

# 使用说明

高炜良

(北京理工大学数学与统计学院, 北京 100081)

## 目录

目录 . . . . .	I
第 1 章 总述 . . . . .	1
第 2 章 使用示例 . . . . .	1
2.1 内存管理 . . . . .	1
2.2 链表 . . . . .	2
2.3 数组 . . . . .	3
2.4 堆 . . . . .	3
2.5 栈 . . . . .	4
2.6 队列 . . . . .	4
2.7 树 . . . . .	5
2.8 图 . . . . .	6

## 第 1 章 总述

命令基本格式为: `function(arg1, ..., argn)`

注意事项:

- 所有函数名、参数及符号均为英文及数字
- 函数名务必完整, 且需要区分大小写
- 函数名后面的参数要用英文圆括号括起来
- 参数之间用英文逗号隔开
- 参数个数务必与说明一致
- 当参数为数值时, 输入整数即可
- 当参数为地址时, 需要输入标准的十六进制格式地址

## 第 2 章 使用示例

### 2.1 内存管理

```
100.00 MB 内存申请成功! 地址为: 00000000096A040

请输入命令: checkSpace()
已使用空间: 0.00 MB
剩余空间: 100.00 MB
总空间: 100.00 MB

请输入命令: checkBlock(00000000096A080)
00000000096A080 未使用!

请输入命令: checkFreeSpaceList()
空闲块链表:
64.00 MB: 00000000096A040
32.00 MB: 000000000496A040
4.00 MB: 000000000696A040

请输入命令: myMalloc(1024)
1.00 KB 内存申请成功! 地址为: 000000000696A040

请输入命令: myCalloc(2, 1024)
2.00 KB 内存申请成功! 地址为: 0000000006D69440

请输入命令: myFree(000000000696A040)
000000000696A040 释放成功!

请输入命令: myFree(0000000006D69440)
0000000006D69440 释放成功!
```

## 2.2 链表

```

请输入命令: initList()
16.00 B 内存申请成功! 地址为: 0000000006A4B040
链表创建成功! 地址为: 0000000006A4B040

请输入命令: listEmpty(0000000006A4B040)
链表为空!

请输入命令: addListNode(0000000006A4B040, 0, 123)
16.00 B 内存申请成功! 地址为: 0000000006E4B030
成功在链表第0个元素后插入新元素123

请输入命令: addListNode(0000000006A4B040, 1, 456)
16.00 B 内存申请成功! 地址为: 0000000006E4B010
成功在链表第1个元素后插入新元素456

请输入命令: addListNode(0000000006A4B040, 2, 789)
16.00 B 内存申请成功! 地址为: 0000000006E4B020
成功在链表第2个元素后插入新元素789

请输入命令: visitList(0000000006A4B040)
遍历链表:
第1个元素值为: 123, 地址为0000000006E4B030
第2个元素值为: 456, 地址为0000000006E4B010
第3个元素值为: 789, 地址为0000000006E4B020
链表遍历结束!

```

```

请输入命令: visitList(0000000006A4B040)
遍历链表:
第1个元素值为: 123, 地址为0000000006E4B030
第2个元素值为: 456, 地址为0000000006E4B010
第3个元素值为: 789, 地址为0000000006E4B020
链表遍历结束!

请输入命令: listPreNode(0000000006A4B040, 456)
元素的前驱地址为: 0000000006E4B030, 值为: 123

请输入命令: listNextNode(0000000006A4B040, 456)
元素的后继地址为: 0000000006E4B020, 值为: 789

请输入命令: changeListVal(0000000006A4B040, 2, 987)
成功将链表第2个元素更改为987

请输入命令: getListNodeVal(0000000006E4B010)
节点元素为: 987

请输入命令: findListNode(0000000006A4B040, 987)
元素位于链表的第2位

请输入命令: delListNode(0000000006A4B040, 1)
0000000006A4B040 释放成功!
成功删除链表第1个元素

```

## 2.3 数组

```

请输入命令: initArray()
0.00 MB 内存申请成功! 地址为: 0000000006E4B000
数组创建成功! 地址为: 0000000006E4B000

请输入命令: myMalloc(0)
0.00 MB 内存申请成功! 地址为: 0000000006E4B000

请输入命令: initArray(5)
40.00 B 内存申请成功! 地址为: 0000000006E4AFC0
数组创建成功! 地址为: 0000000006E4AFC0

请输入命令: changeArrayVal(0000000006E4AFC0, 3, 789)
成功将数组第3位元素修改为789

请输入命令: getArrayVal(0000000006E4AFC0, 3)
数组第3位元素为: 789

请输入命令: visitArray(0000000006E4AFC0)
数组元素为:
0, 0, 0, 789, 0
遍历结束!

请输入命令: delAllArray(0000000006E4AFC0)
0000000006E4AFC0 释放成功!
数组删除成功!

