金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

你就是一个画家！你现在想绘制一幅画，但是你现在没有足够颜色的颜料。为了让问题简单，我们用正整数表示不同颜色的颜料。你知道这幅画需要的 n 种颜色的颜料，你现在可以去商店购买一些颜料，但是商店不能保证能供应所有颜色的颜料，所以你需要自己混合一些颜料。混合两种不一样的颜色 A 和颜色 B 颜料可以产生(A XOR B)这种颜色的颜料(新产生的颜料也可以用作继续混合产生新的颜色,XOR 表示异或操作)。本着勤俭节约的精神，你想购买更少的颜料就满足要求，所以兼职程序员的你需要编程来计算出最少需要购买几种颜色的颜料？

解题思路

在 C++中，将两个数进行 xor，用的是^符号，但是实际上是将十进制转换为二进制之后，再进行 xor，这样，这 n 个十进制的数，就被转换成了 n 个二进制的包含 1，0 的字符串，将每个数转换成二进制之后单成一行，位数小的前面被补全 0，这样这 n 个数就变成了 n 行矩阵，由于 1 ≤ xi ≤ 1,000,000,000，而 2 的 30 次幂是 10 亿多，所以这个矩阵最大是 n＊30 的矩阵。

现在将这个矩阵列出来，如：

101010
111010
101101
110110

然后进行行与行之间的 xor，其中 1^1=0;　0^0=0;　1^0=1; 0^1=1;
有没有发现这种运算很像求矩阵的秩？相同的相减为 0，不同的相减为 1.

矩阵的秩定义：是其行向量或列向量的极大无关组中包含向量的个数。
矩阵的秩求法：用初等行变换化成梯矩阵，梯矩阵中非零行数就是矩阵的秩.

所以这道题就被转化成了求矩阵的秩，求法如下。

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
//
//  main.cpp
//  NiuKe_HunHeYanLiao
//
//  Created by 麦心 on 16/8/18.
//  Copyright © 2016 年 程序工匠 0_0 小姐 . All rights reserved.
//
#include <iostream>
using namespace std ;
#include <vector>
#include <algorithm>
// 求一个数的二进制的最高位是哪位
int getHighBit( int num){
    int highbit = 0 ;
    while (num) {
        // 将该数的二进制右移一位
         num>>= 1 ;
         highbit++;
     }
    return highbit;
}
int main() {
    vector < int > colors;
```

职场精英工作室出品，唯一淘宝旺旺客服：蔚蓝小小天使
职场精英工作室出品，唯一淘宝旺旺客服：蔚蓝小小天使
职场精英工作室出品，唯一淘宝旺旺客服：蔚蓝小小天使

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

```
int n;

while ( cin >> n){
      colors. clear ();
     int result = 0 ;
     int temp;
     int i = n;
     while (i--) {
          cin >>temp;
           colors. push_back (temp);
      }

     // 将 colors 进行从小到大的排序
     sort (colors. begin (), colors. end ());
     int bigger, smaller;
     //bigger 和 smaller 始终指向的是最后一位和倒数第二位数
      bigger = n - 1 ;
      smaller = bigger - 1 ;

     while (colors. size ()> 2 ) {
         // 如果两者的最高位相同，说明最高位可以消掉，
         // 将两者 xor ，或者说将矩阵两行相减消掉最高位
          if ( getHighBit (colors[ bigger ]) == getHighBit (colors[ smaller ])){
              int tem = colors[ bigger ]^colors[ smaller ];
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
        //find 函数头文件是 <algorithm>
        // 泛型算法的 find ，在非 string 类型的容器里，可以直接找出所对应的元素
        // 从 vector 的头开始一直到尾，找到第一个值为 temp 的元素，返回的是一个指向该元素的迭代器
std::vector<int>::iterator 类型

        // 因为现在 xor 的是两个最大的数，而且最高位已被消掉，所以 xor 得到的结果一定比这两个数小

        // 如果 temp 这个 比最大两个数小的 数没有被找到，则将 temp 加到 colors 数组中，进行再次 xor
        // 找不到的话，返回 colors.end 应该是固定用法
        if ( find (colors. begin (), colors. end (), tem)==colors. end ()){
            colors. push_back (tem);
            sort (colors. begin (), colors. end ());
            bigger++; // 因为 colors 中多了一个数，所以需要位数＋ 1
            smaller++;
        }
    } else {
        result++;
    }
    // 如果两者最高位不同，说明已经所有数的最高位已经只有最大的那个数是 1 了，这样它已经不可能被消掉了，结果＋ 1

