

Homework – Notes 8-10

(Solution)

Using Version 2

1. Input String :

E A S Y Q U E S T I O N

QuickSort :-

E A S Y Q U E S T I O N

E A I E **N** U Y S T S O Q

E A **E** **I**

A E

E

O **Q** S T S U Y

O

S T S U **Y**

S T S **U**

S **S** T

S

T

The Sorting Result is :-

A E E I N O Q S S T U Y

Using Version 1

2. Input String :

E A S Y Q U E S T I O N

QuickSort :-

E A S Y Q U E S T I O N

E A E I N U S S T Y O Q

E A E I

E A E

A E

E

U S S T Y O Q

O Q S T Y U S

O

S T Y U S

S S Y U T

S

Y U I

T U Y

U Y

U Y

The Sorting Result is : -

A E E I N O Q S S T U Y

7.6 Example of 6 files in the case where Quicksort gives worst case performance

<u>Ex -1</u>	9	8	7	6	5	4	3	2	1	10
<u>Ex -2</u>	8	7	6	5	4	3	2	1	9	10
<u>Ex -3</u>	7	6	5	4	3	2	1	8	9	10
<u>Ex -4</u>	6	5	4	3	2	1	7	8	9	10
<u>Ex -5</u>	5	4	3	2	1	6	7	8	9	10
<u>Ex -6</u>	4	3	2	1	5	6	7	8	9	10

Note: (Quicksort shows worst case behavior when the partitioning element resides on the extreme sides of the list.) **Answer.**

6.74	0	1	2	3	4	5	6	7	8	9	10
	A	B	R	A	C	A	D	A	B	R	A

		A		B		C		D		R	
A->5		0		5		2		1		1	
B->2											
C->1											
D->1		0		5		7		8		9	
R->2											

	0	1	2	3	4	5	6	7	8	9	10
											(Location)

A	A										
B	A					B					
R	A					B				R	
A	A	A				B				R	
C	A	A						C		R	
A	A	A	A			B		C		R	
D	A	A	A						D	R	
A	A	A	A	A		B		C	D	R	
B	A	A	A	A		B	B	C	D	R	
R	A	A	A	A		B	B	C	D	R	R
A	A	A	A	A	A	B	B	C	D	R	R

(Sorted)

10.41**LSD Radix Sort****Keywords**

Now
is
The
Time
For
All
Good
People
To
Come
The
Aid
Of
Their
Party

1st Leading Place

All
Aid
Come
For
Good
Is
Now
Of
People
Party
The
Time
To
The
Their

2nd Leading Place

Party
People
Of
The
The
their
Aid
Time
All
Now
For
Good
To
Come
Is

(Answer)**3.25**

Returns no. of nodes in a circular linked list

Method:

```
int count-node (Node p) /* p is a node type variable having two fields, val & next
*/
```

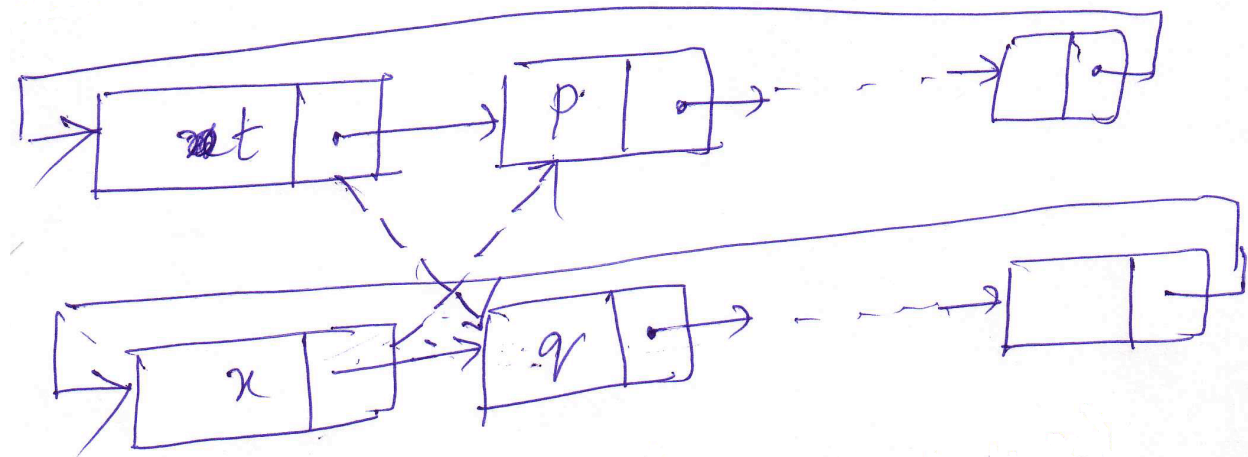
```
{
    int count=0;
    Node start=p;
    While (p.next != start)
    {
        count ++;
        p = p.next;
    }
    return count;
}
```

3.26

Code fragment that determines the number of nodes that are between two given reference x & t

```
int count-node (Node x, Node t)
{
    int count=0;
    While (x.next != t)
    {
        count ++;
        x = x.next;
    }
    return count;
}
```

3.27



```
void list-ins(Node t, Node x)
{
    Node q = x.next;
    Node p = t.next;
    x.next = p;
    t.next = x;
}
```

This entire insertion takes $O(1)$ time.

