## 8.9作业

（1）栈区是向下生成，内存存储是由高地址到低地址，堆区是向上生长，内存储存是由低地址到高地址；

1. 栈区内存存放是连续的，而堆区内存分配是不连续的。

5.

野指针出现原因：（1） 定义指针变量时未初始化；

（2） 循环时指针超过特定的范围，不知道指向哪里；

（3） 指针指向的内存被释放后未置空。

避免方法：（1） 指针定义时初始化；

（2） 释放内存后及时置空——free();

（3） 避免指针越界，检查正确的范围。

6.

int main()

{

int arr[5] = { 1, 2, 3, 4, 5};

//假设arr为0x10000000

int\* ptr1 = (int\*)(&arr + 0x1);

//&arr是数组指针类型，&arr+1会跳过整个数组，跳到数组末尾的下一位置，再将&arr+0x1整数指针化，prt1指向arr[4]的下一个地址。

int\* ptr2 = (int\*)((int)arr + 0x1);

//arr是char指针类型，（int）将地址强制转化为16进制整数（纯数值0x10000000），加0x1后为数值0x10000001,然后将整数数值int指针化，变成地址0x10000001,prt2指向该地址。

printf("%x,%x", ptr1[-1], \*ptr2);

//ptr1[-1]等价于\*(ptr1-1)，向前偏移1个int大小，指向arr[4]，值为16进制5。

//因为prt2为int类型的指针，读取内容时会读取4个字节的内容，又因为prt2指向arr偏移1字节的位置，所以读取的便是arr[0]的后三个字节和arr[1]的第一个字节。

又因为电脑存储为小端存储，十进制1转化为16进制为0x00000001,存储先后为0x01、0x00、0x00、0x00，同理十进制2的存储为0x02、0x00、0x00、0x00，

所以prt2读取的便是0x00、0x00、0x00、0x02，即0x00000002，16进制的2。

return 0;

}

所以输出为5，2

7.

（1）类型安全：枚举常量是有类型的，编译器会检查赋值；

#define无类型，本质上是文本替换。

（2）作用域和命名空间：枚举常量被限制在枚举作用域，避免冲突；

#define全局生效，不能重名。

1. 可读性和可维护性：枚举常量有逻辑关联，修改方便；

#define零散常量无关联，需要逐个修改。

1. 自动赋值：枚举常量自动增值赋值（默认从零开始）；

#define需手动赋值，无内在顺序。