    // 如果两个最大数的最高位可以消掉，那么消除之后，最大数已被消掉，没有用了
    // 如果两个最大数的最高位不可以消掉，那么结果＋ 1 ，最大数也没有用了。
    // 弹出最大数
    colors. pop_back ();
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

```
            // 因为弹出了一个数，所以 bigger 和 smaller 都要相应一 1
            bigger = smaller;
            smaller--;
        }
        cout <<result+2 << endl ;
    }
}
```

一个袋子里面有 n 个球，每个球上面都有一个号码(拥有相同号码的球是无区别的)。如果一个袋子是幸运的当且仅当所有球的号码的和大于所有球的号码的积。
例如：如果袋子里面的球的号码是{1, 1, 2, 3}，这个袋子就是幸运的，因为 1 + 1 + 2 + 3 > 1 * 1 * 2 * 3
你可以适当从袋子里移除一些球(可以移除 0 个,但是别移除完)，要使移除后的袋子是幸运的。现在让你编程计算一下你可以获得的多少种不同的幸运的袋子。

题目可以转化成求符合条件的集合真子集个数。每次从全集中选择若干元素（小球）组成子集（袋子）。集合子集个数为 2^n 个，使用 dfs 必然超时。且此题有重复元素，那么就搜索剪枝。

对于任意两个正整数 a,b 如果满足 a+b>a*b，则必有一个数为 1.可用数论证明：

设 a=1+x,b=1+y，则 1+x+1+y>(1+x)*(1+y)，---> 1>x*y，则 x，y 必有一个为 0，即 a,b 有一个为 1.

推广到任意 k 个正整数，假设 a1,a2,...ak，如果不满足给定条件，即和 sum 小于等于积 pi，

如果此时再选择一个数 b,能使其满足 sum+b > pi*b，则，b 必然为 1，且为必要非充分条件。

反之，如果选择的 b>1，则 sum+b <=pi*b，即 a1,a2,...,ak,b 不满足给定条件。（**搜索剪枝的重要依据**）

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

因此，将球按标号**升序排序**。每次从小到大选择，当选择到 a1,a2,...,ak-1 时满足给定条件，而再增加选择 ak 时不满足条件（ak 必然大于等于 max(a1,a2,...,ak-1)），继续向后选择更大的数，必然无法满足！因此，可以进行剪枝。

如果有多个 1，即当 k=1 时，sum(1)>pi(1)不满足，但下一个元素仍为 1，则可以满足 1+1>1*1，所以要判断当前 ak 是否等于 1.

此外，对于重复数字，要去重复。

```
#include <stdio.h>
#include <stdlib.h>
int bag[1001],n;

int comp(const void *a,const void *b){
        return *(int*)a - *(int*)b;
}
int dfs(int pos,long long sum,long long pi){
        int i,c;
        for(i=pos,c=0;i<n;++i){
                sum+=bag[i];
                pi*=bag[i];
                if (sum>pi) c+=1+dfs(i+1,sum,pi);
                else if (bag[i]==1) c+=dfs(i+1,sum,pi);
                else break;
                sum-=bag[i];
                pi/=bag[i];
                for(;i<n-1 && bag[i]==bag[i+1];++i);// duplicate
        }
        return c;
}
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
int main(){
        int i;
        while(~scanf("%d",&n)){
                for(i=0;i<n;scanf("%d",&bag[i++]));
                qsort(bag,n,sizeof(int),comp);
                printf("%d\n",dfs(0,0,1));
        }
        return 0;
}
```

二货小易有一个 W*H 的网格盒子，网格的行编号为 0~H-1，网格的列编号为 0~W-1。每个格子至多可以放一块蛋糕，任意两块蛋糕的欧几里得距离不能等于 2。
对于两个格子坐标(x1,y1),(x2,y2)的欧几里得距离为:
((x1-x2) * (x1-x2) + (y1-y2) * (y1-y2)) 的算术平方根
小易想知道最多可以放多少块蛋糕在网格盒子里。

```
import java.util.*;

public class Main {
        public static void main(String[] args) {
                Scanner in =  new Scanner(System.in);
                int col = in.nextInt();
                int row = in.nextInt();

                int[][] grid =  new int[row][col];