4.5

E

A

S

pop-1 x

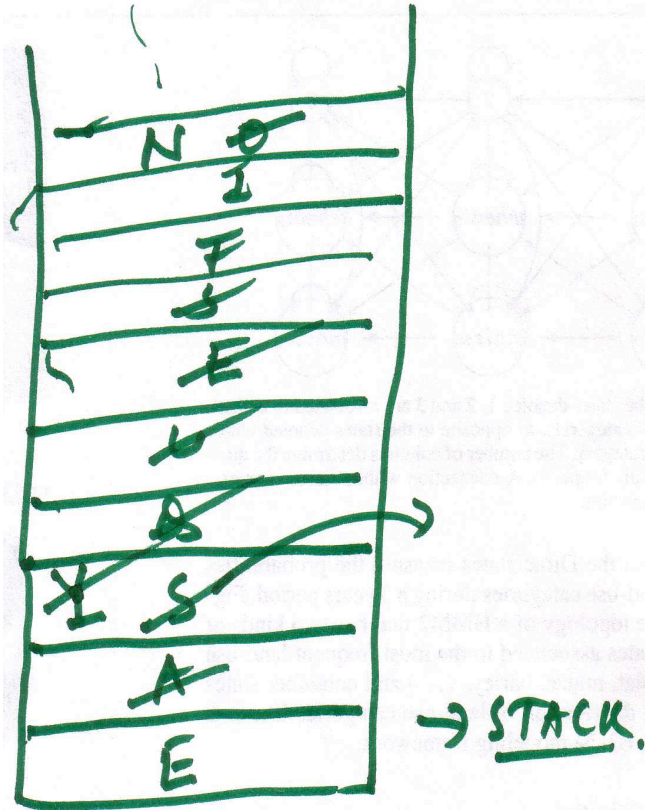
Y

(2) x

Q

U

E



(3) x

(4) x

(5) x

S

T

(6) x

(7) x

(8) x

I

O

(9) x

N

(10) x

(11) x

(12) x

Pop (1) ->

(2)

(3)

(4)

(5)

(6)

(7)

(8)

(9)

(10)

(11)

(12)

Returns ->

S

Y

E

U

Q

T

S

A

O

N

I

E

4.18

Array Representation

```
Class intStack
{
    private int[] S;
    private int[] N;

    intStack(int maxN)
    {
        S=new int [maxN]; N=0;
    }
    int count() {
        return N;
    }
    void push(int item) {
        S[N++] = item;
    }
    int pop() {
        return S[--N];
    }
}
```

(Answer)

Linked List Representation

```
Class intStack
{
    private Node head;
    private class Node {
        - - - -
        - - - -
    }
    intStack(int maxN)
    - - - -
    int count() /* Isempty() is replaced by count method */
    {
        private int count=0;
        while (head.next != null) {
            head = head.next;
            count++;
        }
    }
}
```

```

    }
    return count;
}
void push(int item) {
-   -   -
}
int pop() {
-   -   -
}
}

```

(Answer)

4.9

Convert to Postfix Expression

$$(5 * ((9 * 8) + (7 * (4 + 6))))$$

Input	Output	Stack
(5	
5		
*		*
(*
(*
9	9	*
*		* *
8	8	*
)	*	*
+		* +
(* +
7	7	
*		* + *
(* + *
4	4	
+		* + * +
6	6	* + * +
)	+	
)	*	
)	+	
)	*	

The postfix Expression is : - 5 9 8 * 7 4 6 + * + *

4.10

Postfix Evaluation (showing stack content)

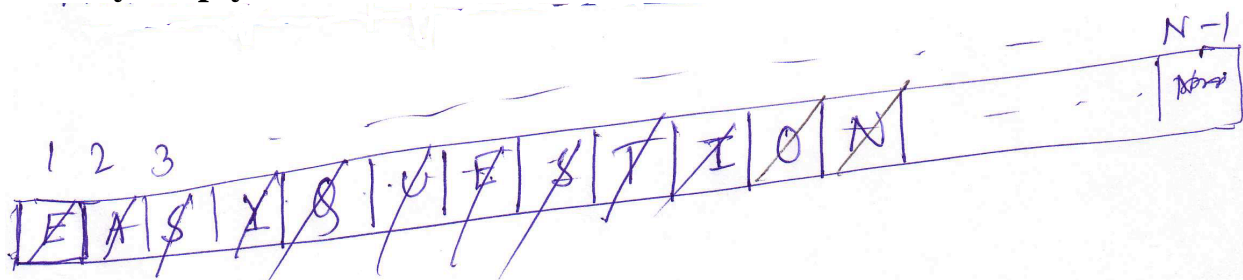
Token	Stack
5	5
9	<u>5</u> 9
*	45
8	45 8
7	45 8 7
4	45 8 7 4
6	45 8 7 4 <u>6</u>
+	45 8 <u>7</u> <u>10</u>
*	45 8 70
2	45 8 70 2
1	45 8 70 2 1
3	45 8 70 2 <u>1</u> <u>3</u>
*	45 8 70 <u>2</u> <u>3</u>
+	45 8 <u>70</u> <u>5</u>
*	45 <u>8</u> <u>350</u>
+	<u>45</u> <u>358</u>
*	16110

(Answer)

4.36

E A S * Y * Q U E * * *
 S T * * * I O * N * * *

Initially Empty Queue



1 st	get (*)	->	Returns	E
2 nd	get (*)	->	Returns	A
3 rd	get (*)	->	Returns	S
4 th	get (*)	->	Returns	Y
5 th	get (*)	->	Returns	Q
6 th	get (*)	->	Returns	U
7 th	get (*)	->	Returns	E
8 th	get (*)	->	Returns	S
9 th	get (*)	->	Returns	T
10 th	get (*)	->	Returns	I
11 th	get (*)	->	Returns	O
12 th	get (*)	->	Returns	N

(Answer)
