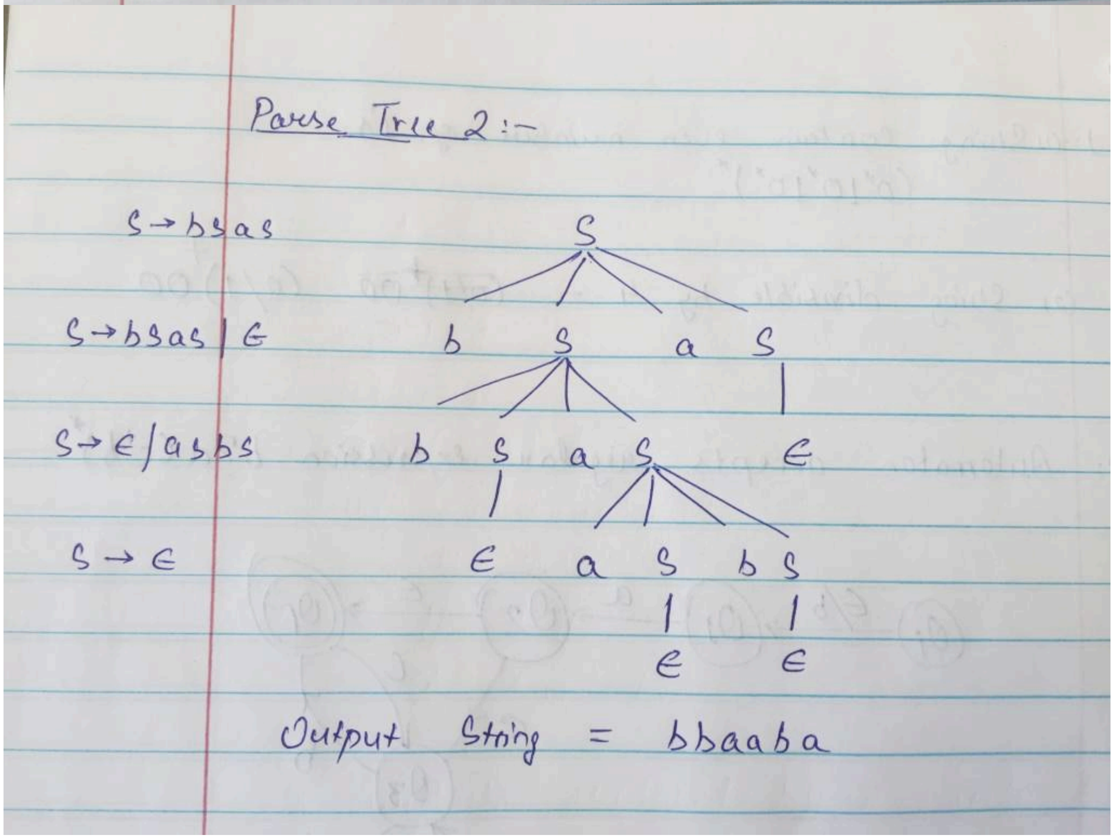


Ques5 → $S \rightarrow aSbS \mid bSaS \mid \epsilon$

Parse Tree 1:-

$S \rightarrow bSaS$
$S \rightarrow bSaS \rightarrow$
$S \rightarrow \epsilon$
S
b S a S
b S a S b S a S
ϵ ϵ ϵ ϵ