```

## 2.4 堆

```

请输入命令: initHeap(5)
24.00 B 内存申请成功! 地址为: 0000000006E4AFC0
8.00 B 内存申请成功! 地址为: 0000000006E4B008
堆创建成功! 地址为: 0000000006E4AFC0

请输入命令: heapPush(0000000006E4AFC0, 12)
入堆成功!

请输入命令: heapPush(0000000006E4AFC0, 34)
入堆成功!

请输入命令: heapPush(0000000006E4AFC0, 56)
入堆成功!

请输入命令: getHeapTop(0000000006E4AFC0)
堆顶元素为: 56

请输入命令: heapPop(0000000006E4AFC0)
堆顶元素为: 56

请输入命令: outputHeap(0000000006E4AFC0)
堆中元素依次为:
34, 12

请输入命令: delAllHeap(0000000006E4AFC0)
0000000006E4B008 释放成功!
0000000006E4AFC0 释放成功!
堆删除成功!

```

## 2.5 栈

```

请输入命令: initStack()
16.00 B 内存申请成功! 地址为: 0000000006E4B000
16.00 B 内存申请成功! 地址为: 0000000006E4AFC0
栈创建成功! 地址为: 0000000006E4B000
请输入命令: stackPush(0000000006E4B000, 123)
16.00 B 内存申请成功! 地址为: 0000000006E4AFF0
进栈成功!
请输入命令: stackPush(0000000006E4B000, 456)
16.00 B 内存申请成功! 地址为: 0000000006E4AFD0
进栈成功!
请输入命令: getStackTopVal(0000000006E4B000)
栈顶元素为: 456
请输入命令: stackPop(0000000006E4B000)
出栈成功! 出栈元素为456
0000000006E4AFD0 释放成功!
请输入命令: getStackSize(0000000006E4B000)
栈中有1个元素
请输入命令: delAllStack(0000000006E4B000)
0000000006E4AFF0 释放成功!
0000000006E4AFC0 释放成功!
0000000006E4B000 释放成功!
栈删除成功!

```

## 2.6 队列

```

请输入命令: initQueue()
24.00 B 内存申请成功! 地址为: 0000000006E4AFC0
16.00 B 内存申请成功! 地址为: 0000000006E4AFF8
队列创建成功! 地址为: 0000000006E4AFC0
请输入命令: queuePush(0000000006E4AFC0, 123)
16.00 B 内存申请成功! 地址为: 0000000006E4AFD8
入队成功!
请输入命令: queuePush(0000000006E4AFC0, 456)
16.00 B 内存申请成功! 地址为: 0000000006E4AFE8
入队成功!
请输入命令: getQueueHeadVal(0000000006E4AFC0)
队首元素为: 123
请输入命令: queuePop(0000000006E4AFC0)
出队成功! 元素值为123
0000000006E4AFD8 释放成功!
请输入命令: getQueueSize(0000000006E4AFC0)
队列中有1个元素
请输入命令: delAllQueue(0000000006E4AFC0)
0000000006E4AFE8 释放成功!
0000000006E4AFF8 释放成功!
0000000006E4AFC0 释放成功!
队列删除成功

```



## 2.7 树

```

请输入命令: initTree()
16.00 B 内存申请成功! 地址为: 0000000006B55040
树创建成功! 地址为: 0000000006B55040

请输入命令: addTreeRootVal (0000000006B55040, 1)
32.00 B 内存申请成功! 地址为: 0000000006F55010
根节点创建成功! 地址为: 0000000006F55010
成功给根节点赋值!

请输入命令: addTreeNodeInRight (0000000006B55040, 0000000006F55010, 2)
32.00 B 内存申请成功! 地址为: 0000000006F54FD0
成功插入右子树!

请输入命令: addTreeNodeInLeft (0000000006B55040, 0000000006F54FD0, 3)
32.00 B 内存申请成功! 地址为: 0000000006F54FF0
成功插入左子树!

请输入命令: preOrderVisitTree (0000000006B55040)
前序遍历:
1 2 3

请输入命令: levelOrderVisitTree (0000000006B55040)
层序遍历:
1
NULL 2
3 NULL
NULL NULL
遍历结束!

