#include<stdio.h>

#include<stdlib.h>

struct stack {

int \*data;

int capacity;

int top;

};

int isFull(const struct stack \*ps)

{

return ps->top==ps->capacity;

}

int isEmpty(const struct stack \*ps)

{

return ps->top==0;

}

int push(struct stack \*ps,int x)

{

if(isFull(ps)) return 0;

else

{

ps->data[ps->top++]=x;

return 1;

}

}

int pop(struct stack \*ps,int \*px)

{

if(isEmpty(ps)) return 0;

else

{

\*px=ps->data[--(ps->top)];

return 1;

}

}

int top(const struct stack \*ps,int \*px)

{

if(isEmpty(ps)) return 0;

else

{

\*px=ps->data[ps->top-1];

return 1;

}

}

void destroy(struct stack \*ps)

{

free(ps->data);

}

void init(struct stack \*ps,int capacity)

{

ps->capacity=capacity;

ps->data=(int\*)malloc(sizeof(int)\*capacity));

ps->top=0;

}

int main()

{

struct stack st;

init(&st,5);

push(&st,11);

int x;

pop(&st,&x);

printf("%d\n",x);

top(&st,&x);

printf("%d\n",x);

destroy(&st);

return 0;

}