Faculty of Computing, Online Examinations 2021

STUDENT NAME	**LNP Ariyarathna**		
INDEX NUMBER (NSBM)	**21386**	**YEAR OF STUDY AND SEMESTER**	**Year 1 semester 2**
MODULE NAME **(As per the paper)**	**Algorithms and Data structures**		
MODULE CODE	**CS106.3**		
MODULE LECTURER	**Mrs Manoja weerasekara**	**DATE SUBMITTED**	**2021/10/13**

For office purpose only:

GRADE/MARK	
COMMENTS	

Declaration

PLEASE TICK TO INDICATE THAT YOU HAVE SATISFIED THESE REQUIREMENTS

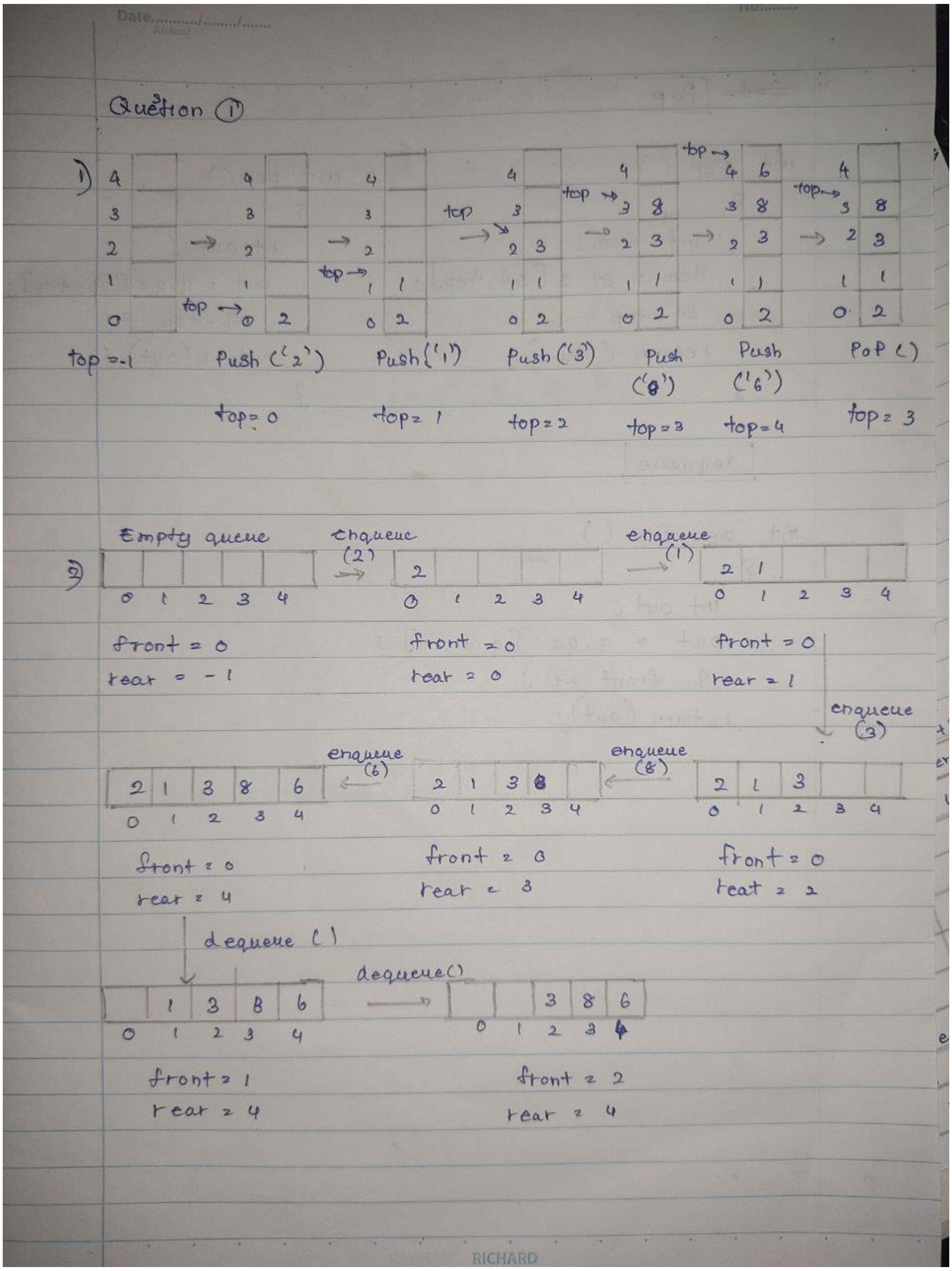
- ✓ I have carefully read the instructions provided by the Faculty
- ✓ I understand what plagiarism is and I am aware of the University's policy in this regard.
- ✓ I declare that the work hereby submitted is my own original work. Other people's work has been used (either from a printed source, Internet or any other source), has been properly acknowledged and referenced in accordance with the NSBM's requirements.
- ✓ I have not used work previously produced by another student(s) or any other person to hand in as my own.
- ✓ I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.
- ✓ I hereby certify that the individual detail information given (name, index number and module details) in the cover page are thoroughly checked and are true and accurate.

I hereby certify that the statements I have attested to above have been made in good faith and are true and correct. I also certify that this is my own work and I have not plagiarized the work of others and not participated in collusion.

