#include<iostream>

using namespace std;

struct node

{

int info;

struct node \*next;

}

\*last;

class massSuicide

{

public:

void create\_node(int value);

void killPerson(int value);

void display\_list();

massSuicide()

{

last = NULL;

}

};

void massSuicide::create\_node(int value)

{

struct node \*temp1;

temp1 = new(struct node);

temp1->info = value;

if (last == NULL)

{

last = temp1;

temp1->next = last;

}

else

{

temp1->next = last->next;

last->next = temp1;

last = temp1;

}

}

void massSuicide::killPerson(int position)

{

node \*headerNode;

headerNode= last->next;

// If list is empty

if ((headerNode) == NULL)

return;

struct node\* temp = headerNode;

//traversing through list to find next person to die

for (int i=0; temp!=NULL && i<position-1; i++)

{

temp = temp->next;

}

last=temp; //saving from where the next person count should start

cout<<"killing: "<<(temp->next)->info<<"\n";

if (temp == NULL || temp->next == NULL)

return;

// Node temp->next gets killed so changing pointer to the next of node in list

struct node \*next = temp->next->next;

temp->next = next;

}

int main()

{

int noOfPeople, killFrom;

massSuicide mList;

cout<<"Enter number of people:\n";

cin>>noOfPeople;

cout<<"Enter position of first person to die:\n";

cin>>killFrom;

//creating the circle of people starting with position 1

for(int i=1;i<noOfPeople+1;i++)

{

mList.create\_node(i);

}

//starting to kill people

for(int i=0;i<noOfPeople;i++){

mList.killPerson(killFrom-1);

}

cout<<"All People are dead";

cout<<endl;

return 0;

}