**typedef struct {**

**int\* stack1;**

**int top;**

**} MyQueue;**

**MyQueue\* myQueueCreate() {**

**MyQueue\* myqueue=malloc(sizeof(MyQueue));**

**(\*myqueue).stack1=malloc(sizeof(int));**

**(\*myqueue).top=-1;**

**return(myqueue);**

**}**

**void myQueuePush(MyQueue\* obj, int x) {**

**(\*obj).top=(\*obj).top+1;**

**(\*obj).stack1=realloc((\*obj).stack1,sizeof(int)\*((\*obj).top+1));**

**\*((\*obj).stack1+(\*obj).top)=x;**

**}**

**int myQueuePop(MyQueue\* obj) {**

**MyQueue\* temporary\_queue=myQueueCreate();**

**(\*temporary\_queue).stack1=realloc((\*temporary\_queue).stack1,((\*obj).top+1)\*sizeof(int));**

**int i;**

**for(i=(\*obj).top;i>=0;i--){**

**//\*((\*temporary\_queue).stack1+(\*temporary\_queue).top++)=\*((\*obj).stack1+(\*obj).top--);**

**myQueuePush(temporary\_queue, \*((\*obj).stack1+(\*obj).top--));**

**}**

**int element;**

**element=\*((\*temporary\_queue).stack1+(\*temporary\_queue).top);**

**(\*temporary\_queue).top-=1;**

**for(i=(\*temporary\_queue).top;i>=0;i--){**

**//\*((\*temporary\_queue).stack1+(\*temporary\_queue).top++)=\*((\*obj).stack1+(\*obj).top--);**

**myQueuePush(obj, \*((\*temporary\_queue).stack1+(\*temporary\_queue).top--));**

**}**

**return(element);**

**}**

**int myQueuePeek(MyQueue\* obj) {**

**return(\*((\*obj).stack1));**

**}**

**bool myQueueEmpty(MyQueue\* obj) {**

**if((\*obj).top==-1){**

**return(true);**

**}**

**else{**

**return(false);**

**}**

**}**

**void myQueueFree(MyQueue\* obj) {**

**free(obj);**

**}**

**/\*\***

**\* Your MyQueue struct will be instantiated and called as such:**

**\* MyQueue\* obj = myQueueCreate();**

**\* myQueuePush(obj, x);**

**\* int param\_2 = myQueuePop(obj);**

**\* int param\_3 = myQueuePeek(obj);**

**\* bool param\_4 = myQueueEmpty(obj);**

**\* myQueueFree(obj);**

**\*/**