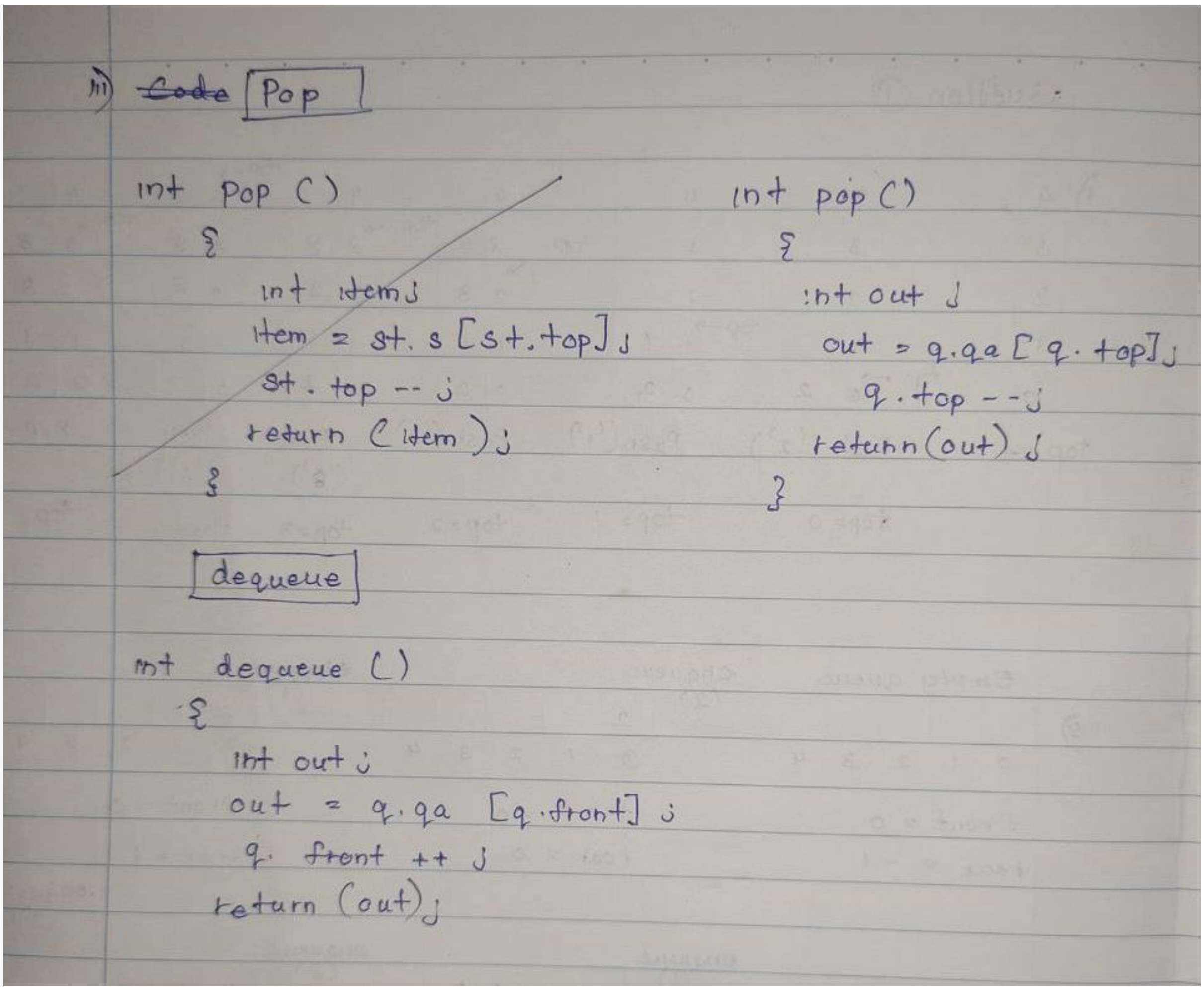
Date: ……2021/10/13……… … **E- Signature: [illegible]

**Please attach a photo/image of your signature in the space provided

question 1



Question ①
1)
top =-1
Push ('2') top= 0
Push('1') top= 1
Push('3') top= 2
Push ('8') top= 3
Push ('6') top= 4
PoP () top= 3
2)
Empty queue
front = 0
rear = -1
enqueue (2)
front = 0
rear = 0
enqueue (1)
front = 0
rear = 1
enqueue (3)
front = 0
rear = 2
enqueue (8)
front = 0
rear = 3
enqueue (6)
front = 0
rear = 4
dequeue ()
front = 1
rear = 4
dequeue()
front = 2
rear = 4



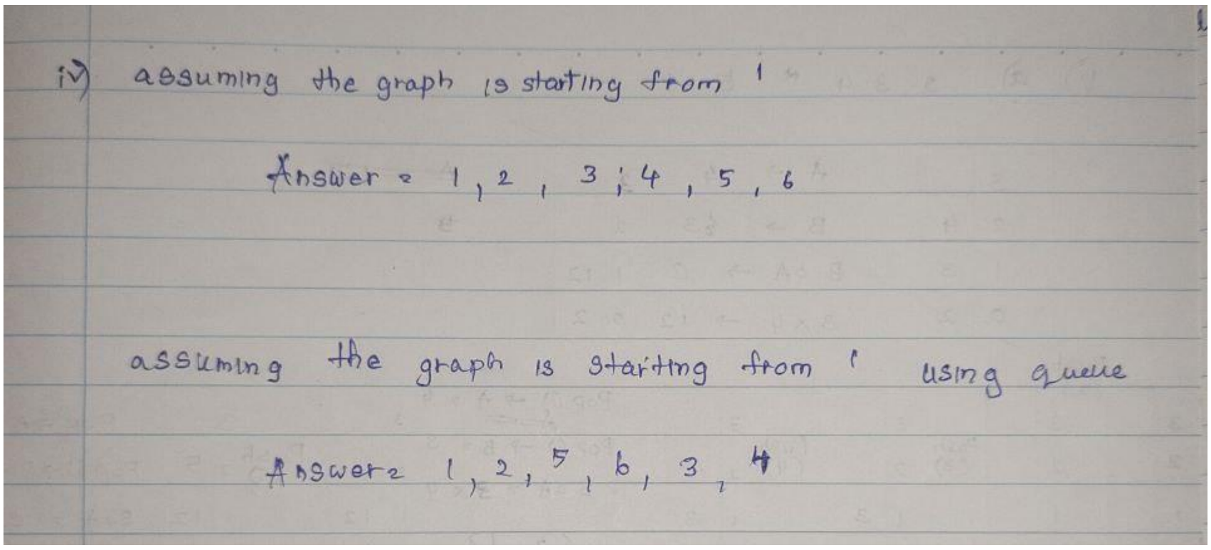
iii) ~~Code~~ Pop

```
~~int Pop ( )~~
~~    {~~
~~        int item;~~
~~        item = st. s [st.top];~~
~~        st. top -- ;~~
~~        return (item);~~
~~    }~~
```

```
int pop ( )
  {
    int out ;
     out = q.qa [ q. top];
       q.top --;
    return(out);
  }
```

dequeue

```
int dequeue ( )
  {
     int out ;
     out  =  q.qa  [q.front] ;
      q. front ++ ;
    return (out);
```

