

# § 13. 动态内存申请 - realloc专题讨论 - realloc专题讨论



要求:

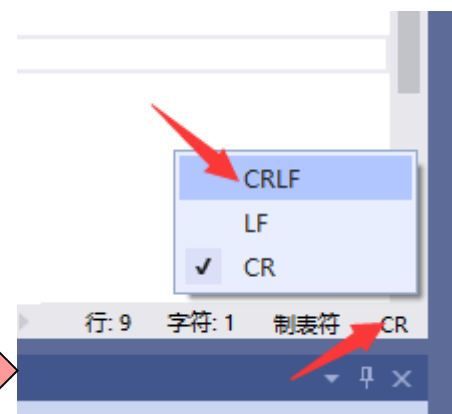
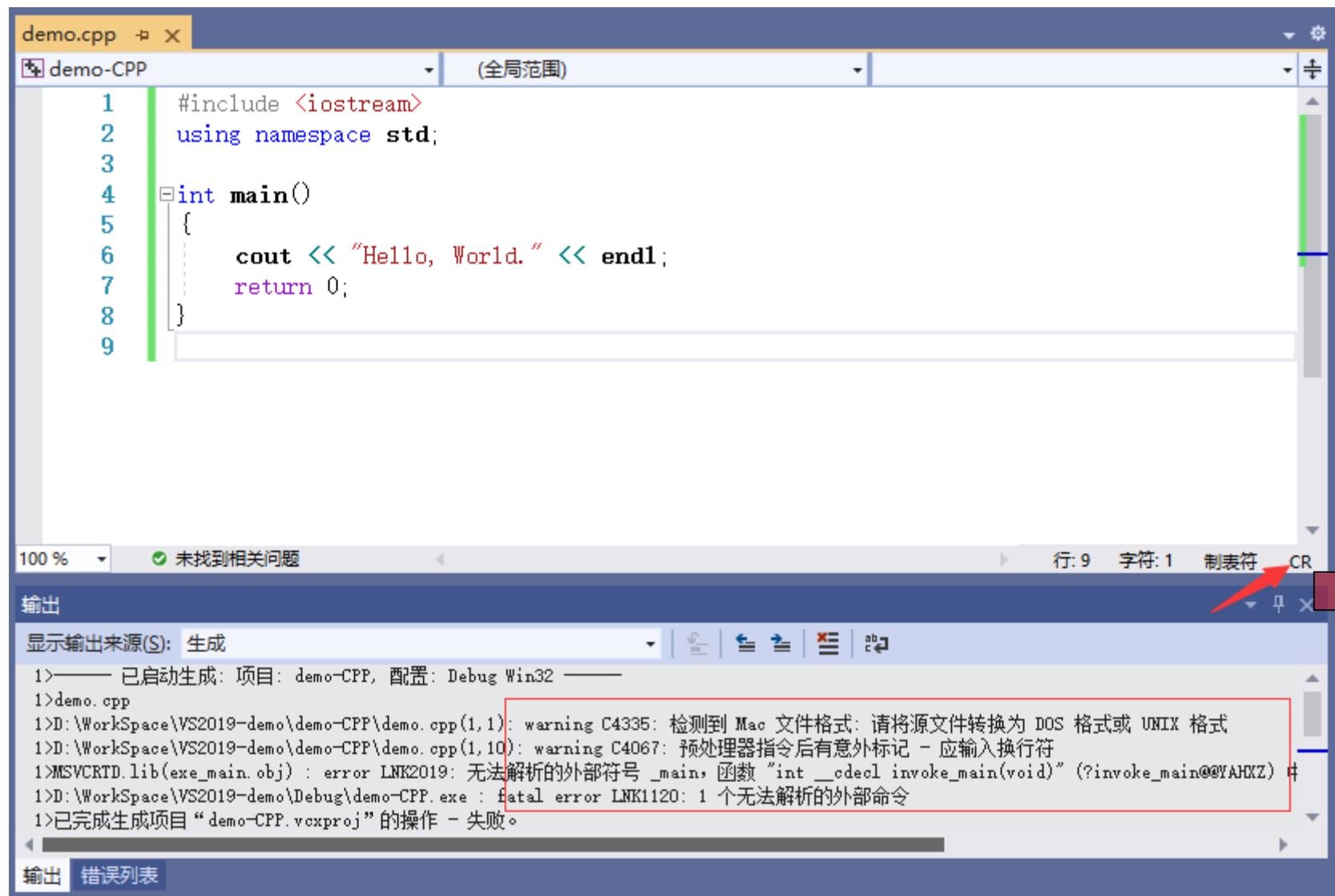
- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用VS2019编译即可
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、不允许手写拍照）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
  - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
  - ★ **不允许**手写在纸上，再拍照贴图
  - ★ **允许**在各种软件工具上手写完成，再截图贴图
- 4、转换为pdf后提交
- 5、**4月1日前**网上提交本次作业（在“实验报告”中提交）