```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
11
12                  for(int i =  0; i < row; i++){
13                          if(i %  4 ==  0 || i %  4 ==  1){
14                                  for(int j =  0; j < col; j++){
15                                          if(j %  4 ==  0 || j %  4 ==  1){
16                                                  grid[i][j] =  1;
17                                          }
18                                  }
19                          }
20                          else {
21                                  for(int j =  0; j < col; j++){
22                                          if(j %  4 ==  2 || j %  4 ==  3){
23                                                  grid[i][j] =  1;
24                                          }
25                                  }
26                          }
27                  }
28
29                  int count =  0;
30                  for(int x =  0; x < row; x++){
31                          for(int y =  0; y < col; y++){
32                                  if(grid[x][y] ==  1){
33                                          count++;
34                                  }
35                          }
36                  }
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

```
37
38                      System.out.println(count);
39              }
40
41      }
```

举个例子，就可以找出规律，是以 4 为周期重复出现的

1 1 x x 1 1

1 1 x x 1 1

x x 1 1 x x

x x 1 1 x x

有一片 1000*1000 的草地，小易初始站在(1,1)(最左上角的位置)。小易在每一秒会横向或者纵向移动到相邻的草地上吃草(小易不会走出边界)。大反派超超想去捕捉可爱的小易，他手里有 n 个陷阱。第 i 个陷阱被安置在横坐标为 x_i，纵坐标为 y_i 的位置上，小易一旦走入一个陷阱，将会被超超捕捉。你为了去解救小易，需要知道小易最少多少秒可能会走入一个陷阱，从而提前解救小易。

```
1   #include<iostream>
2
3   #include<vector>
4
5
6
7   using namespace std;
8
9
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
10
11  intmain()
12
13  {
14
15          intn,tx,ty;
16
17          while(cin>>n)
18
19          {
20
21                  vector<int> vx,vy;
22
23                  for(inti =  0;i<n;++i)
24
25                  {
26
27                          cin>>tx;
28
29                          vx.push_back(tx);
30
31                  }
32
33                  for(inti =  0;i<n;++i)
34
35                  {
```

职场精英工作室出品，唯一淘宝旺旺客服：蔚蓝小小天使
职场精英工作室出品，唯一淘宝旺旺客服：蔚蓝小小天使
职场精英工作室出品，唯一淘宝旺旺客服：蔚蓝小小天使

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
36
37                      cin>>ty;
38
39                      vy.push_back(ty);
40
41              }
42
43              vector<int> vdis;
44
45              intmin =  2001;
46
47              for(inti =  0;i<n;++i)
48
49              {
50
51                      if((vx[i]+vy[i]-2)<min)
52
53                              min = vx[i]+vy[i]-2;
54
55              }
56
57              cout<<min<<endl;
58
59      }
60
61
```

职场精英工作室出品，唯一淘宝旺旺客服：蔚蓝小小天使
职场精英工作室出品，唯一淘宝旺旺客服：蔚蓝小小天使
职场精英工作室出品，唯一淘宝旺旺客服：蔚蓝小小天使

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
62
63          return0;
64
65  }
```

“回文串”是一个正读和反读都一样的字符串，比如“level”或者“noon”等等就是回文串。花花非常喜欢这种拥有对称美的回文串，生日的时候她得到两个礼物分别是字符串 A 和字符串 B。现在她非常好奇有没有办法将字符串 B 插入字符串 A 使产生的字符串是一个回文串。你接受花花的请求，帮助她寻找有多少种插入办法可以使新串是一个回文串。如果字符串 B 插入的位置不同就考虑为不一样的办法。
例如：
A = “aba”，B = “b”。这里有 4 种把 B 插入 A 的办法：
* 在 A 的第一个字母之前: "baba" 不是回文
* 在第一个字母‘a’之后: "abba" 是回文
* 在字母‘b’之后: "abba" 是回文
* 在第二个字母'a'之后 "abab" 不是回文
所以满足条件的答案为 2

```
1   //自然解法
2   #include<iostream>
3   #include<string>
4   using namespace std;
5   bool Huiwen(string str1)      //判断回文
6           {
7           int length=str1.length();
8           for(int i=0;i<length;i++)
9                 {
10                if(str1[i]!=str1[length-1])
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
11                      return false;
12                  length=length-1;
13          }
14          return true;
15  }
16  int main()
17        {
18          string str1,str2,temp;
19          int count,len;
20          while(cin>>str1>>str2)
21                {
22                count = 0;
23                temp=str1;
24                  len=str1.length()+1;
25                for(int i=0;i<len;i++)
26                {
27                      str1=temp;
28                      str1.insert(i,str2);      //在 A 字符串中以此插入 B 字符串
29                      if(Huiwen(str1))              //判断是否是回文
30                          count=count+1;          //统计回文
31                }
32                cout<<count<<endl;
33          }
34          return 0;
35          }
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

小易总是感觉饥饿，所以作为章鱼的小易经常出去寻找贝壳吃。最开始小易在一个初始位置 x_0。对于小易所处的当前位置 x，他只能通过神秘的力量移动到 4 * x + 3 或者 8 * x + 7。因为使用神秘力量要耗费太多体力，所以它只能使用神秘力量最多 100,000 次。贝壳总生长在能被 1,000,000,007 整除的位置(比如：位置 0，位置 1,000,000,007，位置 2,000,000,014 等)。小易需要你帮忙计算最少需要使用多少次神秘力量就能吃到贝壳。