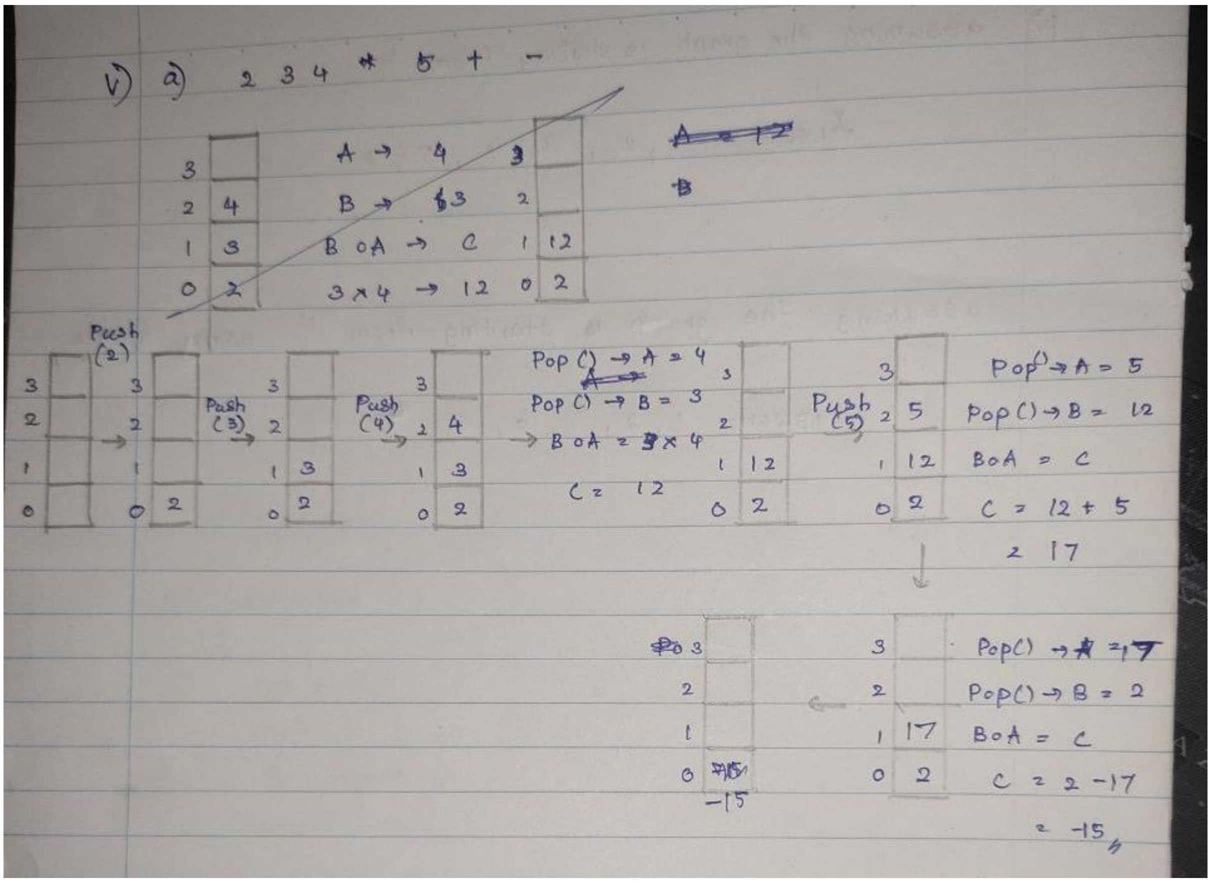


iv) assuming the graph is starting from 1

Answer = 1, 2, 3, 4, 5, 6

assuming the graph is starting from 1 using queue

Answer = 1, 2, 5, 6, 3, 4



v) a) 2 3 4 * 5 + -

~~A → 4~~
~~B → 3~~
~~B o A → C~~
~~3 × 4 → 12~~

~~A = 17~~

~~B~~

Push (2)

Push (3)

Push (4)

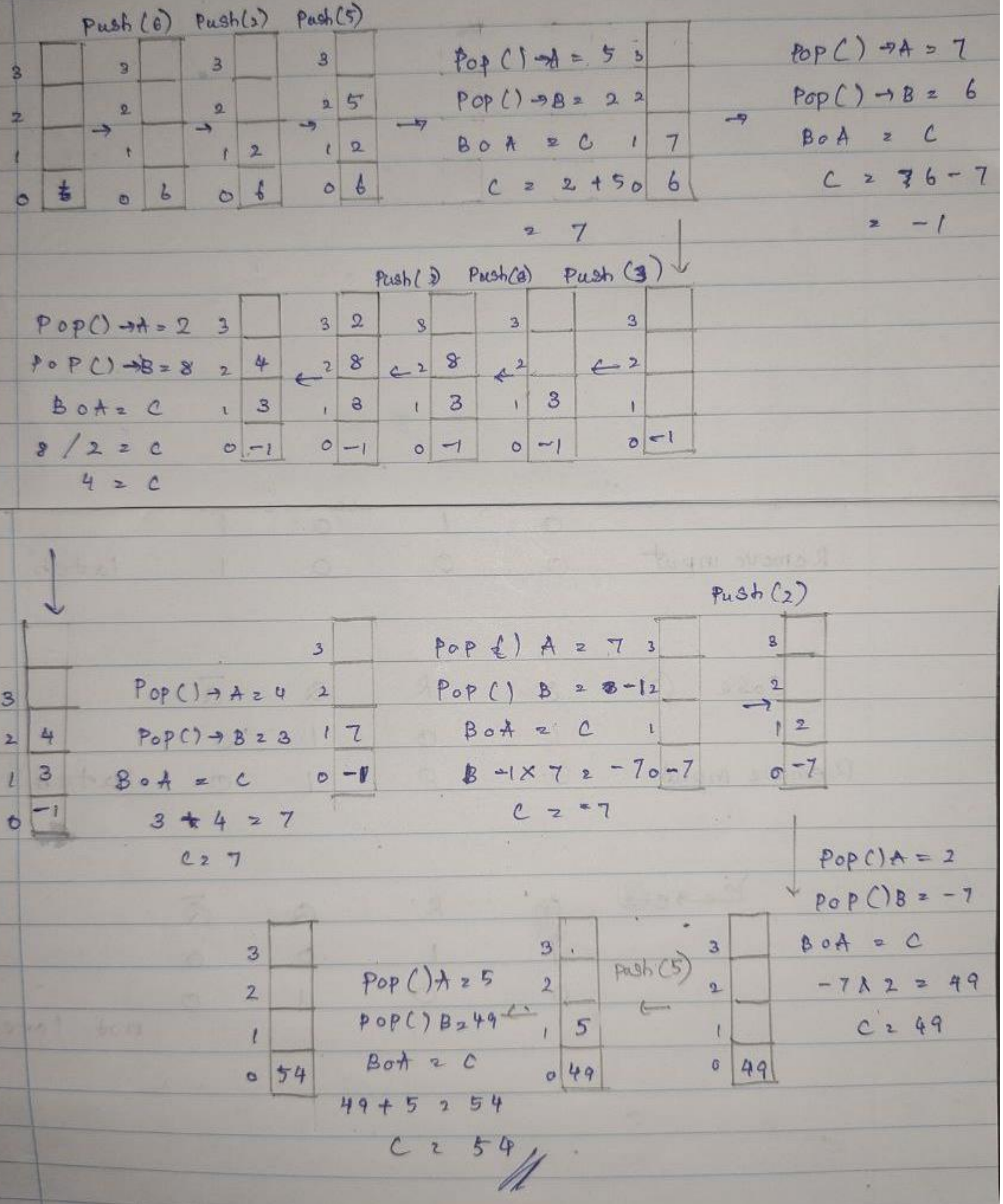
Pop () → A = 4
Pop () → B = 3
→ B o A = 3 × 4
C = 12

Push (5)

Pop() → A = 5
Pop () → B = 12
B o A = C
C = 12 + 5
= 17

Pop() → A = 17
Pop() → B = 2
B o A = C
C = 2 − 17
= −15

~~625~~ 6 2 5 + - 3 8 2 / + ** 2 ^ 5 +



Push(6) Push(2) Push(5)

Pop() → A = 5
Pop() → B = 2
B o A = C
C = 2 + 5
= 7

Pop() → A = 7
Pop() → B = 6
B o A = C
C = ~~7~~6 − 7
= −1

Push(3) Push(8) Push(2)