Output String → bbaaba



Parse Tree 2:-

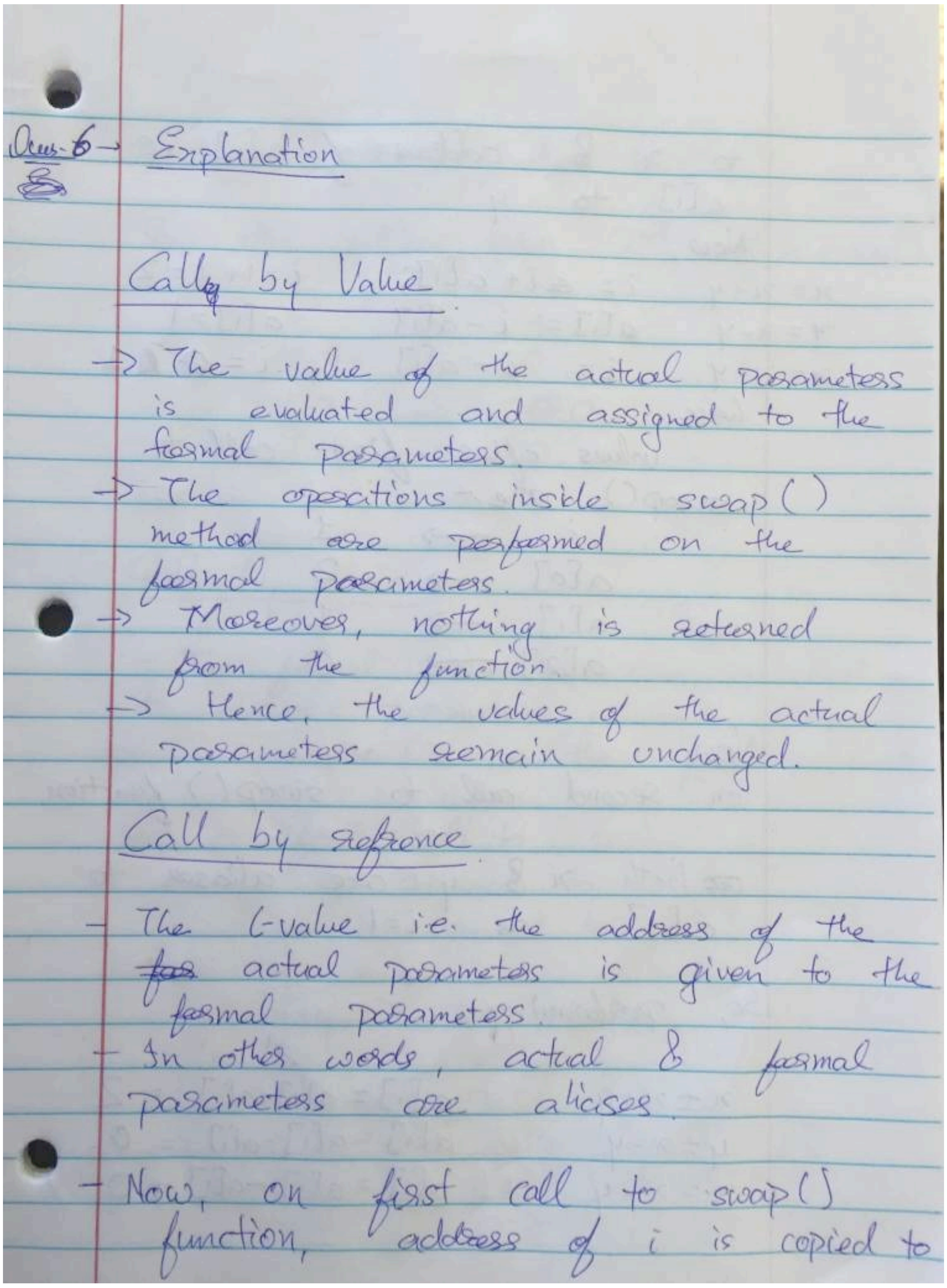
Output String = bbaaba

Question 6:

First Print statement Output				
	i	a[0]	a[1]	**a[2]**
Call by Value	1	2	1	0
Call by Reference	1	2	1	0
Call By name	0	2	1	2

Second Print statement Output			
	a[0]	a[1]	a[2]
Call by Value	2	1	0
Call by Reference	2	0	0
Call By name	0	1	2

Question 6 Solution Continues:



Ques-6 → Explanation

Call by Value

→ The value of the actual parameters is evaluated and assigned to the formal parameters.

→ The operations inside swap( ) method are performed on the formal parameters.

→ Moreover, nothing is returned from the function.

→ Hence, the values of the actual parameters remain unchanged.

Call by refrence

- The L-value i.e. the address of the ~~for~~ actual parameters is given to the formal parameters.
- In other words, actual & formal parameters are aliases.

- Now, on first call to swap( ) function, address of i is copied to

to x & address of a[i] i.e.
a[i] to y.

Now,

$x = x + y$	$i = \cancel{a}\, i + a[i]$;	$i = 1 + 1 = 2$
$y = x - y$	$a[i] = i - a[i]$	$a[i] = 1$
$x = x - y$	$i = i - a[i]$	$i = \cancel{0}\,\cancel{2}\,1$

hence,
values after first call to swap() are -

$i \longrightarrow 1$
$a[0] \longrightarrow 2$
$a[1] \longrightarrow 1$
$a[2] \longrightarrow 0$

Now,
on second call to swap() function,

~~as~~ both x & y are aliases to a[1] as $i=1$.

So, performing,

$x \pm x + y$	$a[1] = \overset{2'}{a[1]} + \overset{x'}{a[1]} = 2$
$y = x - y$	$a[1] = a[1] - a[1] = 0$
$x = x - y$	$a[1] = a[1] - a[1] = 0$

So, a[i] is evaluated to 0.

So, the values after 2nd call to swap()

i	$\longrightarrow$	1
a[0]	$\longrightarrow$	2
a[1]	$\longrightarrow$	0
a[2]	$\longrightarrow$	0

<u>Call by name</u>

- The formal parameters are evaluated to actual parameters whenever they are encountered in the function body

So, on first call to swap(),
x is renamed to i
& y is renamd to a[i]

Now,

$x = x + y \Rightarrow i = i + a[i]$

$i = 1 + a[i] = 2$

$y = x - y \Rightarrow a[2] = i - a[2]$

$= 2 - 0 = 2$

$x = x - y \Rightarrow i = i - a[2]$

$= 2 - 2 = 0$

Hence, values after the first call -

$i \rightarrow 0$
$a[0] \rightarrow 2$
$a[1] \rightarrow 1$
$a[2] \rightarrow 2$

Now, on second call to swap()
x & y are renamed to a[0] (as $i=0$)

Now,

$a[0] = a[0] + a[0]$
$= 2+2 = 4$
$a[0] = a[0] - a[0]$
$= 4 - 4 = 0$
$a[0] = a[0] - a[0]$
$= 0$

Hence, values after the second call to swap()

$i \rightarrow 0$
$a[0] \rightarrow 0$
$a[1] \rightarrow 1$
$a[2] \rightarrow 2$