UNIVERSITY PARTNER
UNIVERSITY OF WOLVERHAMPTON

iAcademy
Quality Education Improving Lives

HERALD COLLEGE KATHMANDU

Academic Year	Module	Assessment Number	Assessment Type
S20	Introductory Data Structures and Algorithms (DipIT02)	A1	Assignment Submission

[Assignment Submission]

Student Id : [NP03A190299]
Student Name : [Yogesh Shrestha]
Section : [DC8]
Module Leader : [Mr. Prakash Gautam]

Submitted on : 06-03-2020

Page No.
Date : / /

Tutorial - 3

1-> A data structure which always preserve the previous version of itself when it is modified is called persistent data structure.
A data structure that destroys the previous version of itself after update is called ephemeral data structure.

2-> For n number of disc in tower of honoi it takes $2^n - 1$ moves
for 3 number of disc in tower of honoi it takes $2^3 - 1$ moves $= 2^3 - 1$
$= 8 - 1$
$= 7$ moves
∴ the time complexity is $O(2^n)$.

Pseudocode.

```
Void towers of Hanoi(int n, source, dest, aux)
  if (n == 1){
  printf (Disk, n from source to dest);
    }
    else{
    tower of Hanoi(n-1, source, aux, dest)
    printf( Disk, n from source to dest)
    tower of Hanoi (n-1, aux, dest, source)
  } for time from
```

Page No.
Date : / /

For time complexity

$T(n) = T(n-1) + 1 + 1 \quad (n-1)$

$= 1 + 2 + T(n-1) \quad \text{--- (i)}$

Let's substitute $T(n)$ with $n-1$, we get

$T(n-1) = 1 + 2 + T((n-1)-1)$

$= 1 + 2 + T(n-2) \quad \text{--- (ii)}$

Now, substitute n with $n-2$ we get

$T(n-2) = 1 + 2 + T((n-2)-1)$

$= 1 + 2 + T(n-3) \quad \text{--- (iii)}$

Again, substitute eqn (ii) in eqn (i) we get

$T(n) = 1 + 2 + (1 + 2 + T(n-2))$

$= 1 + 2 + 4 + T(n-2) \quad \text{--- (iv)}$

Then, substitute eqn (iii) in eqn (iv) we get:

$= 1 + 2 + 4 + (1 + 2) + T(n-3)$

$= 7 + 8 + T(n-3)$

$= 7 + 8 + (1 + 2T(n-4))$ [∵ adding $T(n-3)$]

$= 15 + 8T(n-4)$

We can generalize from above.

