2019/2020

Part B

a) i) sequential (linear) searching.

ii) O(n) . 5 number of comparison

iii) slowest searching algorithm because it use unsorted array .

iv) using improved sequential searching

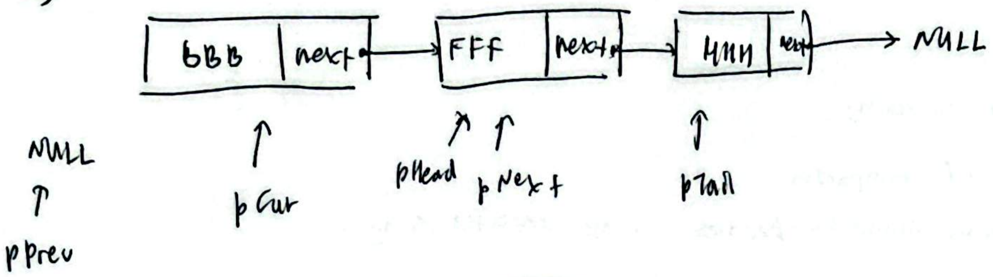
```
if (search_key) == array [p])
        break;
```

b)

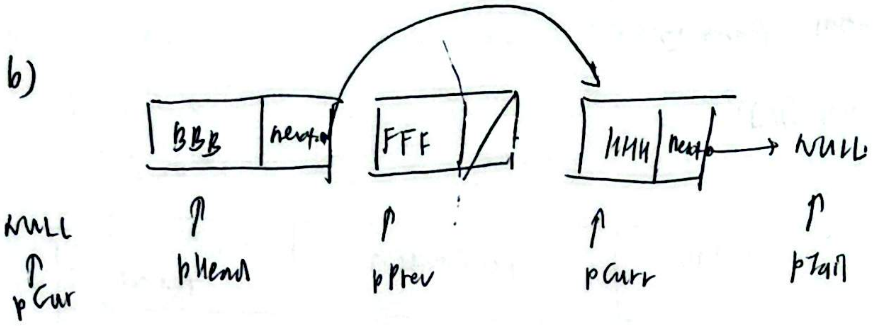
	left	right	middle	input [middle]	found
i)	0	11	5	40	Yes
ii)	0	11	5	40	NO
	6	11	8	66	NO
	9	11	10	88	NO
	11	11	11	100	Yes
iii)	0	11	5	40	NO
	0	4	2	19	NO
	0	1	0	5	NO
	1	1	1	9	NO
	2	1			NO

Question 2

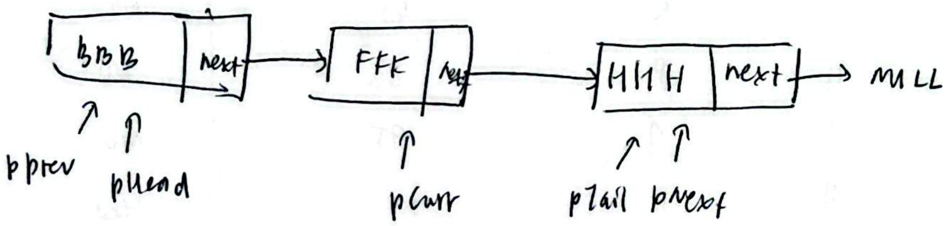
a)



b)



c)



A)

```
Node *NNode1 = new Node ;
 NNode1 -> data = "GGG" ;
 NNode1 -> next = NULL;

pHead -> next -> next = NNode1 ;
NNode1 -> next = pTail ;
```

e)

```
Node *NNode2 = new Node ;
NNode2 -> data = "AAA" ;
NNode2 -> next = NULL;

 NNode2 -> next = pHead ;
 pHead = NNode ;
```

f)

```
Node *temp = pHead -> next
pHead -> next = pTail ;
  temp -> next = NULL;
delete temp;
```

