

## 云短信平台优惠活动

知识点 [哈希表](#) [队列](#) [数组](#) [统计](#) [贪心](#)

时间限制：1s 空间限制：256MB 限定语言：不限

### 题目描述：

某云短信厂商，为庆祝国庆，推出充值优惠活动。

现在给出客户预算，和优惠售价序列，求最多可获得的短信总条数。

### 输入描述：

第一行客户预算M，其中  $0 \leq M \leq 1000000$

第二行给出售价表， $P_1, P_2, \dots, P_n$ ，其中  $1 \leq n \leq 100$ ， $P_i$  为充值  $i$  元获得的短信条数。  
 $1 \leq P_i \leq 1000$ ， $1 \leq n \leq 100$

### 输出描述：

最多获得的短信条数

## 示例1

输入：

6  
10 20 30 40 60

输出：

70

说明：

分两次充值最优，1元、5元各充一次。总条数  $10+60=70$

## 示例2

输入：

15  
10 20 30 40 60 60 70 80 90 150

输出：

210

说明：

分两次充值最优，10元、5元各充一次。总条数  $150+60=210$

## 解题思路：

通过回溯将所有充值的情形模拟一遍，求出其中最大的短信条数

博主使用的是M中取N个数的算法（可以套用到很多题目上），大家也可以使用其他的方法。

```

public class Main{
    public static int max = 0;
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int M = sc.nextInt();
        sc.nextLine();
        String[] P = sc.nextLine().split(" ");

        combine(P, M, new ArrayList<>(), 0);

        System.out.println(max);
    }

    /**
     * M 中取 N 个数的算法
     * @param strings    出售价表
     * @param n          剩余的预算
     * @param list       充值获得的短信条数的集合
     * @param index      出售价表的索引
     */
    public static void combine(String[] strings, int n, List<Integer> list, int index){

        if(n == 0){        //当预算等于 0，退出循环
            int count = 0;
            for(int i=0; i<list.size(); i++){
                count += list.get(i);
            }
            max = Math.max(max, count);
        }else {
            for(int i=index; i<strings.length; i++){
                int x = Integer.valueOf(strings[i]);
                list.add(x);
                combine(strings,n - (i + 1), list, i+1);
                list.remove(list.size()-1);
            }
        }
    }
}

```