Pop() → A = 2
Pop() → B = 8
B o A = C
8 / 2 = C
4 = C

Pop() → A = 4
Pop() → B = 3
B o A = C
3 * 4 = 7
C = 7

Pop() A = 7
Pop() B = ~~3~~ −1
B o A = C
B −1 × 7 = −7
C = −7

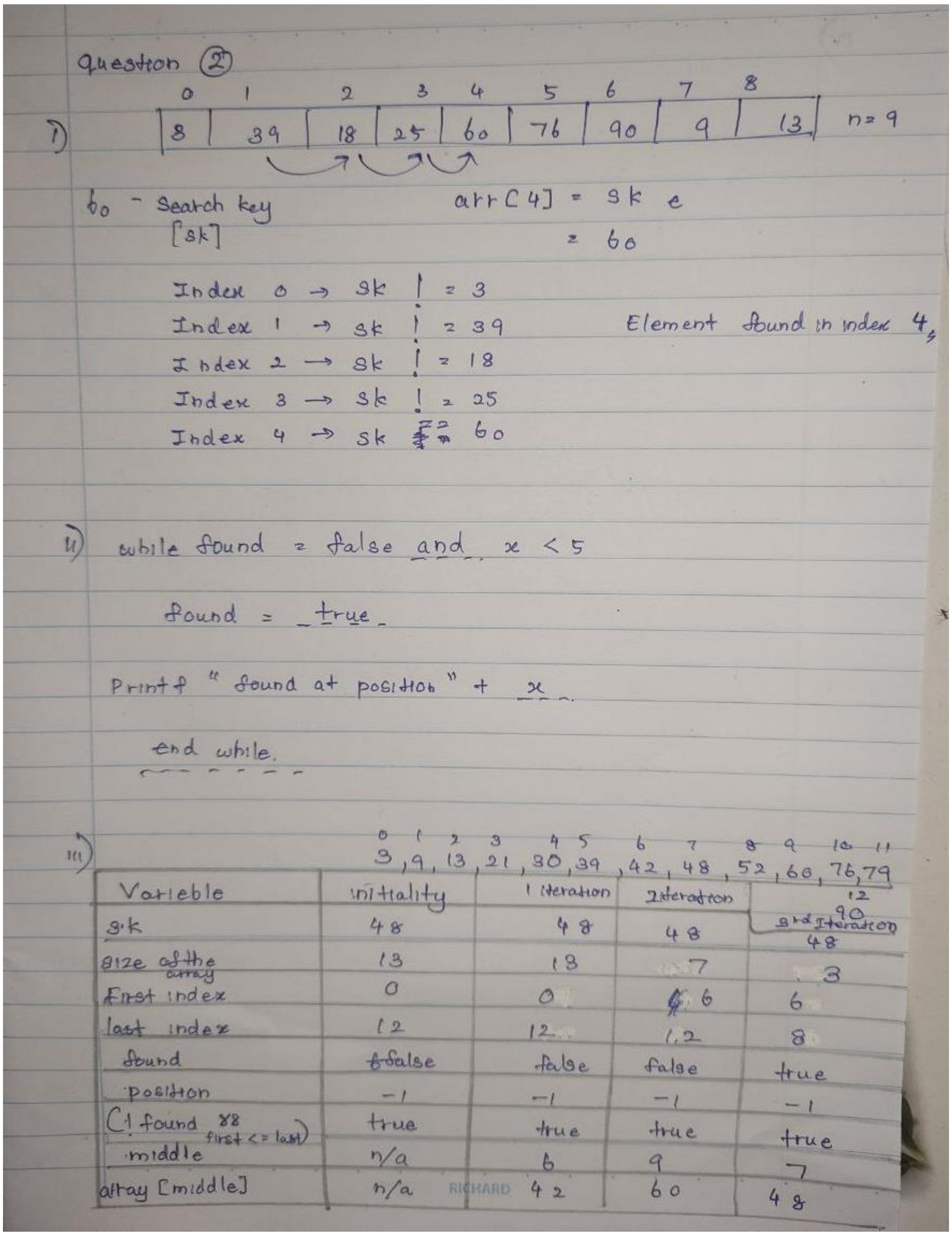
Push(2)

Pop() A = 2
Pop() B = −7
B o A = C
−7 ^ 2 = 49
C = 49

Push(5)

Pop() A = 5
Pop() B = 49
B o A = C
49 + 5 = 54
C = 54

Question 2



question ②

i)

0	1	2	3	4	5	6	7	8
8	39	18	25	60	76	90	9	13

n = 9

60 - Search key [sk]

arr[4] = sk e

= 60

Index 0 → sk != 3

Index 1 → sk != 39

Index 2 → sk != 18

Index 3 → sk != 25

Index 4 → sk == 60

Element found in index 4,

ii) while found = false and x < 5

found = true

Printf " found at position " + x

end while.

iii)

0	1	2	3	4	5	6	7	8	9	10	11	12
3	9	13	21	30	39	42	48	52	66	76	79	90

Varieble	initiality	1 iteration	2 iteration	3rd Iteration
s.k	48	48	48	48
size of the array	13	13	7	3
First index	0	0	~~4~~ 6	6
last index	12	12	12	8
found	~~f~~ false	false	false	true
position	-1	-1	-1	-1
C! found && first <= last)	true	true	true	true
middle	n/a	6	9	7
array [middle]	n/a	42	60	48

iv) <u>Linear search</u>

```
for ( ) {
if ( ) { }
if ( ) { }
  }
```

	step	times
for ( ) {	1	0
if ( ) { }		
if ( ) { }	1.1	1
}	1.2	1
		$n + 2$

$\therefore O(n)$

<u>binary search</u>

```
while ( ) {
      void . (s + e) / 2 ;
 if ( ) { }
  else if ( ) { }
   else ( ) { }
 }
```

