

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。 $1 \leq N \leq 1000$ 。

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 S_i ，单位为字节。 $0 \leq i < N$ ， $0 \leq S_i$

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。

只能选取前两个文件，总大小为1474542字节。虽然后两个文件总大小更大且未超过1474560字节，但因为实际占用的大小超过了1474560字节，所以不能选后两个文件。

示例2

输入：6

400000

200000

200000

200000

400000

400000

输出：1400000

说明：从6个文件中，选择3个大小为400000的文件和1个大小为200000的文件，得到最大总大小为1400000。

也可以选择2个大小为400000的文件和3个大小为200000的文件，得到的总大小也是1400000。

```

1  import java.util.Scanner;
2
3  // 注意类名必须为 Main, 不要有任何 package xxx 信息
4  public class Main {
5      public static void main(String[] args) {
6          Scanner in = new Scanner(System.in);
7          // 注意 hasNext 和 hasNextLine 的区别
8          while (in.hasNextInt()) { // 注意 while 处理多个 case
9              int n = in.nextInt();
10             int[] nums = new int[n];
11             int[] real = new int[n];
12             int[] dp = new int[1474561];
13             for(int i=0;i<n;i++){
14                 nums[i]=in.nextInt();
15                 int t = (nums[i]+511)/512;
16                 real[i]= t*512;
17             }
18
19             for(int i=0;i<n;i++){
20                 int temp = real[i]-1;
21                 for(int j=1474560;j>temp;j--){
22                     dp[j]=Math.max(dp[j-real[i]]+nums[i],dp[j]);
23                 }
24             }
25             System.out.println(dp[1474560]);
26         }
27     }
28 }

```