1-1

```
Ispalindrome(string text)
Create stack st
for i<-0 to i<-text size
    push text[i] in st
Create bool yes <- true
for i<-0 to i<-text size
    if( text[i] isn't same st top )
        yes <- false
        break
    pop st
if( yes is true )
    print "회문입니다."
else
    print "회문이 아닙니다."
```

1-3

```
Ispalindrome(string text)
Create deque dq
for i<-0 to i<-text size
    push_back text[i] in dq
Create bool yes <- true
for i<-0 to i<-text size/2
    if( dq.back isn’t same dq.front )
        yes<-false
        break
    pop_back dq
    pop_front dq
if( yes is true)
    print “회문입니다.”
else
    print “회문이 아닙니다.”
```

2-1

```
HotPotato(player_list, num)
Create int playersNum
Create CircleQueue players
Initialize players
for i<-0 to i<-player_list size
    Enqueue(players, player_list[i])
playersNum <- players.Max
while( playerNum >1 )
    Create int repeatNum <- num
    while( repeatNum isn’t 0)
        Create string getPotato_player <- Dequeue(players)
        Enqueue(players, getPotato_player)
        repeatNum—
    print Dequeue(players)   “ is out”
    playersNum—
return Dequeue(players)
```

3-a-1

V0 : {V1, V4}

V1 : {V0, V2, V3}

V2 : {V1, V4}

V3 : {V1, V4}

V4 : {V0, V2, V3}

3-a-2

	V0	V1	V2	V3	V4
V0	0	1	0	0	1
V1	1	0	1	1	0
V2	0	1	0	0	1
V3	0	1	0	0	1
V4	1	0	1	1	0

3-b-1

0 : {1, 2, 3}

1 : {2, 3}

2 : {4}

3 : {0, 4}

4 : {1, 2}

5 : {4}

3-b-2

	0	1	2	3	4	5
0	0	1	1	1	0	0
1	0	0	1	1	0	0
2	0	0	0	0	1	0
3	1	0	0	0	1	0
4	0	1	1	0	0	0
5	0	0	0	0	1	0

3-b-3

진입차수 가장 낮은 정점 : 5 -> 차수 = 0

진입차수 가장 큰 정점 : 2, 4 -> 차수 = 3

진출차수 가장 낮은 정점 : 2, 5 -> 차수 = 1

진출차수 가장 큰 정점 : 0 -> 차수 = 3