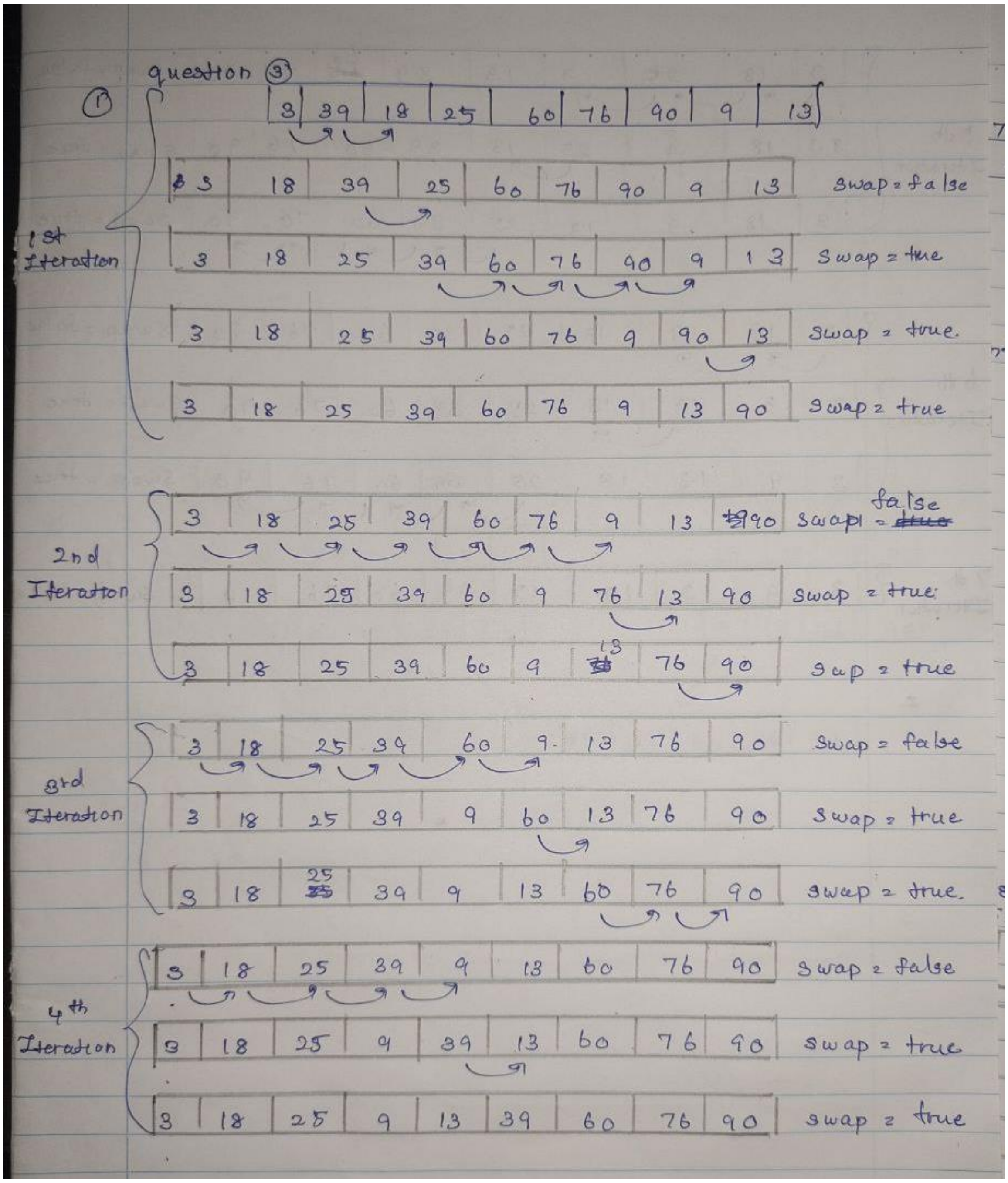
step	times
1	n
1.1	$n/2$
1.2	1
1.3	1
1.4	1

$n + \frac{n}{2} + 3 = \frac{3n-}{2}$

$\log_2 n$

$O(\log n)$

Question 3



question 3

1

3	39	18	25	60	76	90	9	13

1st Iteration

3	18	39	25	60	76	90	9	13	swap = false
3	18	25	39	60	76	90	9	13	swap = true
3	18	25	39	60	76	9	90	13	swap = true.
3	18	25	39	60	76	9	13	90	swap = true

2nd Iteration

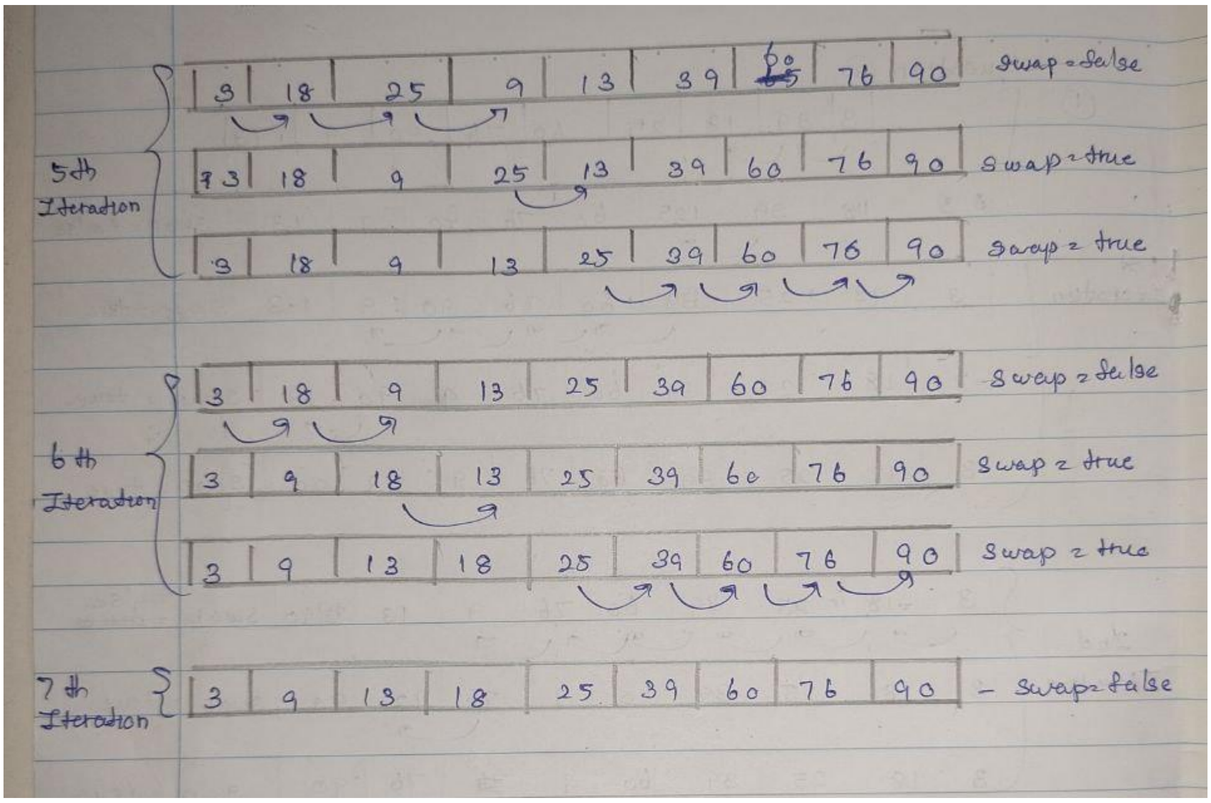
3	18	25	39	60	76	9	13	90	swap = false ~~true~~
3	18	25	39	60	9	76	13	90	swap = true
3	18	25	39	60	9	13	76	90	sap = true

3rd Iteration

3	18	25	39	60	9	13	76	90	swap = false
3	18	25	39	9	60	13	76	90	swap = true
3	18	25	39	9	13	60	76	90	swap = true.

