

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



要求:

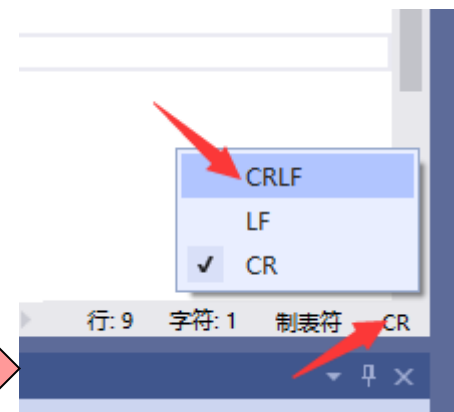
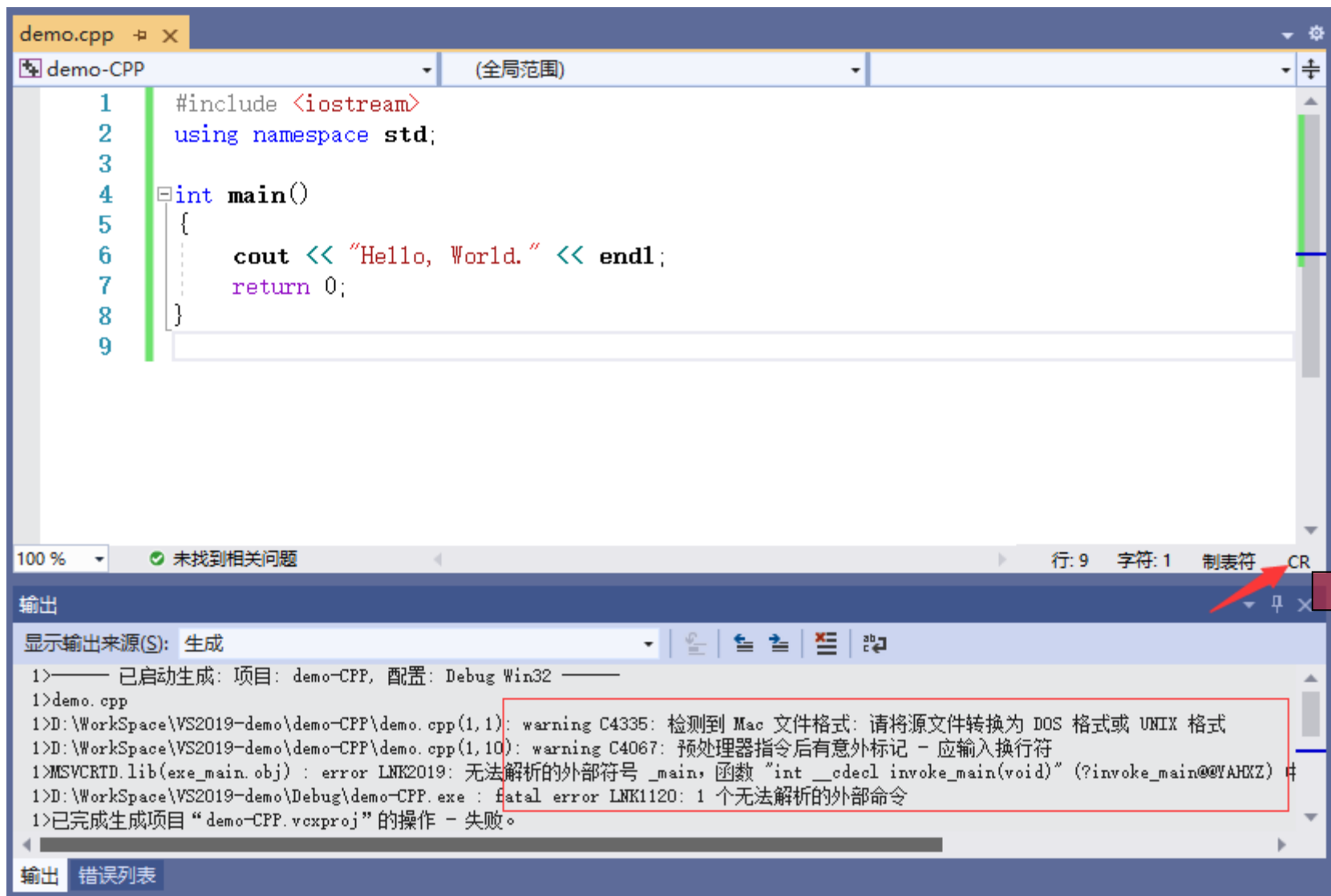
- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用VS2019编译即可
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、不允许手写拍照）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
 - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
 - ★ **不允许**手写在纸上，再拍照贴图
 - ★ **允许**在各种软件工具上手写完成，再截图贴图
- 4、转换为pdf后提交
- 5、**4月1日前**网上提交本次作业（在“实验报告”中提交）

