#include <stdio.h>

#include <stdlib.h>

//Node Structure of XOR List

struct Node

{

int data;

struct Node \*x\_ptr;

};

//XOR Operation on addresses of two nodes

struct Node \* xor(struct Node \*m, struct Node \*n)

{

return (struct Node \*)((unsigned)m ^ (unsigned)n);

}

//Inserting into XOR List from start of list

void insert(struct Node \*\*head, int x)

{

//Create a new node to be inserted

struct Node \*temp = (struct Node \*)malloc(sizeof(struct Node));

temp->data = x;

//As it is inserted at beginning its x\_ptr will (NULL XOR head)

temp->x\_ptr = xor(NULL,(\*head));

//If list is not empty then x\_ptr of head will be (new node XOR next node)

if((\*head) != NULL)

{

//Get address of Next Node

struct Node \*nextNode = xor(NULL,(\*head)->x\_ptr);

//Store XOR of new node and next node

(\*head)->x\_ptr = xor(temp,nextNode);

}

//Make the new node as head

\*head = temp;

}

//Printing XOR List

void printList(struct Node \*head)

{

struct Node \*previous, \*current, \*next;

previous = NULL;

current = head;

while(current)

{

//print data of current node

printf(" %d ",current->data);

//get address of next node as (previous node XOR (previous XOR next node))

//where current->x\_ptr = (previous XOR next node)

next = xor(previous,current->x\_ptr);

//update previous and current for next iteration

previous = current;

current = next;

}

printf("\n");

}

int main()

{

/\*Create doubly linked list as 10->20->30->40->50 \*/

struct Node \*head=NULL;

insert(&head,50);

insert(&head,40);

insert(&head,30);

insert(&head,20);

insert(&head,10);

//Print the XOR list

printList(head);

return 0;

}