



在一组数的编码中，若任意两个相邻的代码只有一位二进制数不同，则称这种编码为格雷码(Gray Code)，请编写一个函数，使用递归的方法生成 N 位的格雷码。

给定一个整数 n ，请返回 n 位的格雷码，顺序为从 0 开始。

测试样例：

1

返回：["0", "1"]

```
1 //方法一：递归
2 //递归的思路就是  $n$  位 gray 码是由  $n-1$  位 gray 码生成，举个例子简单一些：
3 //比如求  $n=3$  的 gray 码，首先知道  $n=2$  的 gray 码是 (00, 01, 11, 10)
4 //那么  $n=3$  的 gray 码其实就是对  $n=2$  的 gray 码首位添加 0 或 1 生成的，添加 0 后变成
5 (000, 001, 011, 010)
6 //添加 1 后需要顺序反向就变成 (110, 111, 101, 100)
7 //组合在一起就是 (000, 001, 011, 010, 110, 111, 101, 100)
8 class GrayCode {
9 public:
10     vector<string> getGray(int n) {
11         // write code here
12         vector<string> gray;
13         if(n == 1){
14             gray.push_back("0");
15             gray.push_back("1");
16             return gray;
17         }
18         vector<string> last_gray = getGray(n-1);
19         for(int i = 0; i < last_gray.size(); i++){
20             gray.push_back("0"+last_gray[i]);
21         }
22         for(int i = last_gray.size()-1; i >= 0; i--){
23             gray.push_back("1"+last_gray[i]);
24         }
25         return gray;
26     };
27
28 //方法二：利用公式  $G(n) = n \text{ XOR } (n/2)$ ，求得 gray 码的十进制之后再转换成二进制字符串
29 class GrayCode {
```



```
30 public:
31     string getbinarystr(int num, int n){
32         string binarystr = "";
33         while(num){
34             int remain = num % 2;
35             stringstream ss;
36             ss << remain;
37             string remainstr;
38             ss >> remainstr;
39             binarystr = remainstr + binarystr;
40             num = num / 2;
41             n--;
42         }
43         while(n-->0)
44             binarystr = "0" + binarystr;
45         return binarystr;
46     }
47     vector<string> getGray(int n) {
48         // write code here
49         int pown = pow(2, n);
50         vector<string> gray;
51         for(int i = 0; i < pown; i++){
52             int graynum = i ^ (i/2);
53             string graystr = getbinarystr(graynum, n);
54             gray.push_back(graystr);
55         }
56         return gray;
57     }
};
```

春节期间小明使用微信收到很多个红包，非常开心。在查看领取红包记录时发现，某个红包金额出现的次数超过了红包总数的一半。请帮小明找到该红包金额。写出具体算法思路和代码实现，要求算法尽可能高效。

给定一个红包的金额数组 **gifts** 及它的大小 **n**，请返回所求红包的金额。
若没有金额超过总数的一半，返回 0。

测试样例：

[1,2,3,2,2],5

返回：2

注意，此题的输入中，不一定有数字出现的次数超过一半，比如一个测试用例 134 个数，有一个数字出现了 67 次，而不是 68，要求返回 0



遍历一遍数组，用一个哈希表存每个数字出现的次数

再遍历一遍哈希表，有次数超过一半的就返回，否则返回 0

```
1  using System.Collections.Generic;
2  class Gift
3  {
4      public int getValue(int[] gifts,  intn)
5      {
6          Dictionary<int,  int> Gifts =  new Dictionary<int,int>();
7          for(int Index =  0; Index < n; Index++)
8          {
9              if(!Gifts.ContainsKey(gifts[Index]))
10                 Gifts.Add(gifts[Index],  1);
11              else
12                 Gifts[gifts[Index]]++;
13          }
14          foreach(KeyValuePair<int,  int> Pair in Gifts)
15          {
16              if(Pair.Value > n /  2)
17                 return Pair.Key;
18          }
19          return 0;
20      }
21 }
```