# § 13. 动态内存申请 - realloc专题讨论 - realloc专题讨论



附：用WPS等其他第三方软件打开PPT，将代码复制到VS2019中后，如果出现类似下面的**编译报错**，则观察源程序编辑窗

的右下角是否为CR，如果是，单击CR，在弹出中选择CRLF，再次CTRL+F5运行即可





## § 13. 动态内存申请 - realloc专题讨论

### 3. 内存的动态申请与释放

#### ★ realloc专题讨论

函数形式:

```
void *realloc(void *ptr, unsigned newsize);
```

- (1) 表示为指针ptr重新申请newsize大小的空间
- (2) ptr必须是malloc/calloc/realloc返回的指针
- (3) 如果ptr为NULL, 则等同于malloc
- (4) 如果ptr非NULL, newsize为0, 则等同于free, 并返回NULL
- (5) 新老空间可重合, 也可能不重合, 若不重合, 原空间原有内容会被复制到新空间, 再释放原空间
- (6) 对申请到的空间不做初始化操作
- (7) 若申请不到, 则返回NULL (此时已有指针ptr不释放)

**本页不用作答**



## § 13. 动态内存申请 - realloc专题讨论

### 3. 内存的动态申请与释放

#### ★ realloc专题讨论 - 例1

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main()
{
    int *p;
    p = (int *)realloc(NULL, 10 * sizeof(int));
    if (p==NULL) {
        cout << "No Memory" << endl;
        return -1;
    }

    for(int i=0; i<10; i++)
        cout << p[i] << endl;
    free(p);
    return 0;
}
```

```
-842150451
-842150451
-842150451
-842150451
-842150451
-842150451
-842150451
-842150451
-842150451
-842150451
```

1、运行截图

2、本例对应的P. 3上列出的知识点是(可多项):

- (1) 表示为指针ptr重新申请newsize大小的空间
- (3) 如果ptr为NULL, 则等同于malloc
- (6) 对申请到的空间不做初始化操作



## § 13. 动态内存申请 - realloc专题讨论

### 3. 内存的动态申请与释放

#### ★ realloc专题讨论 - 例2

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{   int i, *p, *q;
    p = (int *)malloc(10 * sizeof(int)); //省略了是否申请成功的判断
    cout << p << endl;
    for (i=0; i<10; i++)
        p[i] = i*i;

    q = (int *)realloc(p, 20 * sizeof(int));
    cout << p << ' ' << q << endl; //观察地址是否相同
    for (i=0; i<20; i++)
        cout << p[i] << ' '; //观察前10个和后10个数
    cout << endl;
    free(q);
    return 0;
}
```

```
0145ED18
0145ED18 01455590
0 1 4 9 16 25 36 49 64 81 -33686019 -572662307 67108868 14990 21343504 21318608 21343504 21299392 -572662307 0
```

未加载任何符号 × Project4.cpp Project3.cpp Project11.cpp Proje

未加载 wntdll.pdb

F:\C++练习\各种习题练习\De  
00D157C0

```
0x831650
0x831650 0x830500
8586584 8585408 4 9 16 25 36 49 64 81 -1 1952531568 626173492 201345808 1 8585408 626173492 201345813 0 8585408

0xe21650
0xe21654 0
1 4 9 16 25 36 49 64 81 -1 1952531568 773033975 201350761 1 14811328 773033975 201350764 0 14811328 773033975
```

1、运行截图（三编译器）

2、将此处p换成 ++p 或 p+1形式，运行截图（三编译器）

3、本例对应的P. 3上列出的知识点是(可多项):

(5) 新老空间可重合，也可能不重合，若不重合，原空间原有内容会被复制到新空间，再释放原空间



## § 13. 动态内存申请 - realloc专题讨论

### 3. 内存的动态申请与释放

#### ★ realloc专题讨论 - 例3

未加载 wntdll.pdb

```
#include <iostream>
#include <cstdlib>
using namespace std;
```

```
int main()
{
```

```
    int *p, *q;
    p = (int *)malloc(10 * sizeof(int));
    cout << p << endl;
```

```
    q = (int *)realloc(p, 20 * sizeof(int)); //省略了是否申请成功的判断
    cout << p << ' ' << q << endl;
```

```
    free(p);
    free(q);
```

```
    return 0;
```

```
}
```

wntdll.pdb 包含查找模块 ntdll.dll 的源文件

F:\C++练习\各种习题练习\

009F57C0  
009F57C0 009F57C0

011957C0  
011957C0 011957C0

未加载 wntdll.pdb

wntdll.pdb 包含查找模块 ntdll.dll 的源文件

F:\C++练习\各种习题练习\

009F57C0  
009F57C0 009F57C0

009859E8  
009859E8 009859E8

1、运行截图（三编译器）

2、运行截图（三编译器）后再次运行，运行截图（三编译器）

3、此外20换成5(小于原大小即可)，再重复1/2，运行截图（三编译器）

4、本例对应的P. 3上列出的知识点是(可多项):