$= 2^k - 1 + 2^k + T(n-k)$

$= 2^{n-1} - 1 + 2^n \cdot T(n-(n-1))$

Page No.
Date : / /

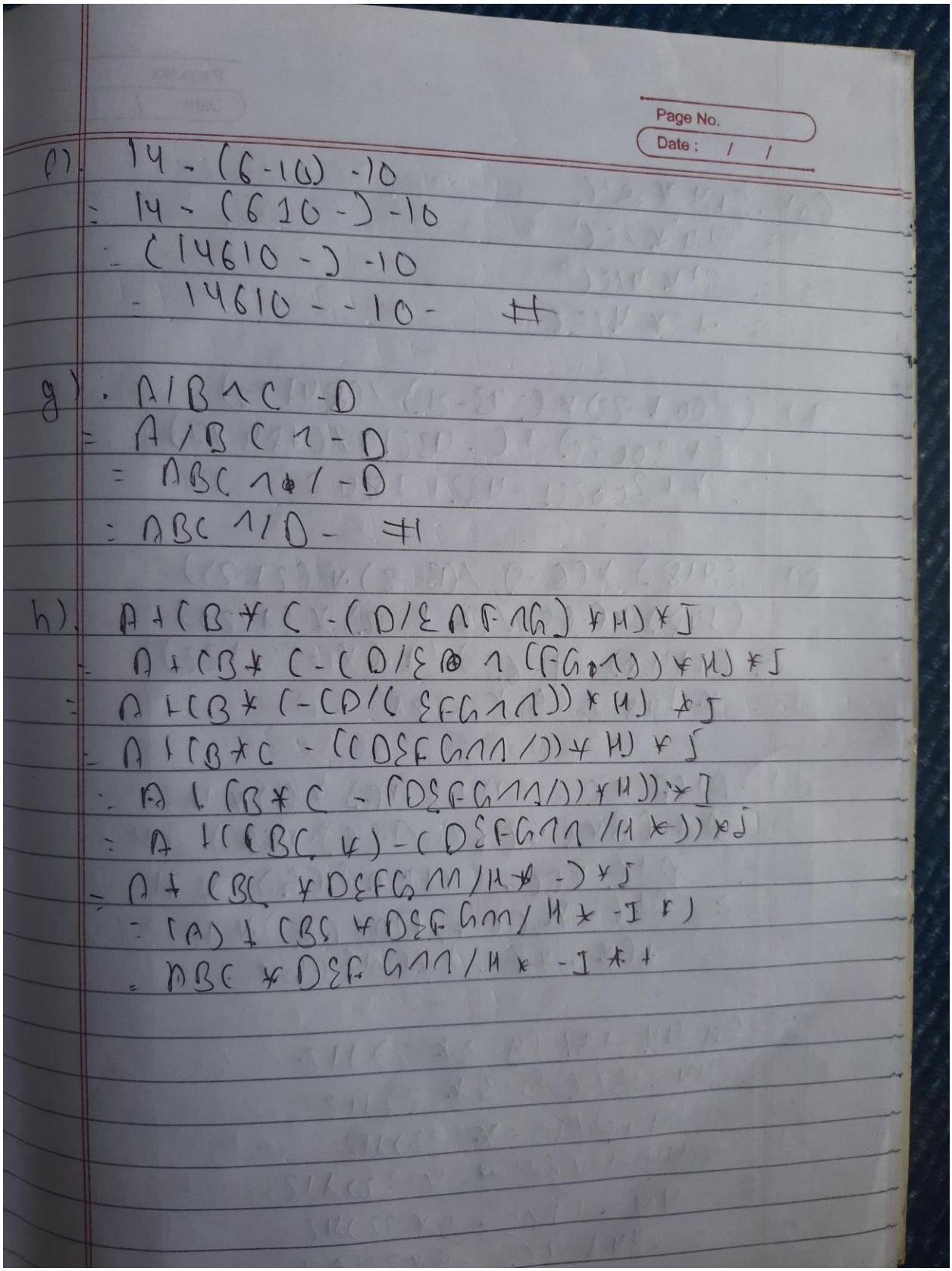
5 a) 3+4*5/6
= 3+(45*)/16)
= 3+(456*1)
= 345(*1+

b) (300+23)*(43-21)/(84+7)
= (30023+)*(4321-)/(847+)
= (3002314321-*)/(847+)
= 30023+4321-*847+1 #

c) (4+8)*(6·5)/(32-2)*(?;?)
= (48+)*(65-)/(32-)*(22+)
= (48+66-*)/(32-22+*)
= 48+66-*32-22+*/ #

d) A+(B*C=(D/E^F]*G)*H
= A·1(B(*(DE^FA)*C)*H
= A·1(BC*(DE^FA]C*/)*H
= A+(BC*DE^FA]C*-)*H
= A+(BC*DE^FA]C*-H*)
= ABC*DE^FA]C*-H*+ #

e) 44-15/(9-3*2)+12
= 44-15/(9-32*)+12
= 44-15/(932*-)+12
= 44-(159 32*-*/)+12
= 44159 32*-1-12+ #



f) 14 - (6-10) -10
= 14 - (6 10 -) -10
= (14610 -) -10
= 14610 - -10 - #

g) A/B^C -D
= A/BC^ -D
= ABC^*/ -D
= ABC^/D - #

h) A+(B*C-(D/E^F-^G)*H)*I
= A+(B*C-(D/E*F-^(G*^))*H)*I
= A+(B*C-(D/(E*F-^(^*G))*H)*I
= A+(B*C-((D*E*F-G^^/))*H)*I
= A+(B*C-(D*E*F-G^^/)*H))*I
= A+((BC*)-(D*E*F-G^^/H*))*I
= A+(BC*D*E*F-G^^/H*-)*I
= (A)+(BC*D*E*F-G^^/H*-I*)
= ABC*D*E*F-G^^/H*-I*+

Page No.
Date : / /

6a) $3+4*.5/6$
$= 3+4*/5(6$
$= 3+*4/5(6$
$= .+3*4/56$

b) $(300+23)*(43-21)/(84+17)$
$= (+30023)+(-4321)/(184/17)$
$= *+300231-4321\ 184?$

c) $(4+8)*(6-5)/(13-2)*(2+2))$
$= (14*)*(-65)/(6\ 1-32)*((22+$
$= (1+4*)+1-65)/1(*-32+22))$
$= *1*8/-65*-32+??$

d) $\cdot A1(B*C-(1)1\}\wedge(-)*G)*H$
$- A+(B*(-(1)\wedge\{F)+(G)*H$
$= A+(B*C-(A10\wedge\{-G)*H$
$= +V*-*BC*/O\wedge\{-G)*H$

e) $44-15/(9-3*2)+12$
$= 44-15/(9-3*2)+12$
$= 44-15/(9-*32)+12$
$= 44-15/(9-6*32)+12$
$= 44-(1/5-9*32)+12$
$= -44+1/5-9*3210?$

