

挑战30天C 入门极限：c 中关于堆内存(heap)的概念和操作方法
的教程 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/134/2021_2022__E6_8C_91_E6_88_9830_E5_A4_c97_134246.htm 堆内存是什么呢？我们知道在c/c 中定义的数组大小必需要事先定义好，他们通常是分配在静态内存空间或者是在栈内存空间内的,但是在实际工作中，我们有时候却需要动态的为数组分配大小，在这里c库中的malloc.h头文件中的malloc()函数就为您解决了问题，（bc 或者是在老的标准中是alloc.h），它的函数原形是void* malloc(size_t size),在动态开辟的内存中，在使用完后我们要使用free()函数来释放动态开辟的内存空间！下面我们来看一个完整的例子！//程序作者:管宁 //站点:www.cndev-lab.com //所有稿件均有版权,如要转载,请务必著名出处和作者 #include <iostream> #include using namespace std. main() { int arraysize. //元素个数 int *array. //用于动态开辟数组的指针变量 cin>>arraysize. array=(int*)malloc(arraysize * sizeof(int)).//利用malloc在堆内存中开辟内存空间,它的大小是元素的个数乘以该数据类型的长度 for(int i=0.i{ array[i]=i. } for(int i=0.i{ cout<<array[i]<<endl. } free(array).//利用free释放动态开辟的堆内存空间 cin.get(). cin.get(). } 这里要特别注意个地方就是: array=(int*)malloc(arraysize * sizeof(int)). malloc()的函数原形本身是void* malloc(size_t size),由于动态分配的空间计算机并不知道是用来做什么的所以是无类型的，但你要把它用在动态的整形数组上的时候就要显式的转换成int*了！下面我们再介绍c 所独有的开辟和释放堆内存空间的方法,new修饰符和delete修饰符. new和delete修饰符的操作并不需要头文件的

支持,这是c所独有的,new操作要比malloc更为简单,直接说明开辟的类型的数目就可以了,delete使用的时候如果是数组那么必须使用delete[]。//程序作者:管宁 //站点:www.cndev-lab.com
//所有稿件均有版权,如要转载,请务必著名出处和作者
#include <iostream> using namespace std. main() { int arraysize. //元素个数 int *array. cin>>arraysize. array=new int[arraysize].//开辟堆内存 for(int i=0;i<arraysize;i++) array[i]=i. } for(int i=0;i<arraysize;i++) cout<<array[i]<<endl. delete[] array.//释放堆内存 cin.get(). cin.get(). } 100Test 下载频道开通, 各类考试题目直接下载。详细请访问 www.100test.com