

针对JAVA环境的额外说明

由于 java 中不存在 C 语言中指针的概念，并且由于本题中模拟的“堆”大小不超过 100 MB，在 java 版本的解题框架下我们采用相对于堆起始地址的偏移量的值（`int` 型）来表示一个 C 语言指针对应的地址的概念（C 语言中的 `NULL` 这里用 `-1` 指代）。

在框架中，我们模拟的内存模型采用了一个 `byte` 数组模拟了一个有限大小的堆（heap），其数组索引值即表示相对于堆起始地址的偏移量的值，从而实现类似 C 语言中通过指针访问内存的功能。当你需要访问相应“地址”的内容或者存放“地址”到元数据中，你只需要使用该索引值来代替内存中的地址的概念。

同时我们也提供了一些函数用于对“堆”进行赋值或者取值的操作，你可以通过 `MyUtils` 对象进行调用，下发的框架中也有示例。你不允许直接读写 `MyUtils` 里的 `heap` 数组，因为最终评测使用的框架会有所不同。

具体的函数列表如下：

```
//用byte数组模拟堆，提供类似指针赋值的操作，将value存放到从ptr起后面n个byte中，其中n根据value的类型而定
//这里为了和C语言统一，这里采用了小端模式进行存储（即数据的高字节保存在“内存”的高地址中）。
```

```
//赋值相关操作
```

```
final void setPtrValue(int ptr, byte value)
final void setPtrValue(int ptr, char value)
final void setPtrValue(int ptr, short value)
final void setPtrValue(int ptr, int value)
final void setPtrValue(int ptr, long value)
final void setPtrValue(int ptr, float value)
final void setPtrValue(int ptr, double value)
final void setPtrValue(int ptr, boolean value)
```

```
//取值相关操作
```

```
//返回ptr指向的“地址”存放的相应类型的值。
```

```
final int getPtrValuePtr(int ptr)
final int getPtrValueInt(int ptr)
final int getPtrValueChar(int ptr)
final int getPtrValueShort(int ptr)
final int getPtrValueLong(int ptr)
final int getPtrValueByte(int ptr)
final int getPtrValueBool(int ptr)
```

```
//拷贝相关操作
```

```
//将由src指向地址为起始地址的连续size个字节的数据复制到以dst指向地址为起始地址的空间内。
```

```
void memcpy(int dst, int src, int size)
```