Page No.
Date : / /

f) $14-(6-10)-10$
$= 14-(-6+10)-10$
$= -14-6+10+10$
$= -14--6+10+10$

g) $A+B+C-D$
$= A+(+BC)-D$
$= A+BC-D$
$= -(A+BCD)$

h) $A+(B*C-(D/E+F-G)*H)*J$
$= A+(*BC)-(D)(+E+FG)*H)*J$
$= A+((*BC)-(D/(+E+FG)*H))*J$
$= A+((*BC)-(D/+E+FG)*H))*J$
$= A+(-*BC*HD/+E+FG)*J$
$= A+-*BC*HD/+E+FG*J$

8a) $23+$
$= 2+3$
$= 5$

b) $78*$
$= 7*8$
$= 56$

c) $62/5+$
$= (6+2)/5+$
$= 3+5$
$= 8$

d) $56/2+$
$= 5(6/2)+$
$= 5+(6/2)$
$= 5+3$
$= 8$

e) $65*73-48+*$
$= (6*5)(7-3)(4+8)*+$
$= (30)+(84)*(12)$
$= 30+48$
$= 78$

Page No.
Date : / /

9 a) 6S+73-48+V+

+

= 8 x

x 3 4 12 +

S S 7 4 4 4 48

C 6 6 30 30 30 30 30 30 78

∴ the total value is 78.

b 38+S+10-3/4+S+7

Soln

+ + - ÷ / + + -

8 S 10 3 4 S 7

3 3 11 16 6 2 6 11 4

∴ The total value is 4.

c. 14 6 10 -- 10 -

—

10 ÷

6 6 4 -

4 4 4 4 10

1 1 1 1 1 1 9

∴ The total value is 9.

Page No.
Date : / /

10a) +xy
= x+y

b) 2/84
/ +
8 8 2
4 4 4 2 4
∴ Total value = 4

c) +*a-bc/de+-fgh
Soln
- - -
f f + d / b
g g g g-af e d-e c
h h h h h h*g-f h*g-f h*g+/e-d

+
a *
b-c a*(-b

(h*g-f)/(e-d) (h*g-f) (e-d) (h*g-f)/
(e-d) *(a*(-b)
∴ (a*b-c) *(d-e) /f(-g+h) is
the final form.

d) +1(25
/ 2 +
+ + 6 3 5 8

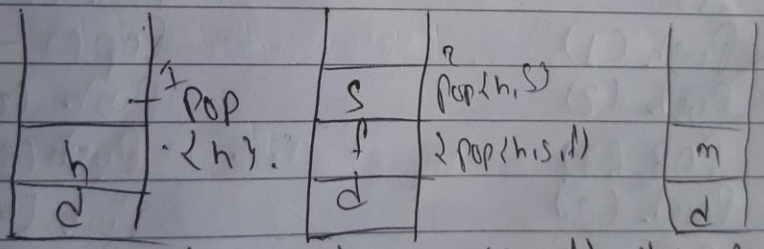
Page No.
Date : / /

11. Sol:

```
void reverse(){
  int stack, array;
  for (int) i=0 to size of array)
  {
    stack.push(array[i])
  }
  int[] reverse Array;
  for (int i=0 to size of array){
    reverse array[i] = stack.pop();
  }
}
```

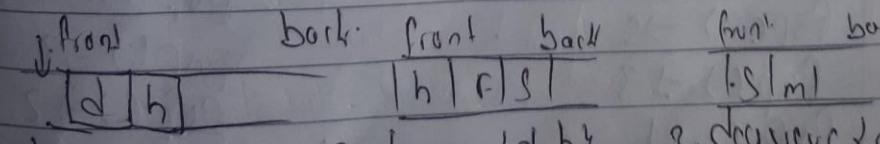
12 →)

a →



∴ Sequence of popped value (h,s,f) : stack of stack from top to bottom) : md.

b -



1. dequ. (d) 2. dequeue(d,h). 3. dequeue(d,h,f)

∴ Sequence of dequeued value (d,h,f).
remaining value (s,m)

Page No.
Date : / /

13-> Soln.

a = 10
b = 3
c = -2

Now, aa+bc-*
= (a+a) (b-c)*
= (a+a) *(b-c)
= (10+10) * (3-(-2)
= 20 * 5
= 100

14a) 2, 4, 5, 3, 1.

