Insertion of elements in the middle of Doubly Linked List

locked

Write a program that takes as input a sequence of distinct integers, until the character *t* is read. It then iteratively inserts them into the 'middle' of a doubly linked list, as detailed below. Let *j = (i + 1)/2* if *i* is odd and *j = i/2*, otherwise. Then, after the ith insertion, the ith integer should be present in the jth node of the list (assume that the head pointer points to the 1st node of the list). The relative ordering of the other nodes should not change.

Input Format

- Each line contains an integer that is to be inserted into the doubly linked list.
- Character 't' is to 'terminate' the program.

Constraints

- All the elements of the doubly linked list should be distinct integers.
- *i* starts from 1.

Output Format

- Print space separated elements of the doubly linked list

Sample Input 0

```
12
35
50
59
60
73
t
```

Sample Output 0

```
35 59 73 60 50 12
```


Current Buffer (saved locally, editable) C

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code