**目标：**理解分页内存管理的基本原理。实践编程技巧，实现内存分配算法。

**项目描述**：

内存管理器必须提供能够被其他进程调用的下列函数：

1.void \*malloc(int n)：此函数请求一个有n个连续字节的内存块。如果该请求得到满足，它返回指向被分配的内存块的第一个可用字节的指针。如果由于没有足够大的内存空闲块或者参数n是非法的(如为负数)，请求没有被满足，它返回NULL。（该函数同UNIX中的 malloc 函数相似。）

2.void free(int \*p)：此函数会释放先前分配的、由指针p指向的内存块。为了防止内存碎片增多而形成更小的空闲块，如果被释放的内存块同其他空闲块相连，该函数必须合并这些空闲块。如果释放的内存块被已分配的内存块围绕着，就简单地将它添加到空闲块列表中。（这个函数同UNIX中的 free 函数相似。）

3.void \*init()：这个函数通过创建适当的标志、大小和指针域来将内存初始化为一个单一的空闲块。

**测试用例：**假设目前的内存大小为256KB，页大小为4KB，且

空闲内存块情况：

起始地址：0，空闲块大小：32

起始地址：64，空闲块大小：8

起始地址：128，空闲块大小：128

已分配内存块情况:

起始地址：32，大小：32

起始地址：72，大小：56

**实验要求：**

1.设计并实现内存管理器的3个函数。

2.在malloc中至少实现三种不同的分配策略（最佳适应、最差适应、首次适应）。

1)实现最佳适应：遍历整个内存，找到能够满足请求且大小最小的空闲区域。

2)实现最坏适应：找到最大的空闲区域，即使它比请求的内存大很多。

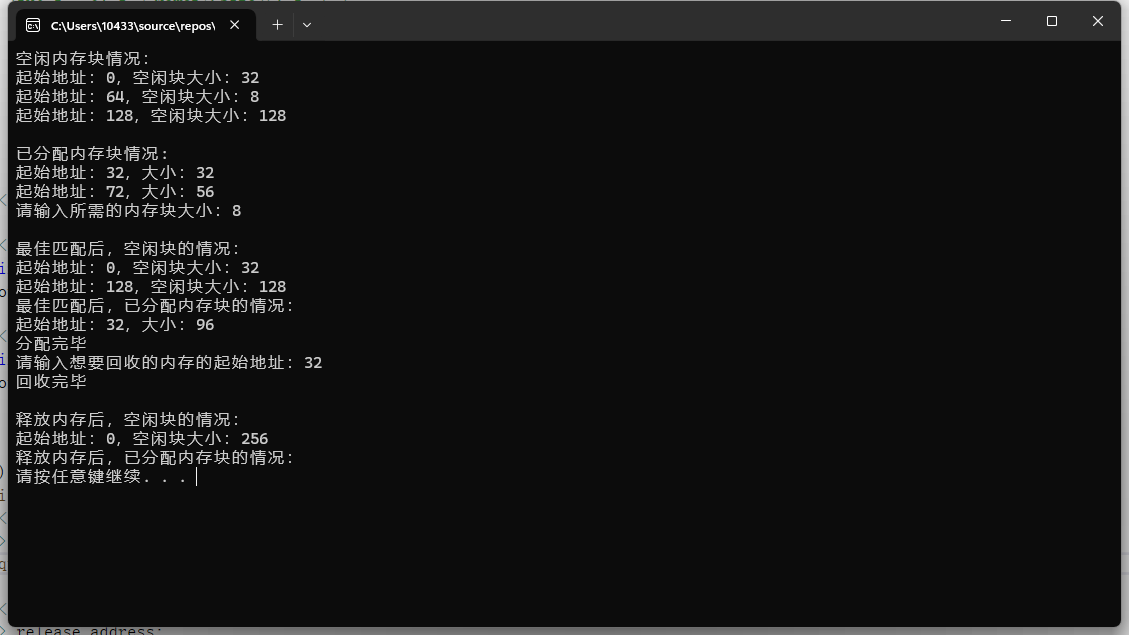
3)实现首次适应：从内存的开始处找到第一个足够大的空闲区域。

3.根据测试用例，分别用以上三种策略分配空间，给出以下两个实验结果。

**实验一：**申请4KB内存块，并回收第一个已分配的内存块。

**实验二：**申请8KB内存块，并回收第一个已分配的内存块。

以下给出实验二的部分实验结果，可供参考。



4.用C/C++实现，提交源代码、实验报告（markdown格式）和视频报告，在关键代码处给出注释。

5.视频报告：提交人讲解实验设计、代码及测试实验演示，时间控制在5分钟以内。