Sol:
Push (1)
Push (2)
POP (2)
Push (3)
Push(4)
POP() →4
Push (5)
POP() →5
POP() →3
POP() →1

∴ 2, 4, 5, 3, 1 is ~~equal~~.

b). 1, 5, 4, 2, 3

Push (1)
Pop() → 1
Push (2)
Push(3)
Push (4)
Push(5)
Pop() →3
Pop() →4
pop() →3
Pop() →2

∴ which is not equal to 1, 5, 4, 7

Page No.
Date : / /

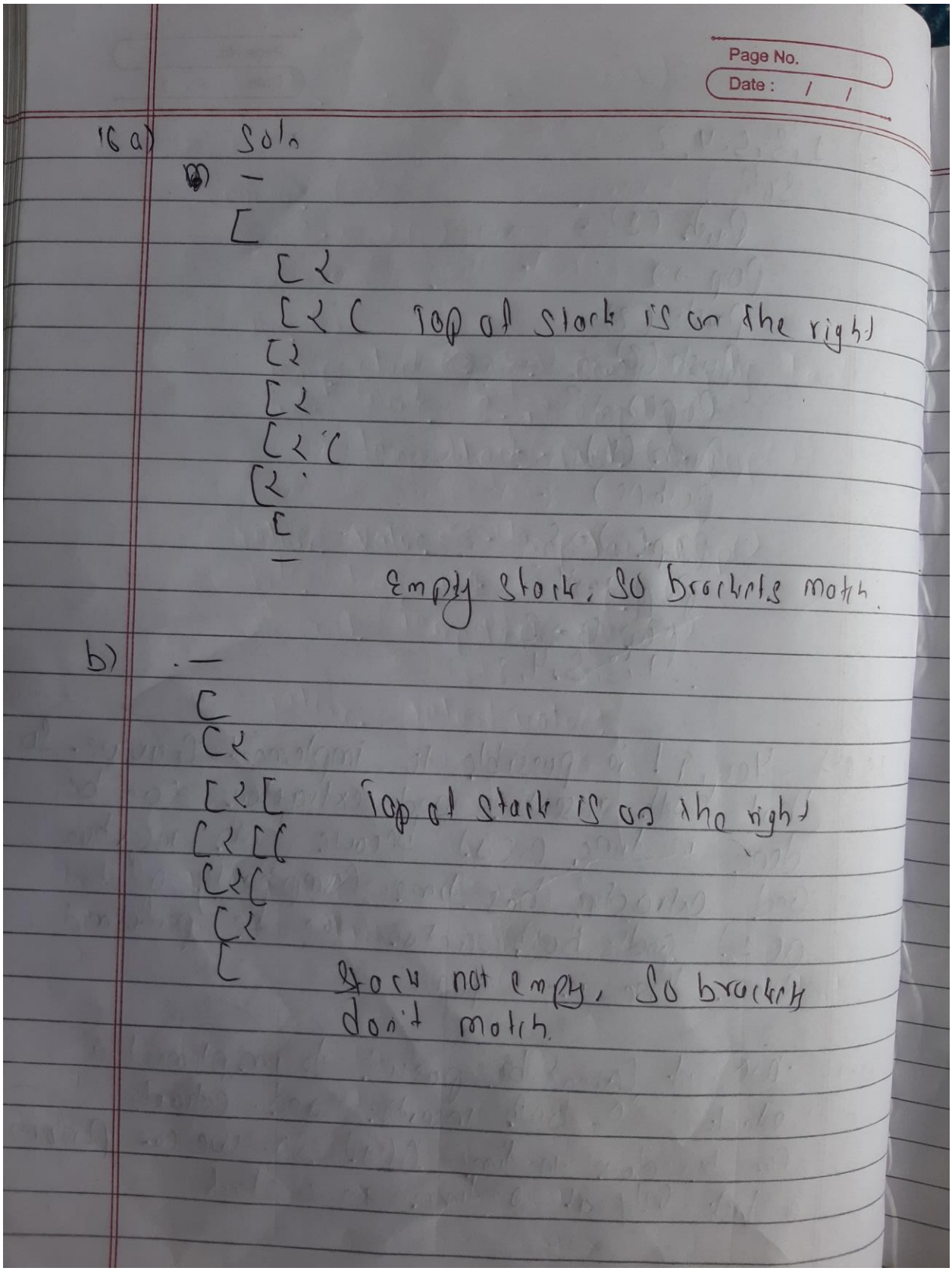
(a) 1, 3, 5, 4, 2

Soln:

Push (1)
Pop → 1
Push (2)
Push (3)
Pop() → 3
Push (4)
Push (5)
Pop() → 5
Pop() → 4
Pop() → 2
∴ 1, 3, 5, 4, 2

Is a → Yes, It is possible to implement a queue. So that both insertion and extraction can be done in time O(1) because both insertion and extraction has time complexity of O(1) and both action can be performed same time in a queue.

b → No, it cannot be possible to implement a stack. So both insertion and extraction can be done in time O(1) so, we can perform a task only at a time in a stack.