§ 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;
}
```

Microsoft Visual Studio 调试控制台

```
-842150451
-842150451
-842150451
-842150451
-842150451
-842150451
-842150451
-842150451
-842150451
-842150451
D:\workspace\2021-spring-s
```

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
- (7) 若申请不到, 则返回NULL (此时已有指针ptr不释放)

```
Microsoft Visual Studio 调试控制台
00DCE6E0
00DCE6E0 00DC5480
0 1 4 9 16 25 36 49 64 81 -33686019 0 1557769218 -1996485376 14457136 14478256 2054378288 129 2 43

D:\workspace\2021-spring-sj-TinyHW\test_2\源.exe

0x1f7130
0x1f7130 0x1f71a0
0 1 4 9 16 25 36 49 64 81 0 0 -1057626531 -1771485859 -1643688873 900392662 903193892 902489854 -1926697208 32766

[u2051995@10074201 TinyHW]$ ./a.out
0xd74eb0
0xd74eb0 0xd752f0
0 0 14037008 0 16 25 36 49 64 81 1041 0 540024880 858797105 942682167 824193056 892477494 540422944 908081460 825761844
```

int main()

```
D:\workspace\2021-spring-sj-TinyHW\Project1\Debug\test_2.exe
0153D300

Microsoft Visual C++ Runtime Library

Debug Assertion Failed!

Program:
D:\workspace\2021-spring-sj-TinyHW\Project1\Debug\test_2.exe
File: minkernel\crt\src\appcrt\heap\debug_heap.cpp
Line: 604

Expression: CrtIsValidHeapPointer(block)

D:\workspace\2021-spring-sj-TinyHW\Project1\Debug\test_2.exe
0x887110

Process exited after 1 with return value 3221226356
请按任意键继续. . .

[u2051995@10074201 TinyHW]$ ./a.out
0x130feb0
realloc(): invalid pointer
已放弃 (核心已转储)
```

省略了是否申请成功的判断

; //省略了是否申请成功的判断
址是否相同

后10个数

```
D:\workspace\2021-spring-sj-TinyHW\Project1\Debug\test_2.exe
0128DAD8

Microsoft Visual C++ Runtime Library

Debug Assertion Failed!

Program:
D:\workspace\2021-spring-sj-TinyHW\Project1\Debug\test_2.exe
File: minkernel\crt\src\appcrt\heap\debug_heap.cpp
Line: 604

Expression: _CrtIsValidHeapPointer(block)

D:\workspace\2021-spring-sj-TinyHW\Project1\Debug\test_2.exe
0xa27110

Process exited after 1 with return value 3221226356
请按任意键继续. . .

