Program for Check Linked List is Palindrome or not:

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

struct node

{

int data;

struct node \*next;

};

struct node \*getnode(int data)

{

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

temp->data=data;

temp->next=NULL;

return temp;

}

void print(struct node \*head)

{

struct node \*current=head;

while(current!=NULL)

{

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

current=current->next;

}

}

struct node \*hasCycle(struct node \*head) {

struct node\* slow=head;

struct node \*fast=head;

while(slow!=NULL && fast!=NULL && fast->next!=NULL)

{

slow=slow->next;

fast=fast->next->next;

}

return slow;

}

struct node \*reverse(struct node \*head2)

{

struct node \*current=head2;

struct node \*prev=NULL;

struct node \*next1=NULL;

while(current!=NULL)

{

next1=current->next;

current->next=prev;

prev=current;

current=next1;

}

return prev;

}

bool isPalin(struct node \*head1, struct node \*temp2)

{

while(temp2 != NULL && head1!=NULL)

{

if(temp2->data != head1->data)

return false;

head1 = head1->next;

temp2 = temp2->next;

}

return true;

}

int main() {

struct node \*head1=getnode(1);

head1->next=getnode(2);

head1->next->next=getnode(3);

head1->next->next->next=getnode(9);

head1->next->next->next->next=getnode(1);

//print(head);

struct node \*temp1=hasCycle(head1);

struct node \*head2=temp1->next;

temp1->next=NULL;

struct node \*temp2=reverse(head2);

if(isPalin(head1,temp2))

printf("palindrome");

else

printf("not palindrome");

return 0;

}