Page No.
Date : / /

17-> Sol.n,

A		A	A
B		B	B
C	B	D	C
D	A	C	D
S_1		S_2	S_3	S_3

S_2. push (S_1. pop)
S_2. push (S_1. pop)
S_3. push (S_1. pop)
S_3. push (S_1. pop)
S_3. push (S_2. pop)
S_3. push (S_2. pop)

18->

A		B		B
B	C	D		D
C	B	A		A
D	A	C	B	C
S_1	S_2	S_3	S_4	S_3

A			B			B
B			D			D
C	C		A	A	C	A
D	B	A	C	D	B	C
S_1	S_2	S_3	S_3			

Page No.
Date : / /

19→ Sol:

```
Enqueue (e){
    S.push(e)
}

Deque (){
    tempStack ◎ → new empty stack;
    while( S.isEmpty ()
    {   tempStack.push (S.pop())
    }
    return.value ← tempStack.pop()
    while (tempStack is empty ()){
    S.push (tempstack.pop())
    }
    return returnvalue;
}
}
```

7a: 3 + 4 * 5 / 6

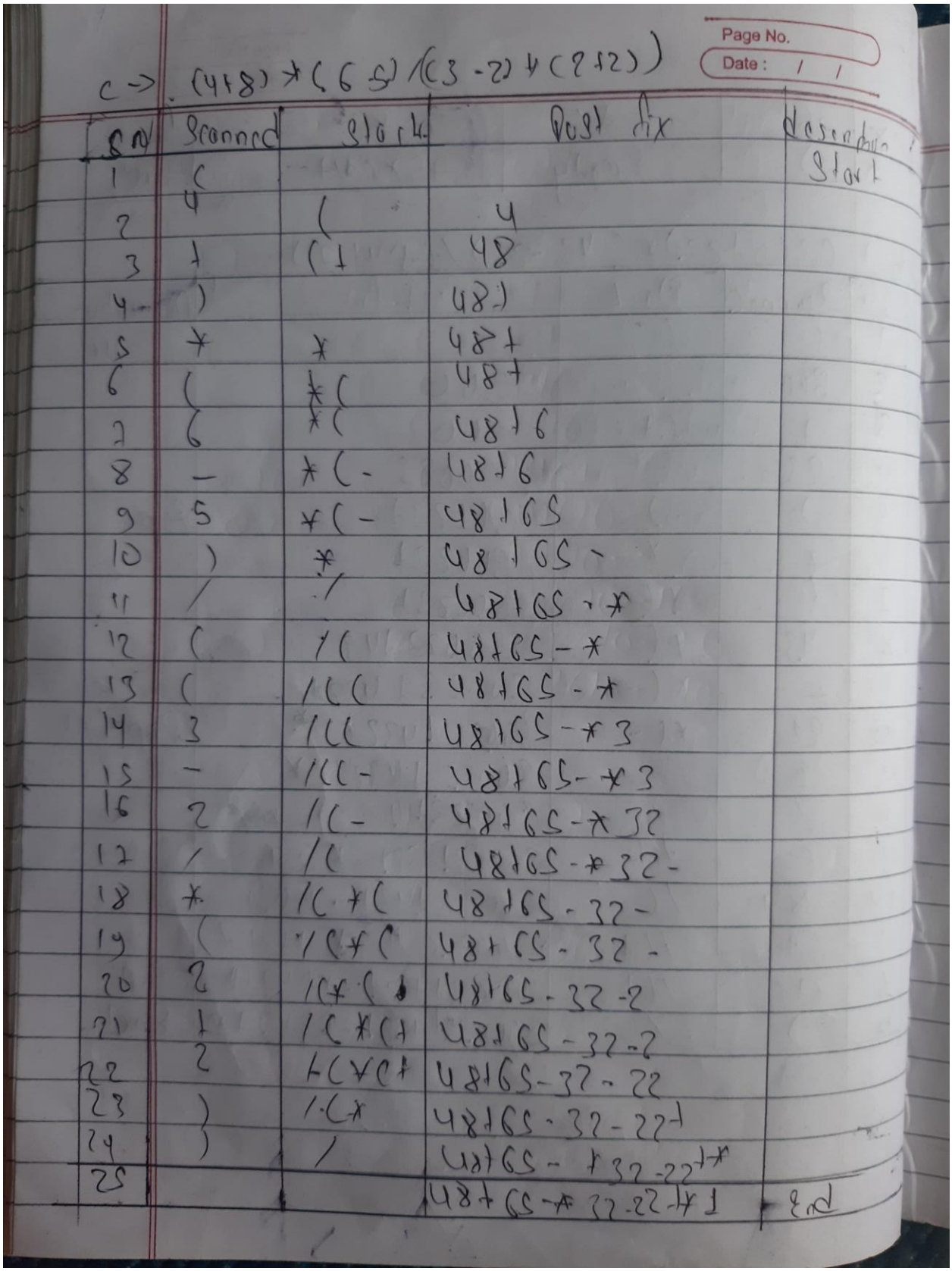
S.no	Scanned	Stack	Postfix Expression	Description
1		(		Start
2	3	(	3	
3	+	(+	3	
4	4	(+	34	
5	*	(+*	34	
6	5	(+*	345	
7	/	(+/	345*	

