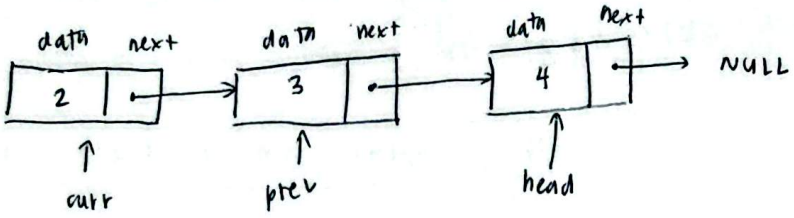
Part B (2018/2019)

Question 1

a) i) Output:

4 → 3 → 2 → NULL

ii)



iii)

4 → NULL

b) i)

```
w4 -> prev = w3;
w3 -> prev = w2;
w2 -> prev = w1;
w1 -> prev = w4;
```

ii)

```
Word *w = w1;
while (w != w4) {
    cout << w -> phrase << " ";
    w = w -> next; }

cout << w -> phrase << " " << endl;

Word *pw = w4;
while (pw != w1) {
    cout << pw -> phrase << " ";
    pw = pw -> prev; }

cout << pw -> phrase << " " << endl;
```

Question 2

a) ⑤ ⑥ ① ④ ③ ② ⑦

A * B + (C % D) / (8 + R / 5) - 2

prefix: - + * A B / % C D + 8 / R 5 2

postfix: A B * C D % 8 R 5 / + / + 2 -

b)

post fix	ch	op	oprn1	oprn2	result	stack
25 8 3 - / 6 * 10 %						
8 3 - / 6 * 10 %	25					25
3 - / 6 * 10 %	8					25 8
- / 6 * 10 %	3					25 8 3
/ 6 * 10 %	-	-	3	8		25 5
6 * 10 %	/	/	5	25		5
* 10 %	6					5 6
10 %	*	*	6	5		30
%	10					30 10
	%	%	10	30		0

c)

num

13
6
3
1
0

stack S

[4]
[0]

1
1
0
1

d) decimal number: 13 to binary: 1 1 0 1

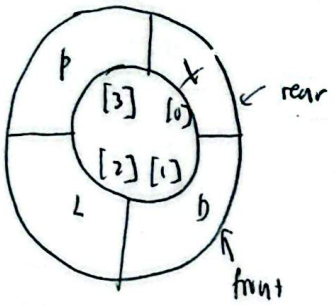
Question 3

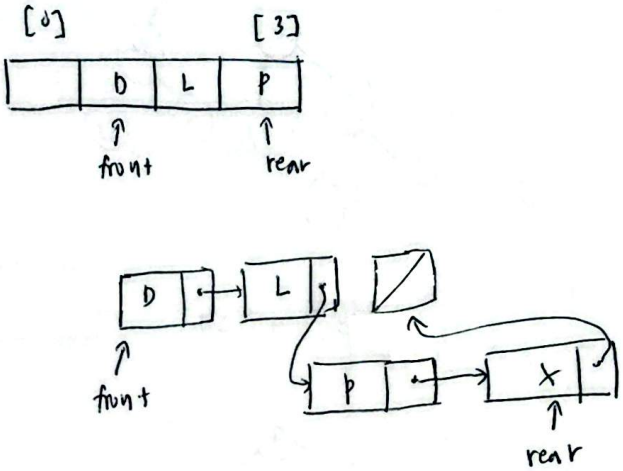
a) i) In linear-array, front = 0, rear = -1 while in circular-array, count = 0, front = 0
rear = SIZE - 1.

In linear-array, there are only front and rear, while in circular-array, there are addition of count.

ii) In linear-array, rear = rear + 1 used to calculate rear, while in circular-array, rear = (rear + 1) % SIZE used to calculate rear.

b)





c) at front

Queue using linear-array. It wasted space, when dequeue, after that enqueue, no data can be inserted if rear already at the end of the list because queue already full.

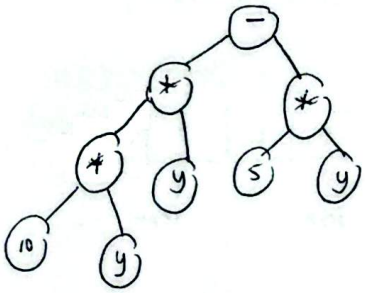
(d) use circular array since it will utilize all empty space.

Question 4

a) i) A, D, F, J, L, M

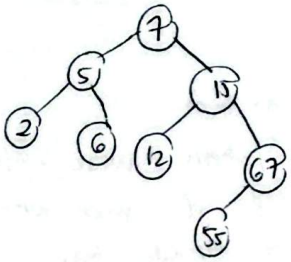
ii) J, D, A, F, L, M

b) i) $10y^2 - 5y$ → ① ② ④ ③ 10 * y * y - 5 * y



ii) 10y*y*y*5y*-

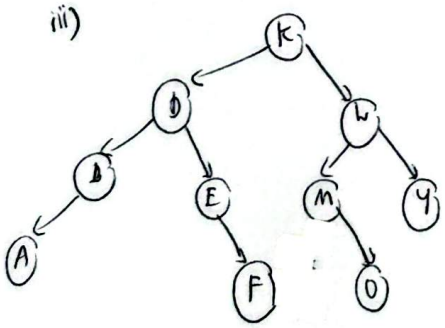
c)



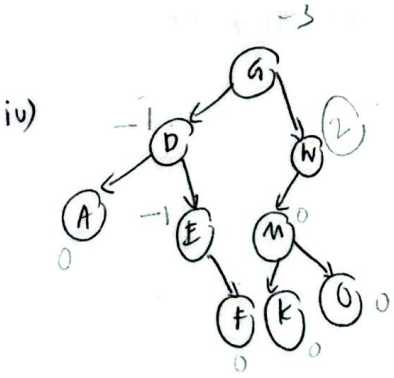
d) i) 4

ii) 3

iii)



iv)



v) Not full, not complete, not balanced