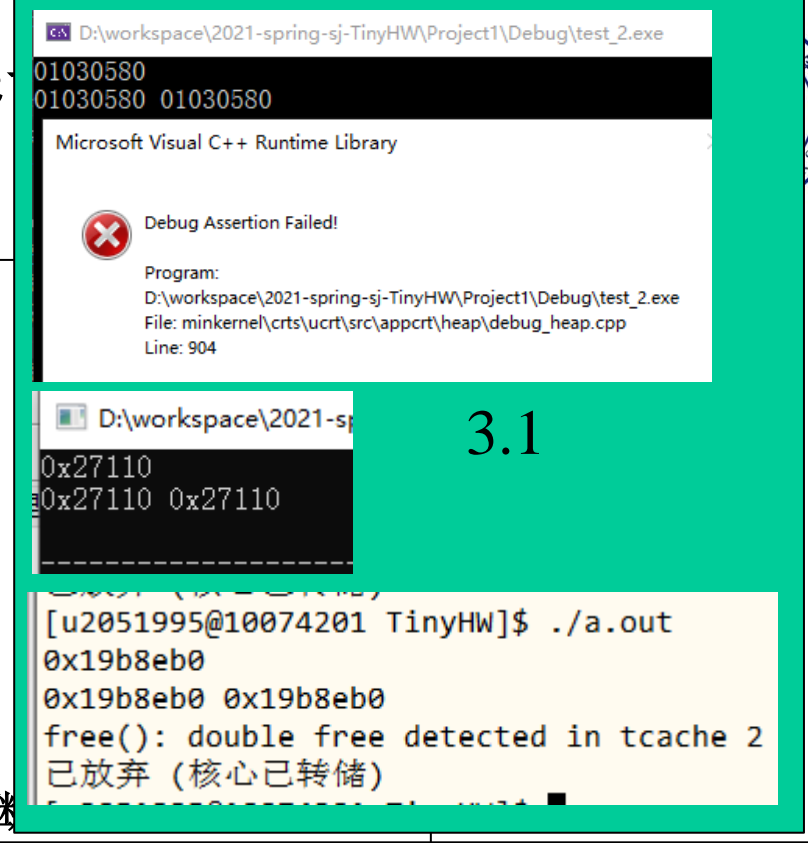
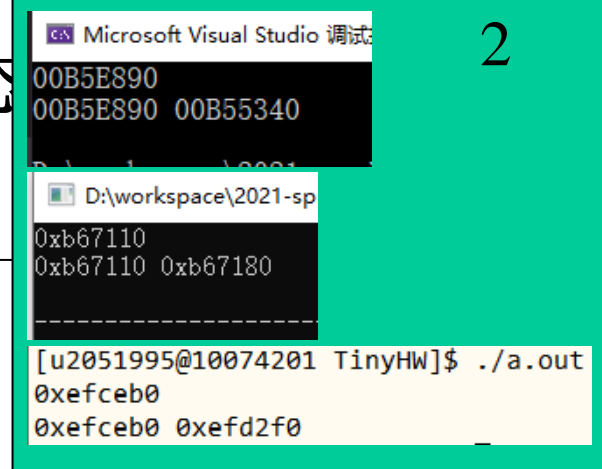
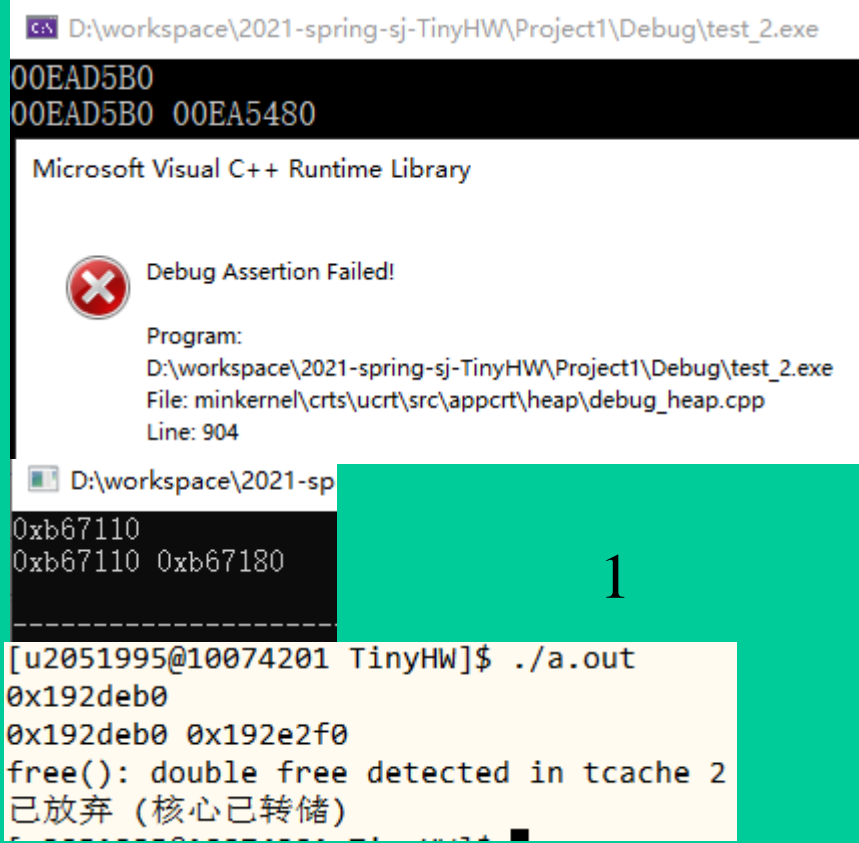
[u2051995@10074201 TinyHW]$ ./a.out
0x2390eb0
realloc(): invalid pointer
已放弃 (核心已转储)
```

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）

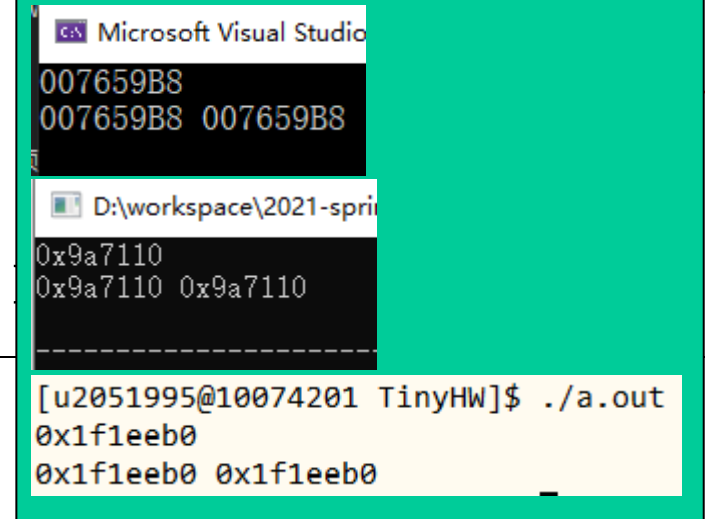
3、本例对应的P. 3上列出的知识点是(可多项)：

- (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
- (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



t)); //省略了是否申请成功的判断
f(int)); //省略了是否申请成功的判断

cout << p << " " << q << endl;



- 1、运行截图（三编译器）
- 2、注释掉free(p)后再次运行，运行截图（三编译器）
- 3、此处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):
 - (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
 - (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
- 加：不能free已经free掉的指针，即使它指向对应的地址数值被分配了

§ 13. 动态内存

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 * 1024 * sizeof(char)); //100MB
    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; //NULL不能直接打印
    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;
}
```