Page No.
Date : / /

8	.6	(+/	345*6	
9	)	Empty	345*(6/+	End

b) (300+23)*(43-21)/(54+7)

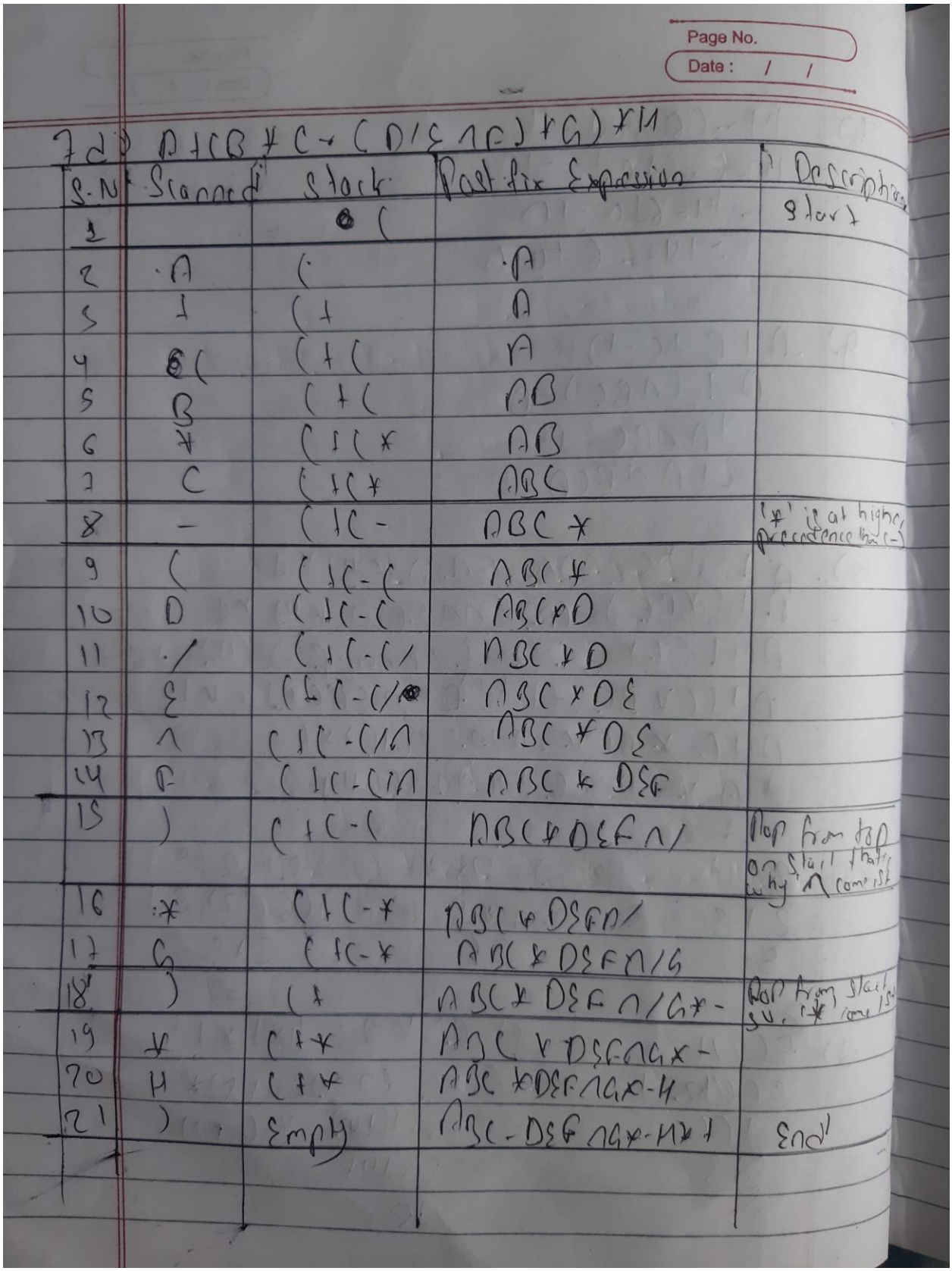
S.no	Scanned	Stack	Post fix	Description
1	(	(		Start
2	300	(	300	
3	+	(+	300	
4	23	(+	30023	
5	)	)	30023+	Pop
6	*	*	30023+	
7	(	*(	30023+	
8	43	*(	30023+43	
9	-	*(-	30023+43	
10	21	*(-	30023+4321-	Pop (*)
11	)	)	30023+4321-*	
12	/	/(	30023+4321-*	
13	(	/(	30023+4321-*54	
14	54	/(	30023+4321-*54	
15	+	/(+	30023+4321-*547	
16	+	/(+	30023+4321-*547+	Pop
17	)	/	30023+4321-*547+*	End



