#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#define OVERFLOW 0

#define OK 1

#define ERROR 0

//栈的顺序存储表示

#define STACK\_INIT\_SIZE 100 //存储空间初始分配量

#define STACKINCREMENT 10 //存储空间分配增量

typedef struct

{

int \*base; //在栈构造之前和销毁之后,base的值为NULL

int \*top; //栈顶指针

int stacksize; //当前已分配的存储空间，以元素为单位

}SqStack;

int InitStack(SqStack &S)

{ //构造一个空栈S

S.base=(int\*)malloc(STACK\_INIT\_SIZE\*sizeof(int));

if(!S.base)

exit(OVERFLOW);//存储分配失败

S.top=S.base;

S.stacksize=STACK\_INIT\_SIZE;

return OK;

}//InitStack

int GetTop(SqStack S,int &e)

{

//若栈不空，则用e返回s的栈顶元素，并返回OK;否则返回ERROR if(S.top==S.base)

return ERROR;

e=\*(S.top-1);

return OK;

}//GetTop

int Push(SqStack &S, int e)

{

//插入元素e为新的栈顶元素

if(S.top-S.base >= S.stacksize)

{

//栈满，追加存储空间

S.base=(int\*)realloc(S.base,

(S.stacksize+STACKINCREMENT)\*sizeof (int));

if(!S.base)

exit(OVERFLOW);//存储分配失败

S.top=S.base+S.stacksize;

S.stacksize+=STACKINCREMENT;

}

\*S.top++=e;

return OK;

}//Push

int Pop(SqStack &S,int &e)

{

//若栈不空，则删除s的栈顶元素,用e返回其值，

//并返回OK;否则返回ERROR

if(S.top==S.base)

return ERROR;

e=\*--S.top;

return OK;

}//Pop

void conversion(int N,int d)

{

//N:表示十进制正整数,d表示2到16之间的某种进制

//下面实现将十进制整数N转换成某种d进制数,并输出显示。 SqStack S;

int e;

InitStack(S);//构造空栈

while(N)

{

Push(S, N % d);

N = N / d;

}

while(S.top!=S.base)

{

Pop(S,e);

if(e<10)

printf("%d",e);

else

printf("%c",e+55);

}

printf("\n");

}

void main()

{

int N,d;

printf("请输入一个十进制数:");

scanf("%d",&N);

printf("请输入要转换的进制数:");

scanf("%d",&d);

conversion(N,d);

}