(5) 新老空间可重合，也可能不重合，若不重合，原空间原有内容会被复制到新空间，再释放原空间

0x951650  
0x951650 0x951650

0x7c1650  
0x7c1650 0x7c0500

0x7d1650  
0x7d1650 0x7d0500

0xde1650  
0xde1650 0xde1650



# § 13. 动态内存申请 - realloc专题讨论

## 3. 内存的动态申请与释放

### ★ realloc专题讨论 - 例4

```
//先打开Windows的任务管理器，再观察程序的运行

#include <iostream>
#include <cstdlib>
using namespace std;

int main()
{
    char *p, *q;

    p = (char *)malloc(100 * 1024);
    if (p == NULL) {
        cout << "申请空间失败，请减少申请值后重试" << endl;
        return -1;
    }
    cout << "申请完成，请在任务管理器中观察内存占用" << endl;
    getchar(); //暂停，不释放内存

    q = (char *)realloc(p, 0); //0字节
    cout << (q==NULL ? "NULL" : q) << endl;
    cout << "realloc 0字节完成，请在任务管理器中观察内存占用" << endl;
    getchar(); //暂停，不退出程序

    return 0;
}
```

|                                |      |          |        |        |
|--------------------------------|------|----------|--------|--------|
| Visual Studio Debugger Cons... | 0.3% | 102.7 MB | 0 MB/秒 | 0 Mbps |
| Visual Studio Debugger Cons... | 0%   | 2.6 MB   | 0 MB/秒 | 0 Mbps |
| ConsolePauser.exe (32 位) (3)   | 0%   | 2.4 MB   | 0 MB/秒 | 0 Mbps |
| ConsolePauser.exe (32 位) (3)   | 0%   | 2.4 MB   | 0 MB/秒 | 0 Mbps |

- 1、运行截图（VS+Dev双编译器，截对应程序的内存占用图）
- 2、本例对应的P. 3上列出的知识点是(可多项):  
(4) 如果ptr非NULL, newsize为0, 则等同于free, 并返回NULL





## § 13. 动态内存申请 - realloc专题讨论

### 3. 内存的动态申请与释放

#### ★ realloc专题讨论 - 例5

//先打开Windows的任务管理器，再观察程序的运行

```
#include <iostream>
#include <cstdlib>
using namespace std;
```

```
int main()
{
```

```
    char *p, *q;
```

```
    p = (char *)malloc(100 * 1024 * 1024 * sizeof(char)); //100MB, 此处要保证成功
```

```
    if (p == NULL) {
        cout << "申请空间失败, 请减少申请值后重试" << endl;
        return -1;
    }
```

```
    cout << "申请完成, 请观察内存占用" << endl;
    getchar(); //暂停, 不释放内存
```

```
    q = (char *)realloc(p, 2048U * 1024 * 1024 * sizeof(char));
    if (q == NULL) //如果不提示失败, 则2048继续增大
        cout << "realloc失败, 请观察内存占用" << endl;
    getchar(); //暂停, 不退出程序
```

```
    free(p);
    return 0;
}
```

> Visual Studio Debugger Cons...

> Visual Studio Debugger Cons...

> ConsolePauser.exe (32 位) (3)

> ConsolePauser.exe (32 位) (3)

|      |          |        |        |
|------|----------|--------|--------|
| 0.3% | 102.7 MB | 0 MB/秒 | 0 Mbps |
|------|----------|--------|--------|

|      |          |        |        |
|------|----------|--------|--------|
| 0.3% | 102.7 MB | 0 MB/秒 | 0 Mbps |
|------|----------|--------|--------|

|    |        |        |        |
|----|--------|--------|--------|
| 0% | 2.4 MB | 0 MB/秒 | 0 Mbps |
|----|--------|--------|--------|

|    |        |        |        |
|----|--------|--------|--------|
| 0% | 2.4 MB | 0 MB/秒 | 0 Mbps |
|----|--------|--------|--------|

1、运行截图 (VS+Dev双编译器, 截对应程序的内存占用图)

2、为什么2048要加U?

有符号整型已超最大值

3、本例对应的P. 3上列出的知识点是(可多项):

4、(网上常见的)realloc的用法中, 如果传入指针和返回指针用同一个, 是\_错误\_(正确/错误)的, 会有什么隐患出现?