Java-动态规划-有一名科学家想要从一台古董电脑中拷贝文件到自己的电脑中加以研究 题目描述:

有一名科学家想要从一台古董电脑中拷贝文件到自己的电脑中加以研究。但此电脑除了有一个 3.5 寸软盘驱动器以外,没有任何手段可以将文件拷贝出来,而且只有一张软盘可以使用。因此这一张软盘是唯一可以用来拷贝文件的载体。

科学家想要尽可能多地将计算机中的信息拷贝到软盘中,做到软盘中文件内容总大小最大。 已知该软盘容量为 **1474560** 字节。文件占用的软盘空间都是按块分配的,每个块大小为

**512** 个字节。一个块只能被一个文件使用。拷贝到软盘中的文件必须是完整的,且不能采取任何压缩技术。

## 输入描述:

第 1 行为一个整数 N,表示计算机中的文件数量。1 <= N <= 1000。

接下来的第2行到第N+1行(共N行),每行为一个整数,表示每个文件的大小Si,单

位为字节。O <= i < N, O <= Si

输出描述:

科学家最多能拷贝的文件总大小。

补充说明:

为了充分利用软盘空间,将每个文件在软盘上占用的块记录到本子上。即真正占用软盘空间的只有文件内容本身。

示例 1

输入:

3

737270

737272

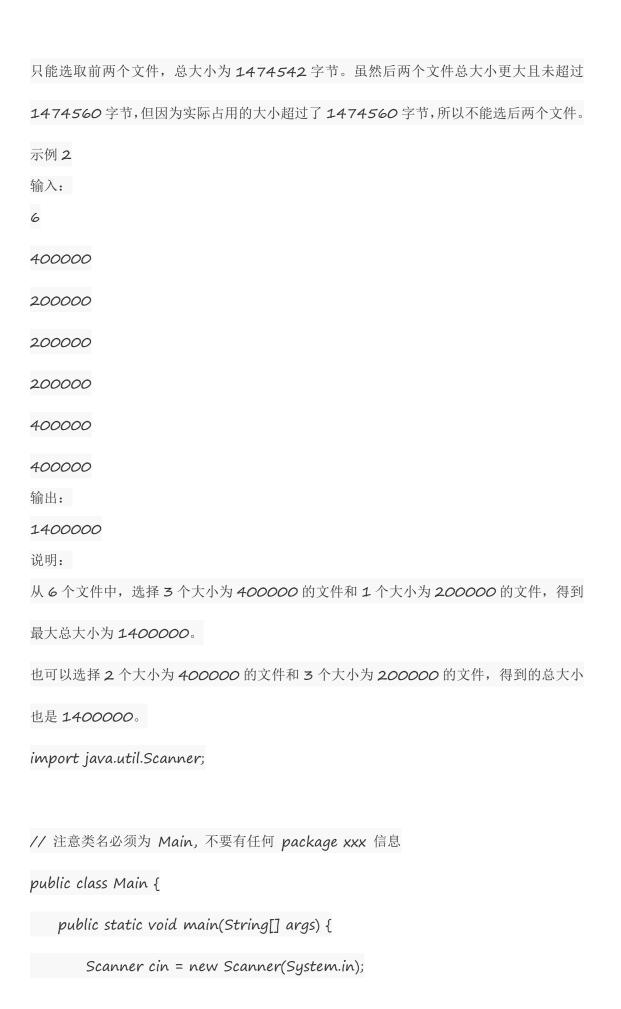
737288

输出:

1474542

说明:

3个文件中,每个文件实际占用的大小分别为 737280 字节、737280 字节、737792 字节。



```
// 注意 hasNext 和 hasNextLine 的区别
    int n = Integer.parseInt(cin.nextLine());
    int[] nums = new int[n];
    int m = 1474560 / 512;
    int[] f = new int[m + 1];
    int[] w = new int[n];
    for(int j = 0; j < n; ++j){
        nums[j] = cin.nextInt();
        w[j] = (int)Math.ceil(nums[j] / 512.0);
    for(int i = 0; i < n; ++i){
        int a = nums[i];
        int b = w[i];
        for(int j = m; j >= b; --j){}
            f[j] = Math.max(f[j], f[j-b] + a);
}
    System.out.println(f[m]);
```