c -> (4+8)*(6-5)/(3-2)*(2+2)

Page No.
Date : / /

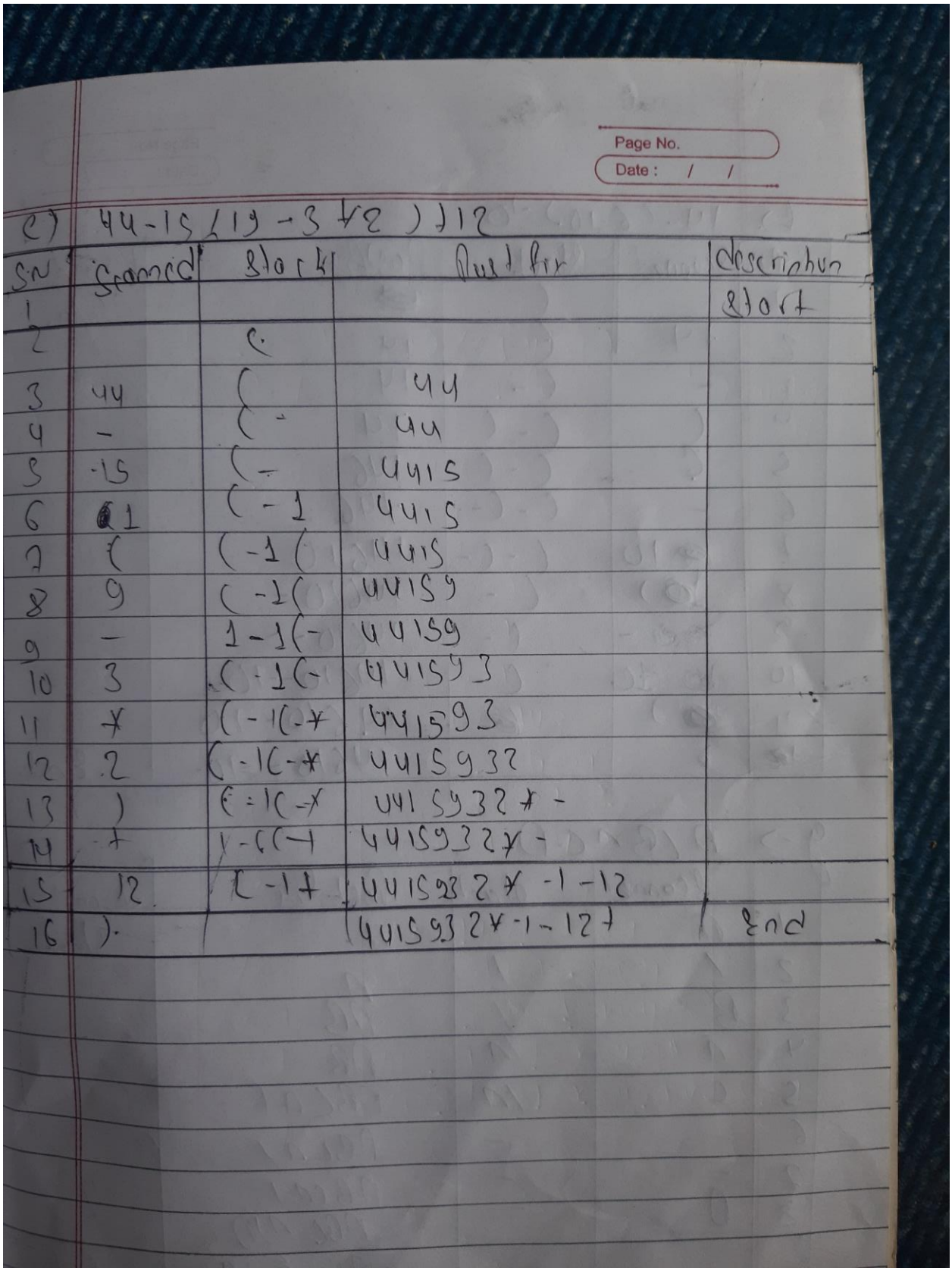
SN	Scanned	Stack	Post fix	Description
1	(			Start
2	4	(	4	
3	+	(+	48	
4	)		48+	
5	*	*	48+	
6	(	*(	48+	
7	6	*(	48+6	
8	-	*(-	48+6	
9	5	*(-	48+65	
10	)	*	48+65-	
11	/	/	48+65-*	
12	(	/(	48+65-*	
13	(	/((	48+65-*	
14	3	/((	48+65-*3	
15	-	/((-	48+65-*3	
16	2	/(-	48+65-*32	
17	)	/(	48+65-*32-	
18	*	/(*	48+65-32-	
19	(	/(*(	48+65-32-	
20	2	/(*(	48+65-32-2	
21	+	/(*(+	48+65-32-2	
22	2	/(*(+	48+65-32-22	
23	)	/(*	48+65-32-22+	
24	)	/	48+65-*32-22+*	
25			48+65-*32-22+*	End



