Set – A Data Structure Time : 1 hour

Problem - 1

You have to create a stack using linked list.

Input: First line of the input file is an integer n (0 < n < 100) which denotes how many integer number will be push. Next line contains n integers.

Output : print the top value.

|  |  |
| --- | --- |
| Sample input | Sample output |
| 5  7 3 0 5 1 | top = 1 |
| 7  3 1 9 2 6 4 5 | top = 5 |

Problem – 2

Each node have two part. First one is data part and second one is address part. Every node is connected (linked list).

Input: First line of the input file is an integer n (0 < n < 100) which denotes how many integer number will be insert. Second line contains n integers. Third line contains two integer, first one is data and second one is position where you will insert the data.

Output: After insert the data to the 3rd position you have to show the final linked list.

|  |  |
| --- | --- |
| Sample input | Sample output |
| 4  4 6 9 13  5 3 | 13 9 5 6 4 |
| 7  5 4 6 2 9 1 3  10 6 | 3 1 9 2 6 10 4 5 |

Set – B Data Structure Time : 1 hour

Problem - 1

You have to create a stack using linked list.

Input: First line of the input file is an integer n (0 < n < 100) which denotes how many integer number will be push. Next line contains n integers.

Output : pop the value one by one from the stack and print.

|  |  |
| --- | --- |
| Sample input | Sample output |
| 5  7 3 0 5 1 | 1 5 0 3 7 |
| 7  3 1 9 2 6 4 5 | 5 4 6 2 9 1 3 |

Problem – 2

Each node have two part. First one is data part and second one is address part. Every node is connected (linked list).

Input: First line of the input file is an integer n (0 < n < 100) which denotes how many integer number will be insert. Second line contains n integers.

Output: After insert the data to the ending you have to show the final linked list.

|  |  |
| --- | --- |
| Sample input | Sample output |
| 4  13 9 6 4 | 13 9 6 4 |
| 7  3 1 9 2 6 4 5 | 3 1 9 2 6 4 5 |