

## 卡牌分组

给定一副牌，每张牌上都写着一个整数。

此时，你需要选定一个数字  $x$ ，使我们可以将整副牌按下述规则分成 1 组或更多组：

- 每组都有  $x$  张牌。
- 组内所有的牌上都写着相同的整数。

仅当你可选的  $x \geq 2$  时返回 `true`。

### 示例 1：

输入：[1,2,3,4,4,3,2,1]

输出：true

解释：可行的分组是 [1,1]，[2,2]，[3,3]，[4,4]

### 示例 2：

输入：[1,1,1,2,2,2,3,3]

输出：false

解释：没有满足要求的分组。

### 示例 3：

输入：[1]

输出：false

解释：没有满足要求的分组。

### 示例 4：

输入：[1,1]

输出：true

解释：可行的分组是 [1,1]

### 示例 5：

输入：[1,1,2,2,2,2]

输出：true

解释：可行的分组是 [1,1]，[2,2]，[2,2]

解题思路：排序+找合适分组间隔 分组间隔判断以每组首尾元素是否相等进行判断

```
int cmp(int*a,int*b)
{
    return *a-*b;
}
bool hasGroupsSizeX(int* deck, int deckSize){
    bool res=false;
    int x,num;
    //先从小到大排序
    qsort(deck,deckSize,sizeof(int),cmp);
    for(x=1;x<deckSize;x++)//分组大小不一定，只能通过循环来查找
    {
        if(deckSize%x==0)
        {
            num=deckSize/x;
            for(int i=0;i<deckSize;i+=num)
            {
                if(deck[i]==deck[i+num-1])//由于有序排列，判断分组后直接判断每
组首尾元素即可
                {
                    res=true;
                }
                else
                {
                    res=false;
                    break;
                }
            }
            if(res)//此处需要注意，加一个判断，如果分组条件不满足则继续找，如果
分组满足条件则结束程序
            return res;
        }
    }
    return false;
}
```