4th Iteration

3	18	25	39	9	13	60	76	90	swap = false
3	18	25	9	39	13	60	76	90	swap = true
3	18	25	9	13	39	60	76	90	swap = true



5th Iteration

3	18	25	9	13	39	~~65~~ 60	76	90	swap = false
~~7~~ 3	18	9	25	13	39	60	76	90	swap = true
3	18	9	13	25	39	60	76	90	swap = true

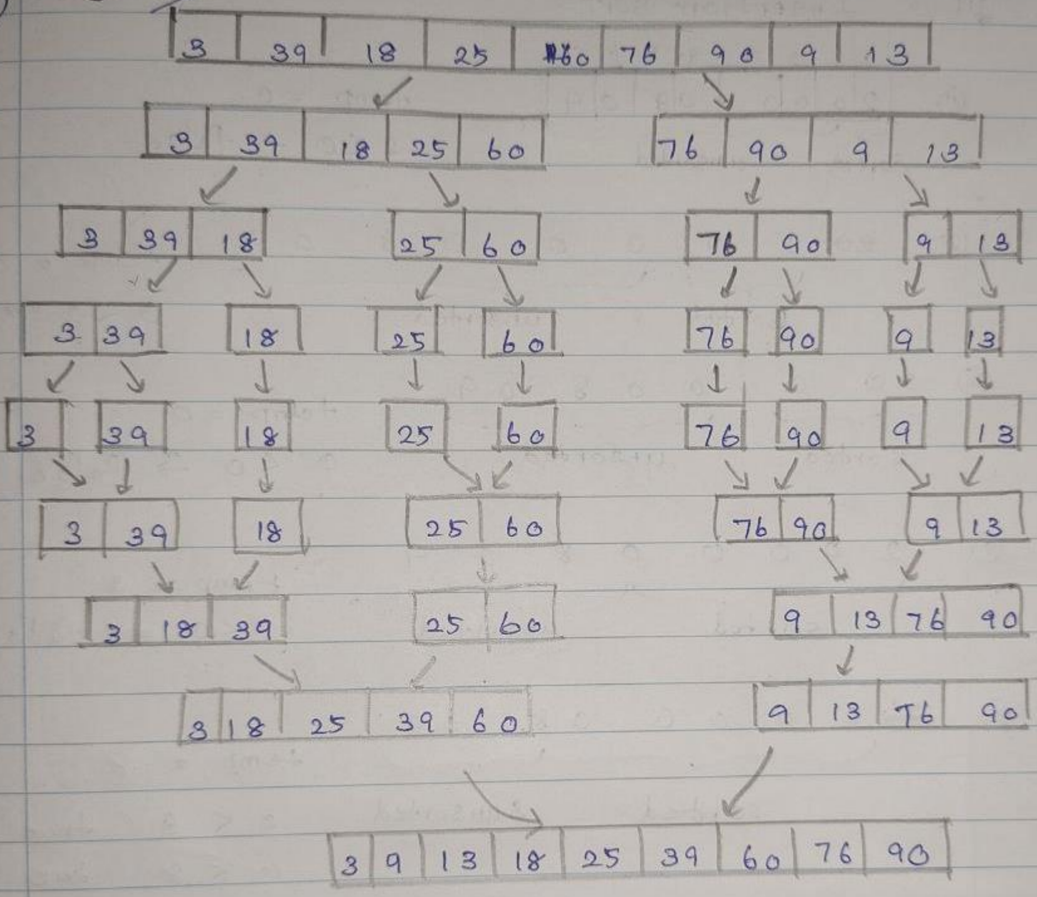
6th Iteration

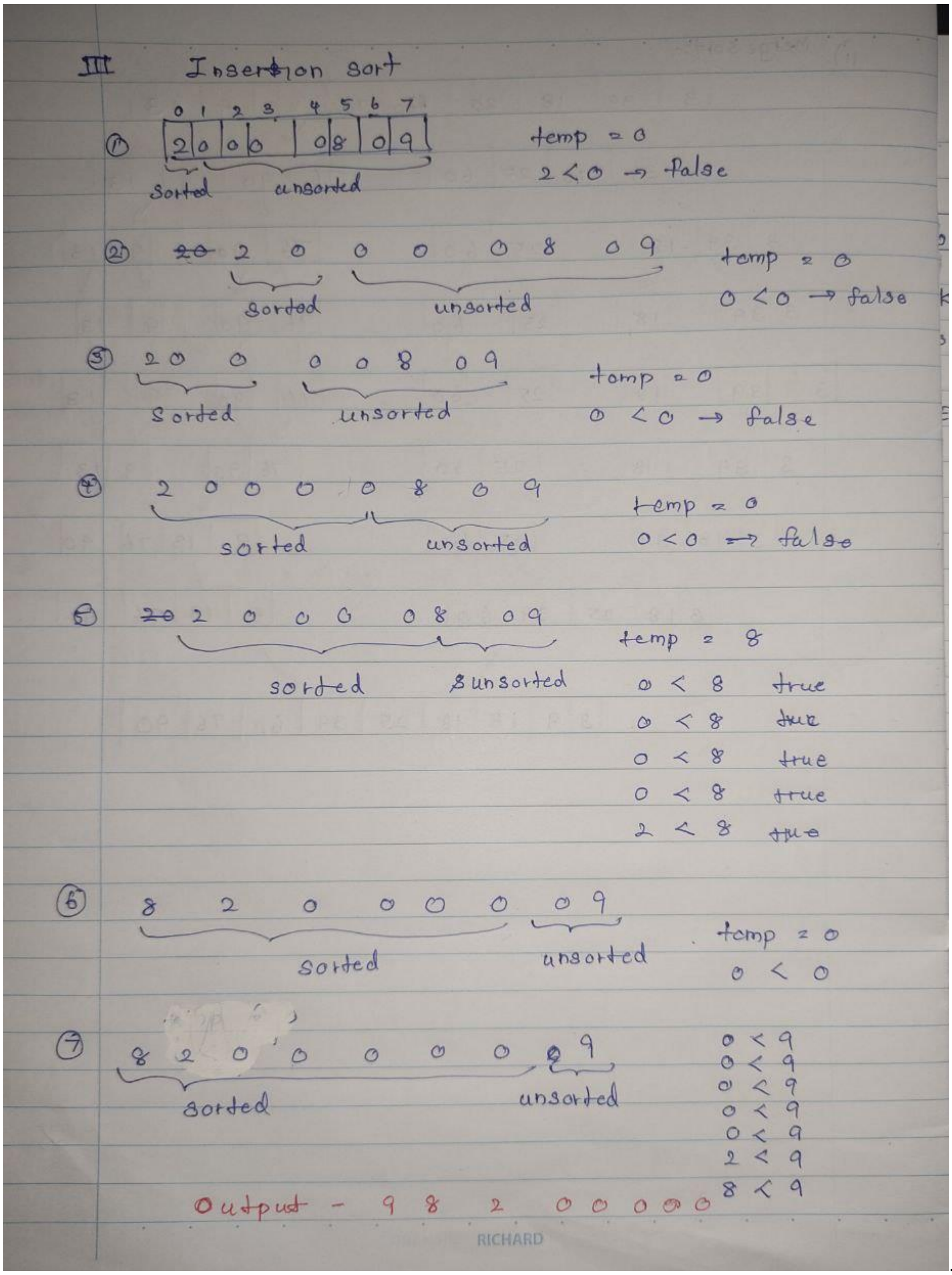
3	18	9	13	25	39	60	76	90	swap = false
3	9	18	13	25	39	60	76	90	swap = true
3	9	13	18	25	39	60	76	90	swap = true

