

```
#define QUEUE_SIZE_UNIT 500

typedef struct queueInfo
{
    int size;
    int cur;
    int front;
    int rear;
    int *q;
} QUEUE;

QUEUE* createQueue(int size)
{
    QUEUE *obj = malloc(sizeof(QUEUE));
    obj->size = size;
    obj->cur = 0;
    obj->front = 0;
    obj->rear = -1;
    obj->q = malloc(sizeof(int)*size);
    return obj;
}

void destroyQueue(QUEUE *obj)
{
    free(obj->q);
    free(obj);
}

bool isEmpty(QUEUE *obj)
{
    return (obj->cur == 0 ? true : false) ;
}

bool isFull(QUEUE *obj)
{
    return (obj->cur == obj->size ? true : false) ;
}

void addQueue(QUEUE *obj, int idx)
{
    if (isFull(obj))
```

```

{
return;
}

obj->rear = (obj->rear+1) % obj->size;
obj->q[obj->rear] = idx;
obj->cur++;
}

int delQueue(QUEUE *obj)
{
if (isEmpty(obj))
{
return 0;
}
int ret = obj->q[obj->front];
obj->front = (obj->front+1) % obj->size;
obj->cur--;
return ret;
}

typedef struct
{
int size;
int cur;
QUEUE *q1;
QUEUE *q2;
} MyStack;

/** Initialize your data structure here. */

MyStack* myStackCreate()
{
MyStack *obj = malloc(sizeof(MyStack));
obj->q1 = createQueue(QUEUE_SIZE_UNIT);
obj->q2 = createQueue(QUEUE_SIZE_UNIT);
return obj;
}

/** Push element x onto stack. */
void myStackPush(MyStack* obj, int x)
{

```

```

if (isFull(obj->q1))
{
obj->q1->size += QUEUE_SIZE_UNIT;
obj->q1->size += QUEUE_SIZE_UNIT;
obj->q1->q = realloc(obj->q1->q, sizeof(int)*(obj->q1->size));
obj->q2->q = realloc(obj->q2->q, sizeof(int)*(obj->q2->size));
}
addQueue(obj->q1, x);
}

/** Removes the element on top of the stack and returns that element. */
int myStackPop(MyStack* obj)
{
int ret;
if (isEmpty(obj->q1))
{
return -1;
}

ret = delQueue(obj->q1);
while (!isEmpty(obj->q1))
{
addQueue(obj->q2, ret);
ret = delQueue(obj->q1);
}
QUEUE *tmp = obj->q1;
obj->q1 = obj->q2;
obj->q2 = tmp;
return ret;
}


/** Get the top element. */
int myStackTop(MyStack* obj)
{
int ret = myStackPop(obj);
addQueue(obj->q1, ret);
return ret;
}

/** Returns whether the stack is empty. */
bool myStackEmpty(MyStack* obj)


```


```
{  
return isEmpty(obj->q1);  
}  
  
void myStackFree(MyStack* obj)  
{  
delQueue(obj->q1);  
delQueue(obj->q2);  
free(obj);  
}
```

Accepted

 Shashank_C_254 submitted at Jan 09, 2024 23:33

 Editorial

 Solution


 Runtime

3 ms

Beats **33.60%** of users with C

 Memory

6.55 MB

 Beats **75.27%** of users with C