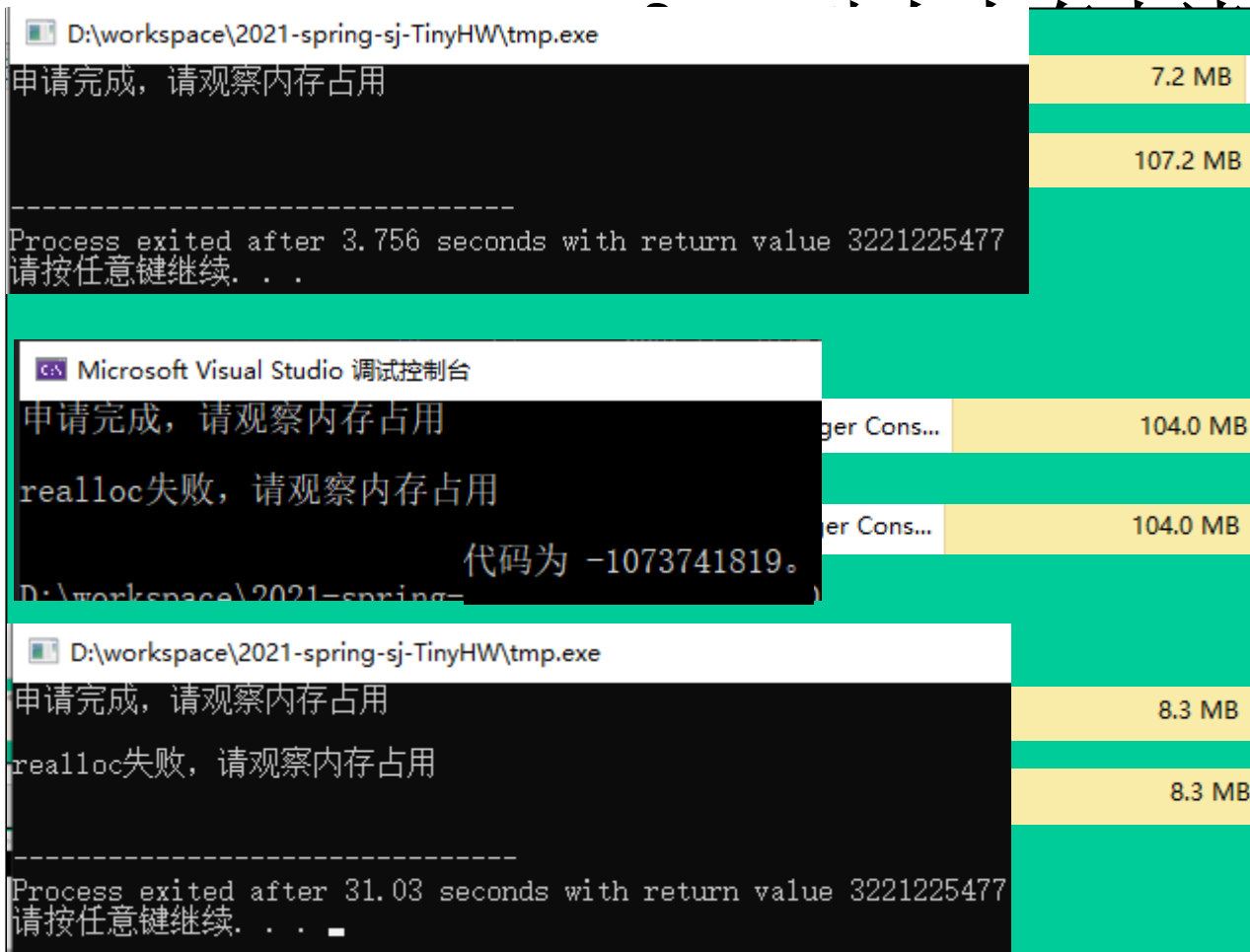
1、运行截图（VS+Dev双编译器，截对应程序的内存占用图）

2、本例对应的P. 3上列出的知识点是(可多项):

- (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
 - (4) 如果ptr非NULL, newsize为0, 则等同于free, 并返回NULL
- 加: 不同编译器对malloc处理不同



- realloc专题讨论



DEV测试时第一次按回车没显示，然后要再按一次才会有错误并且进程退出？
有一次DEV测试时，内存到了2G，可是没有截图就进程结束了。
后来把2048改大成4096就出来了.....

- 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？
不然全都是默认int，乘出来超了2147483647。
- 3、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
 - (2) ptr必须是malloc/calloc/realloc返回的指针
 - (7) 若申请不到，则返回NULL（此时已有指针ptr不释放）
- 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____(正确/错误)的，会有什么隐患出现？

正确
隐患：若分配失败，那那个指针又会被赋值成NULL，原来指向的空间就丢失了。