Page No.
Date : / /

7d) A+(B*C-(D/E^F)+G)*H

S.No	Scanned	stack	Post fix Expression	Description
1		(		start
2	A	(	A	
3	+	(+	A	
4	(	(+(	A	
5	B	(+(	AB	
6	*	(+(*	AB	
7	C	(+(*	ABC	
8	-	(+(-	ABC*	'*' is at higher precedence than (-)
9	(	(+(-(	ABC*	
10	D	(+(-(	ABC*D	
11	/	(+(-(/	ABC*D	
12	E	(+(-(/	ABC*DE	
13	^	(+(-(/^	ABC*DE	
14	F	(+(-(/^	ABC*DEF	
15	)	(+(-	ABC*DEF^/	Pop from top on stack that is why ^ come 1st
16	*	(+(-*	ABC*DEF^/	
17	G	(+(-*	ABC*DEF^/G	
18	)	(+	ABC*DEF^/G*-	Pop from stack bcz. * come 1st
19	+	(+	ABC*DEF^/G*-	
20	H	(+	ABC*DEF^/G*-H	
21	)	Empty	ABC-DEF^/G*-H*+	End



Page No.
Date : / /

c) 44-15/19-3+2)*12

SN	Scanned	Stack	Postfix	description
1				start
2		(		
3	44	(	44	
4	-	(-	44	
5	15	(-	4415	
6	1	(-1	4415	
7	(	(-1(	4415	
8	9	(-1(	441599	
9	-	1-1(-	4415 99	
10	3	(-1(-	441599 3	
11	*	(-1(-*	44159 93	
12	2	(-1(-*	4415993 2	
13	)	(=1(-*	44159932*-	
14	+	(-(-+	4415932*-	
15	12	(-+	4415932*-1-12	
16	).		4415932*-1-12+	end

Page No.
Date : / /

Ex: 14-(6-10)-10

S.no.	Scanned	Stack	Postfix	Description
1		(		Start
2	14	(	14	
3	-	(-	14	
4	(	(-(	14	
5	6	(-(	146	
6	-	(-(-	146	
7	10	(-(-	14610	
8	)	(-	14610-	
9	-	(-	14610--	
10	10	(-	14610--10	
11	)		14610--10-	
12			14610--10-	End.

Q -> A/B^C-D

	Scanned	stack	Postfix	Description
1	A		A	Start
2	/	/		
3	B	/	AB	
4	^	/^	AB	
5	C	/^	ABC^/	
6	-	-	ABC^/	
7	-	-	ABC^/	
8	D	-	ABC^/D	
			ABC^/D-	End.