```

```

请输入命令: getTreeNodeParent (0000000006B55040, 0000000006F54FD0, 3)
该节点的祖先地址为: 0000000006F55010, 值为: 1

请输入命令: getTreeNodeChildren (0000000006B55040, 0000000006F54FD0, 3)
节点的左孩子地址为: 0000000006F54FF0, 值为3
节点的右孩子为空!

请输入命令: getTreeNodeBrother (0000000006B55040, 0000000006F54FD0, 3)
该节点左兄弟为空!

请输入命令: addTreeNodeInLeft (0000000006B55040, 0000000006F55010, 9)
32.00 B 内存申请成功! 地址为: 0000000006F54F50
成功插入左子树!

请输入命令: visualTree (0000000006B55040)
2
3
1 9

请输入命令: delAllTree (0000000006B55040)
0000000006F55010 释放成功!
0000000006F54F50 释放成功!
0000000006F54FD0 释放成功!
0000000006F54FF0 释放成功!
0000000006B55040 释放成功!
树删除成功!

```

## 2.8 图

```

请输入命令: initMap()
32.00 B 内存申请成功! 地址为: 0000000006B55040
16.00 B 内存申请成功! 地址为: 0000000006F55020
24.00 B 内存申请成功! 地址为: 0000000006F54FE0
图创建成功! 地址为: 0000000006B55040, long long

请输入命令: addMapNode(0000000006B55040, 1)
16.00 B 内存申请成功! 地址为: 0000000006F55030
节点添加成功!

请输入命令: addMapNode(0000000006B55040, 2)
16.00 B 内存申请成功! 地址为: 0000000006F54FF8
节点添加成功!

请输入命令: addMapNode(0000000006B55040, 5)
16.00 B 内存申请成功! 地址为: 0000000006F55008
节点添加成功!

请输入命令: addMapNode(0000000006B55040, 8)
16.00 B 内存申请成功! 地址为: 0000000006F54F60
节点添加成功!

请输入命令: addMapEdge(0000000006B55040, 1, 8)
24.00 B 内存申请成功! 地址为: 0000000006F54FB0
16.00 B 内存申请成功! 地址为: 0000000006F54FC8
16.00 B 内存申请成功! 地址为: 0000000006F54F70
边添加成功!

```

```

请输入命令: addMapEdge(0000000006B55040, 1, 5)
16.00 B 内存申请成功! 地址为: 0000000006F54FA0
边添加成功!

请输入命令: addMapEdge(0000000006B55040, 2, 5)
24.00 B 内存申请成功! 地址为: 0000000006F54F80
16.00 B 内存申请成功! 地址为: 0000000006F54E60
16.00 B 内存申请成功! 地址为: 0000000006F54F50
边添加成功!

请输入命令: addMapEdge(0000000006B55040, 2, 2)
16.00 B 内存申请成功! 地址为: 0000000006F54F30
边添加成功!

请输入命令: visitAllMapNode(0000000006B55040)
图的顶点有:
1 2 5 8

请输入命令: visitAllMapEdge(0000000006B55040)
图的边有:
1->8 1->5
2->5 2->2

请输入命令: inMapNode(0000000006B55040, 2)
进入该节点的点有: 2

请输入命令: outMapNode(0000000006B55040, 2)
离开该节点的点有: 5 2

```

```

请输入命令: delMapNode(0000000006B55040, 2)
0000000006F54FF8 释放成功!
0000000006F54F50 释放成功!
0000000006F54F30 释放成功!
0000000006F54E60 释放成功!
0000000006F54F80 释放成功!
节点删除成功!

请输入命令: visitAllMapEdge(0000000006B55040)
图的边有:
1->8 1->5

请输入命令: delAllMap(0000000006B55040)
0000000006F54F70 释放成功!
0000000006F54FA0 释放成功!
0000000006F54FC8 释放成功!
0000000006F54FB0 释放成功!
0000000006F54FE0 释放成功!
0000000006F55030 释放成功!
0000000006F55008 释放成功!
0000000006F54F60 释放成功!
0000000006F55020 释放成功!
0000000006B55040 释放成功!
图删除成功!

请输入命令: checkFreeSpaceList()
空闲块链表:
64.00 MB: 0000000006B55040
32.00 MB: 0000000004B55040
4.00 MB: 0000000006B55040

请输入命令: exit()

请输入命令: checkSpace()
已使用空间: 0.00 MB
剩余空间: 100.00 MB
总空间: 100.00 MB

```