Question 3

a) $x \overset{\textcircled{4}}{-} y \overset{\textcircled{2}}{*} 2 \overset{\textcircled{3}}{/} (p \overset{\textcircled{1}}{-} r) \overset{\textcircled{5}}{+} s$

b) + - x / * y 2 - p r s

c) x y 2 * p r - / - s + .

d)

post fix	ch	op	operand 1	operand 2	Result	stack
3 9 % 20 * 7 -						
9 % 20 * 7 -	3					3
% 20 * 7 -	9					3 9
20 * 7 -	%	%	9	3	3	3
* 7 -	20					3 20
7 -	*	*	20	3	60	60
-	7					60 7
	-	-	7	60		53

Question 4

a) i) Queue 1 use queue linear linked list.
Queue 2 use queue circular linked list.

ii)
```
enQueue ( int d) {
                    // Queue 1
    Node * newNode = newNode( d) ;
    if (!isEmpty) {
    newNode -> next = NULL ;
    rear -> next = newNode;
    rear = newNode; }
    else {
        newNode -> next = NULL;
        front = rear = newNode ; } }

//Queue 2

    isEmpty()  ->  newNode -> next = newNode;
                   rear = newNode ;

    !isEmpty() ->  newNode -> next = rear -> next
                   rear -> next = newNode;
                   rear = newNode;
```

Question 4

a) i) Queue 1 is queue linear while Queue 2 is queue circular implementation

ii) Queue1

```
rear→next = newNode;
  rear = newNode
```

Queue 2

```
newNode→next = rear→next;
rear→next = newNode
rear = newNode;
```

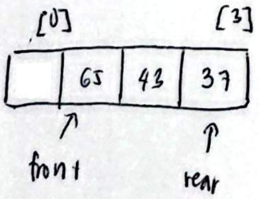
iii) Queue 1

```
Node *temp = front;
front = front→next;
delete temp;
```

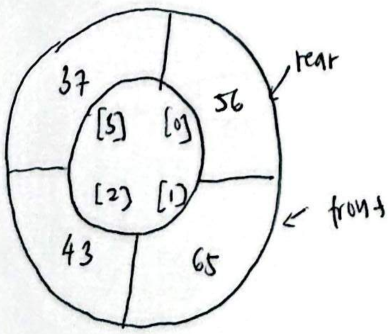
Queue 2

```
Node *temp = rear→next ;
rear→next = temp→next;
temp→next = NULL;
delete temp;
```

b) a)



b)

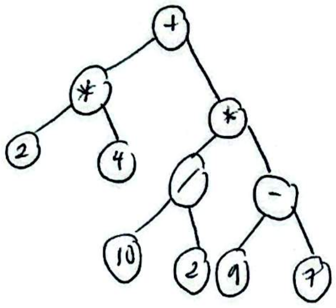


Queue (b) (queue circular array) because no wasted memory since

CamScanner

Question 5

a) $2 \overset{②}{*} 4 \overset{⑤}{+} 10 \overset{③}{/} 2 \overset{④}{*} \overset{①}{(} 9 - 7)$

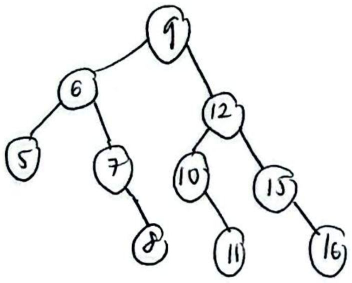


b)

prefix : +, +, 5, *, 4, 2, −, 2, 2

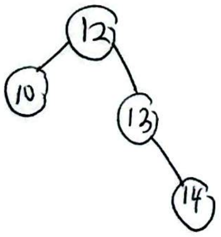
postfix : 5, 4, 2, *, +, 2, 2, −, +

c)

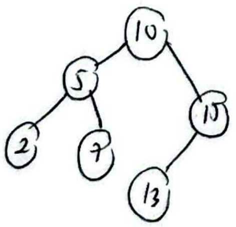


d) level 2 .

e)



f)



CamScanner