7th Iteration

3	9	13	18	25	39	60	76	90	– swap = false

11) Merge sort





III Insertion sort

①

0	1	2	3	4	5	6	7
2	0	0	0	0	8	0	9

Sorted unsorted

temp = 0

$2 < 0 \rightarrow$ false

② ~~20~~ 2 0 0 0 0 8 0 9

sorted unsorted

temp = 0

$0 < 0 \rightarrow$ false

③ 2 0 0 0 0 8 0 9

Sorted unsorted

temp = 0

$0 < 0 \rightarrow$ false

④ 2 0 0 0 0 8 0 9

sorted unsorted

temp = 0

$0 < 0 \rightarrow$ false

⑤ ~~20~~ 2 0 0 0 0 8 0 9

sorted unsorted

temp = 8

$0 < 8$ true

$0 < 8$ true

$0 < 8$ true

$0 < 8$ true

$2 < 8$ true

⑥ 8 2 0 0 0 0 0 9

sorted unsorted

temp = 0

$0 < 0$

⑦ 8 2 0 0 0 0 0 9

sorted unsorted

$0 < 9$

$0 < 9$

$0 < 9$

$0 < 9$

$0 < 9$

$2 < 9$

$8 < 9$

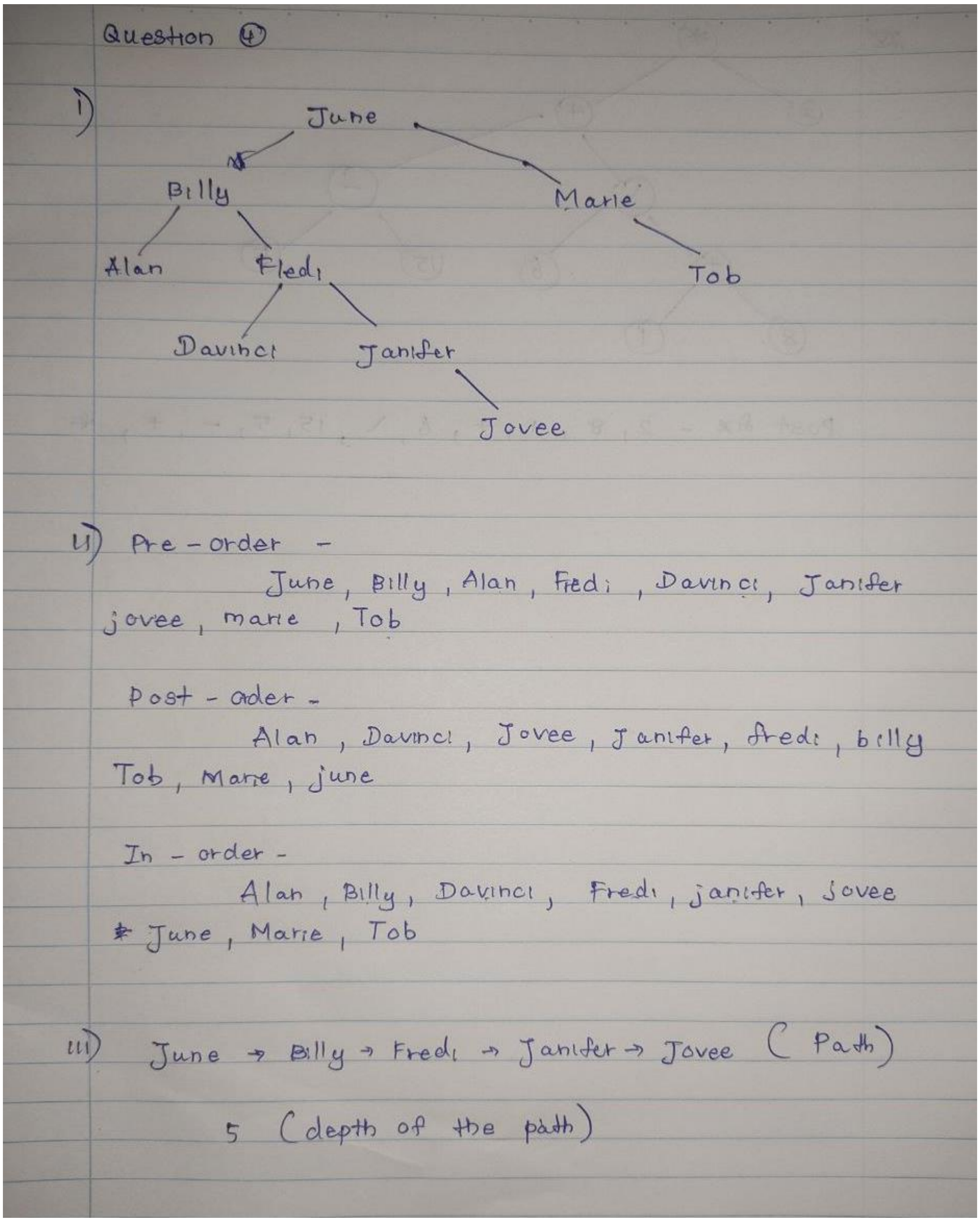
Output - 9 8 2 0 0 0 0 0

.

4)

It’s better to use the selection sort, since it uses less number of swaps than the other sorting algorithms. Also, the big O value of the selection sort is also less than the other sorting algorithms, which means its complexity and the memory.selection sort makes 0(n) swaps which is minimum among all sorting algorithms. Bwcause of that I prefer selection sort.

Question 4



Question ④

i)

$$\begin{tikzcd}
 & & & \text{June} \arrow[dll, dash] \arrow[drr, dash] & & & \\
 & \text{Billy} \arrow[dl, dash] \arrow[dr, dash] & & & & \text{Marie} \arrow[dr, dash] & \\
\text{Alan} & & \text{Fredi} \arrow[dl, dash] \arrow[dr, dash] & & & & \text{Tob} \\
 & \text{Davinci} & & \text{Janifer} \arrow[dr, dash] & & & \\
 & & & & \text{Jovee} & &
\end{tikzcd}$$

ii) Pre-order -

June, Billy, Alan, Fredi, Davinci, Janifer jovee, marie, Tob

Post-oder -

Alan, Davinci, Jovee, Janifer, fredi, billy Tob, Marie, june

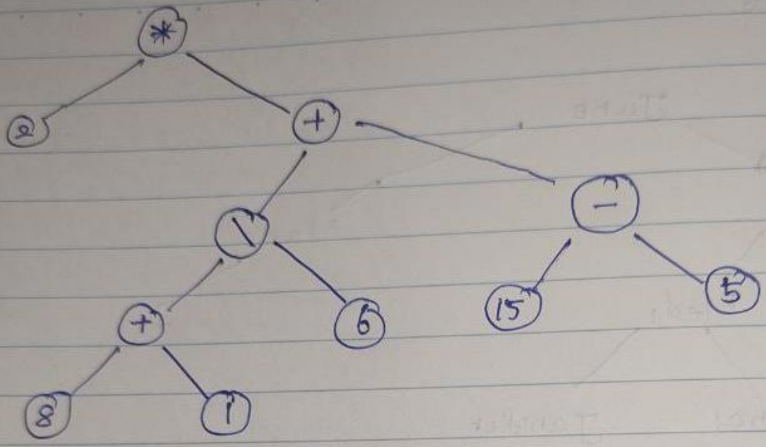
In-order -

Alan, Billy, Davinci, Fredi, janifer, jovee ~~*~~ June, Marie, Tob

iii) June → Billy → Fredi → Janifer → Jovee (Path)

