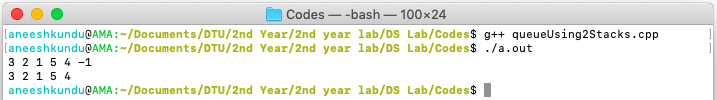
**Experiment 6**

**AIM:** Write a program to implement a queue using 2 stacks.

**THEORY:** A queue is an abstract data structure in which insertion is done at the end and deletion is done at the front. It is FIFO i.e First In First Out. It has 2 pointers-front and rear. Front is for deletion and rear is for insertion. In linked list we do not have to give the size and insertion and deletion is done using pointers.  
In this method, in en-queue operation, the new element is entered at the top of stack1. In de-queue operation, if stack2 is empty then all the elements are moved to stack2 and finally top of stack2 is returned.

**Code:**

1. #include < iostream >
2. #include < stack >
3. using namespace std;
4. struct Queue {
5. stack < int > a, b;
6. void insert(int v) {
7. a.push(v);
8. }
9. int remove() {
10. while (!a.empty()) {
11. b.push(a.top());
12. a.pop();
13. }
14. int ans = -1;
15. if (!b.empty()) {
16. ans = b.top();
17. b.pop();
18. }
19. while (!b.empty()) {
20. a.push(b.top());
21. b.pop();
22. }
23. return ans;
24. }
25. };
26. int main() {
27. Queue q;
28. int temp;
29. cin >> temp;
30. while (temp != -1) {
31. q.insert(temp);
32. cin >> temp;
33. }
34. while (1) {
35. int top = q.remove();
36. if (top == -1) break;
37. cout << top << " ";
38. }
39. cout << endl;
40. return 0;
41. }

**Output:**