Lab7 Linked Stack & Queue 4/19 (월)

1. 문제: **Linked Stack** 과 **Linked Queue**에 각각 데이터를 입력하고, 이 데이터들을 Merge하여 **Linked List**로 만들어 출력하는 프로그램을 구현 하시오

2. 데이터 구성

```
class Node {
private:
        int data;       Node* next;
        Node(int value) {
          data = value; next = 0; }
        friend class linkedStack;
        friend class linkedQueue;
        friend class linkedList;
};

class linkedStack {
private:Node* top;
public:
        linkedStack() { top = 0; }
        ~linkedStack() {};
        void createStack();
        void push(int);
        char pop();
        int isEmptys();
        void displayStack();
};
```

```
class linkedQueue {
private:  Node* front;   Node* rear;
public:
    linkedQueue() { front = 0; rear = 0; }
    ~linkedQueue() {};
    void createQueue();
    void enqueue(int);
    char dequeue();
    int  isEmptyq();
    void displayQueue();
};

class linkedList {
private:        Node* head;
public:
        linkedList() { head = 0; }
        ~linkedList() { };
        void insert(int);
        int is_empty();
        void display();
};
```

3. Menu: 1. PUSH, 2.POP, 3. ENQUEUE, 4.DEQUE, 5. MakeList. 6. Quit

MakeList: { while (!stack-empty) pop(data1); insert(data1)
while(!queue-empty) deque(data2); insert(data2)}

4. 수행 순서

1) check Stack empty 2) check Queue empty 3) check List empty

4) enter data into Stack: Push 10, Push 20, Push 30 ➔ 30 20 10
pop ➔ 20 10

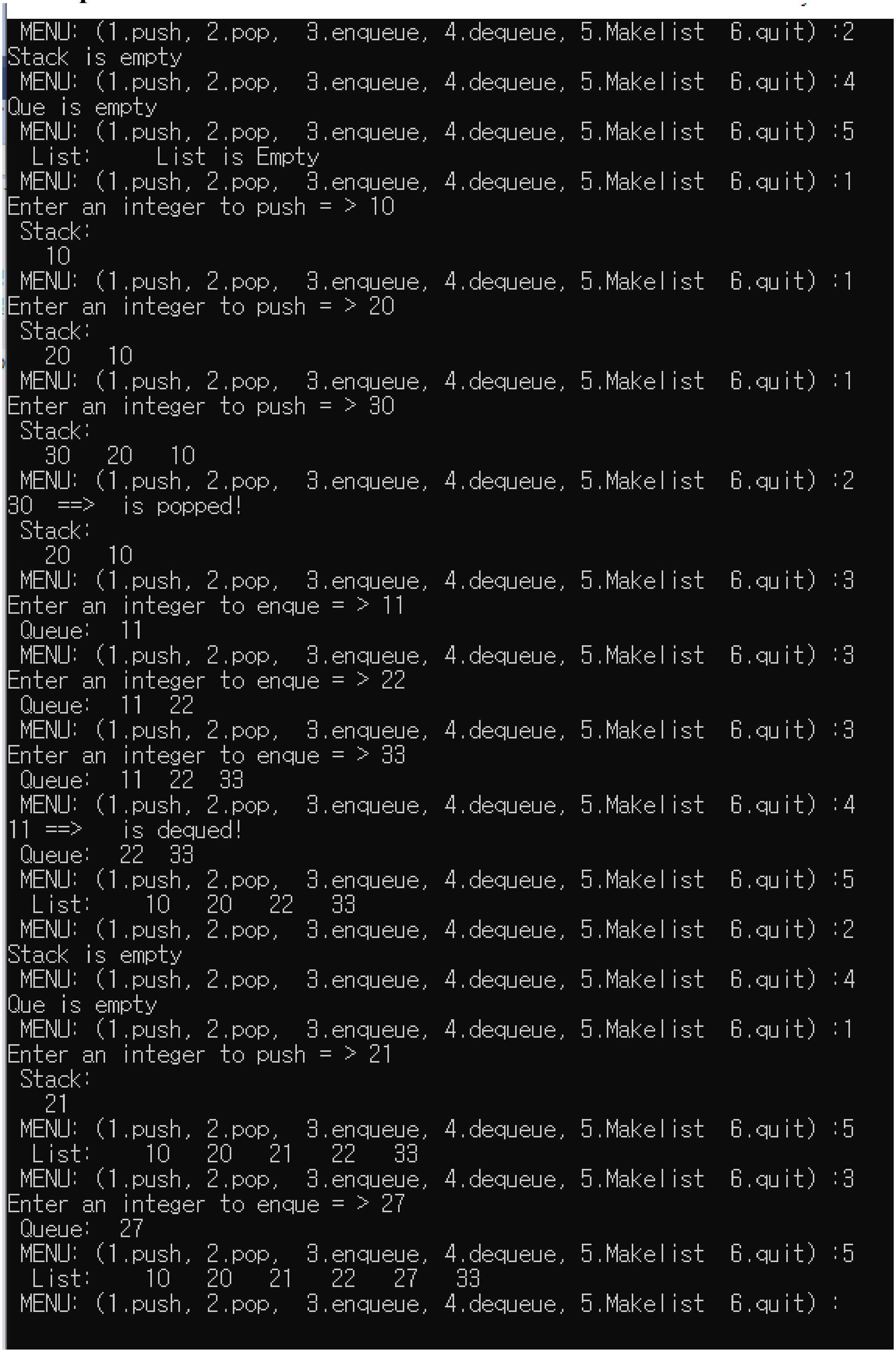
5) enter data into queue enque 11, enque 22, enque 33 ➔ 11 22 33
Deque ➔ 22 33

6) MakeList: ➔ 10 20 22 33

7) Push 21 ➔ 21 8) Makelist ➔ 10 20 21 22 33

9) Enque 44 ➔ 27 10) Makelist ➔ 10 20 21 22 27 33

5. Output



```
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :2
Stack is empty
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :4
Que is empty
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :5
  List:     List is Empty
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :1
Enter an integer to push = > 10
 Stack:
   10
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :1
Enter an integer to push = > 20
 Stack:
   20   10
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :1
Enter an integer to push = > 30
 Stack:
   30   20   10
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :2
30  ==>  is popped!
 Stack:
   20   10
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :3
Enter an integer to enque = > 11
 Queue:  11
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :3
Enter an integer to enque = > 22
 Queue:  11  22
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :3
Enter an integer to enque = > 33
 Queue:  11  22  33
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :4
11 ==>   is dequed!
 Queue:  22  33
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :5
  List:    10   20   22   33
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :2
Stack is empty
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :4
Que is empty
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :1
Enter an integer to push = > 21
 Stack:
   21
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :5
  List:    10   20   21   22   33
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :3
Enter an integer to enque = > 27
 Queue:  27
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :5
  List:    10   20   21   22   27   33
 MENU: (1.push, 2.pop,  3.enqueue, 4.dequeue, 5.Makelist  6.quit) :
```