**KTH ELEMENT FROM LAST IN LINKED LIST**

Given a linked list with n nodes. Find the kth element from last without computing the length of the linked list.

**Input Format:**

First line contains space separated integers representing the node values of the linked list. The list ends when the input comes as '-1'. The next line contains a single integer k.

**Constraints:**

n < 10^5

**Output Format**

Output a single line containing the node value at the kth element from last.

**Sample Input**

1 2 3 4 5 6 -1

3

**Sample Output**

4

#include<iostream>

using namespace std;

class Node{

public:

int data;

Node \*next;

Node(int d)

{

data=d;

next=NULL;

}

};

void insertAtEnd(Node \*&head,int num)

{

if(head==NULL)

{

head=new Node(num);

return;

}

Node \*tail=head;

while(tail->next!=NULL)

tail=tail->next;

Node \*n=new Node(num);

tail->next=n;

return;

}

void reverse(Node \*&head)

{

Node \*c=head;

Node \*p=NULL;

Node \*n;

while(c!=NULL)

{

n=c->next;

c->next=p;

p=c;

c=n;

}

head=p;

}

int findelement(Node \*head,int k)

{

int x=0;

while(head!=NULL)

{

if(x==k)

return head->data;

x++;

head=head->next;

}

return -1;

}

/\*void print(Node \*head)

{

while(head!=NULL)

{

cout<<head->data<<" ";

head=head->next;

}

cout<<endl;

}\*/

int main()

{

Node \*head=NULL;

int k,i,num;

cin>>num;

insertAtEnd(head,num);

while(num!=-1)

{

cin>>num;

insertAtEnd(head,num);

}

cin>>k;

//print(head);

reverse(head);

cout<<findelement(head,k);

}