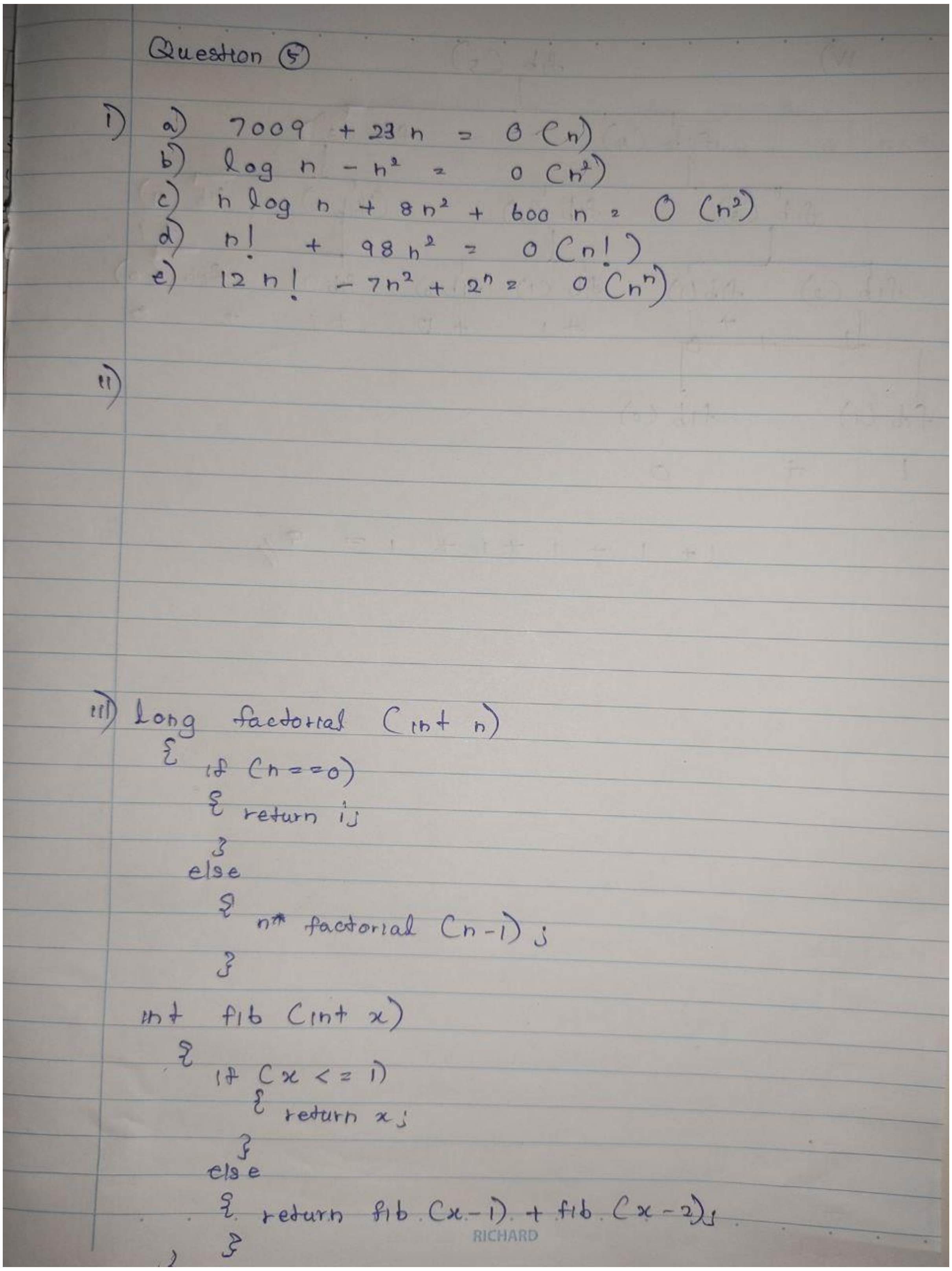
5 (depth of the path)

IV



Post fix - 2, 8, 1, +, 6, \, 15, 5, −, +, *

Question 5



Question (5)

i) a) $7009 + 23n = O(n)$

b) $\log n - n^2 = O(n^2)$

c) $n \log n + 8n^2 + 600n = O(n^2)$

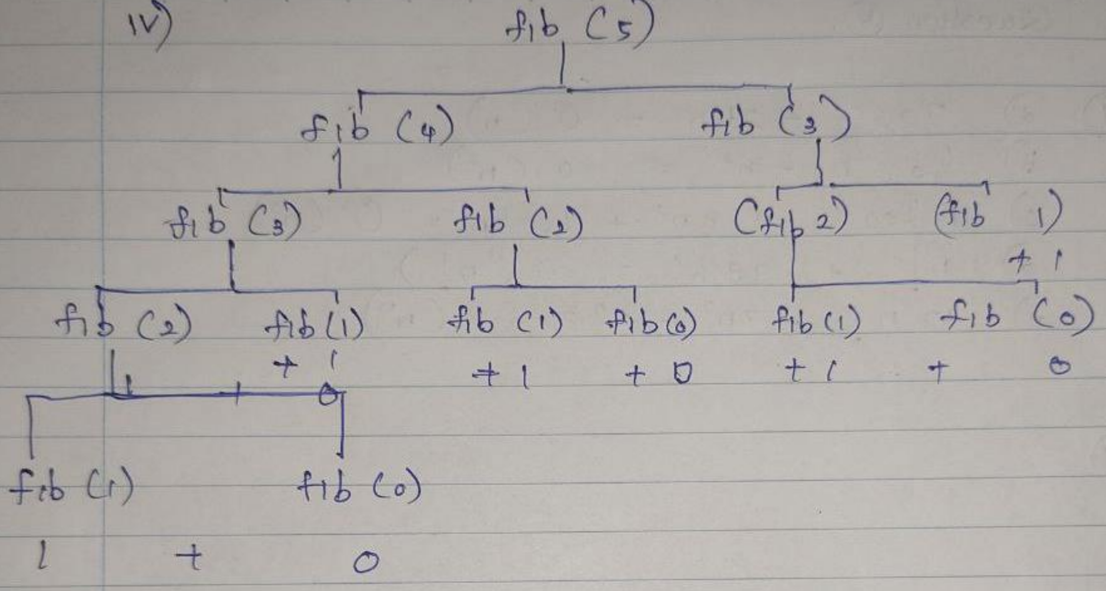
d) $n! + 98n^2 = O(n!)$

e) $12n! - 7n^2 + 2^n = O(n^n)$

ii)

iii)
```
Long factorial (int n)
  {
     if (n==0)
     { return 1;
     }
    else
     {
        n* factorial (n-1);
     }
int fib (int x)
  {
     if (x <= 1)
       { return x;
      }
     else
     { return fib (x-1) + fib (x-2);
     }
  }
```

iv)


fib (5)
fib (4)
fib (3)
fib (3)
fib (2)
(fib 2)
(fib 1)
+ 1
fib (2)
fib (1)
+ 1
fib (1)
fib (0)
+ 1
+ 0
fib (1)
fib (0)
+ 1
+
0
1
+
0
fib (1)
fib (0)
1
+
0

1 + 1 + 1 + 1 + 1 = 5