

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

struct Node

{

int ele;

struct Node \*next;

};

struct Node \*hptr=NULL,\*tptr;

void createList(int data)

{

struct Node \*nptr;

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

nptr->ele=data;

if(hptr==NULL)

hptr=tptr=nptr;

else

tptr->next=nptr;

tptr=nptr;

nptr->next=NULL;

}

void display()

{

struct Node \*t;

for(t=hptr;t!=NULL;t=t->next)

printf(" %d ",t->ele);

}

int findLoop()

{

struct Node \*slow = hptr, \*fast = hptr;

while (slow && fast && fast->next) {

printf(" %d ", slow->ele);

slow = slow->next;

fast = fast->next->next;

if (slow == fast)

return 1;

}

return 0;

}

int main()

{

createList(10);

createList(20);

createList(30);

createList(40);

createList(50);

display();

hptr->next->next->next->next->next = hptr->next->next;

printf("\n");

printf(" %d ",findLoop());

return 0;

}

1. Fast and Slow Pointer Approach
2. Using the Flag Approach
3. Using the HashMap