```
1   import java.util.HashMap;
2   import java.util.LinkedList;
3   import java.util.Map;
4   import java.util.Queue;
5   import java.util.Scanner;
6
7   public class Main{
8           public static void main(String[] args) {
9                   Scanner in =  new Scanner(System.in);
10
11                  while(in.hasNext()){
12                          int x=in.nextInt();
13                          Map<Long, Integer> map=new HashMap<Long, Integer>();
14
15                          Queue<Long> queue=new LinkedList<Long>();
16                          queue.offer((long)x);
17                          map.put((long)x,  1);
18
19                          while(!queue.isEmpty()){
20                                  long n=queue.poll();
21                                  if(n==0) {System.out.println(map.get(n)-1);  return;}
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
22                                      if(map.get(n)>=100001)  continue;
23
24                                      if(!map.containsKey((4*n+3)%1000000007)){
25                                              map.put((4*n+3)%1000000007, map.get(n)+1);
26                                              queue.offer((4*n+3)%1000000007);
27                                      }
28                                      if(!map.containsKey((8*n+7)%1000000007)){
29                                              map.put((8*n+7)%1000000007, map.get(n)+1);
30                                              queue.offer((8*n+7)%1000000007);
31                                      }
32                              }
33                              System.out.println(-1);
34                      }
35              }
36  }
37
38
```

考拉有 n 个字符串字符串，任意两个字符串长度都是不同的。考拉最近学习到有两种字符串的排序方法： 1.根据字符串的字典序排序。例如：
"car" < "carriage" < "cats" < "doggies < "koala"
2.根据字符串的长度排序。例如：
"car" < "cats" < "koala" < "doggies" < "carriage"
考拉想知道自己的这些字符串排列顺序是否满足这两种排序方法，考拉要忙着吃树叶，所以需要你来帮忙验证。

```
1   import java.util.Scanner;
2
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
3   /**
4     * Created by Genge on 2016-08-20.
5     */
6   public class Main {
7           public static void main(String[] args) {
8                   Scanner scanner =  new Scanner(System.in);
9                   while (scanner.hasNext()) {
10                          int n = scanner.nextInt();
11                          String[] words =  new String[n];
12                          for (int i =  0; i < n; i ++) {
13                                  words[i] = scanner.next();
14                          }
15                          System.out.println(validate(words));
16                  }
17                  scanner.close();
18          }
19
20          private static String validate(String[] words) {
21                  boolean isABC = isAbc(words);
22                  boolean isLEN = isLen(words);
23                  if (isABC && isLEN) {
24                          return "both";
25                  }
26                  if (isABC) {
27                          return "lexicographically";
28                  }
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
29                  if (isLEN) {
30                          return "lengths";
31                  }
32                  return "none";
33          }
34
35          private static boolean isLen(String[] words) {
36                  boolean result =  true;
37                  for (int i =  1; i < words.length; i++) {
38                          if (words[i].length() <= words[i -  1].length()) {
39                                  result =  false;
40                                  break;
41                          }
42                  }
43                  return result;
44          }
45
46          private static boolean isAbc(String[] words) {
47                  boolean result =  true;
48                  for (int i =  1; i < words.length; i++) {
49                          if (words[i].compareTo(words[i -  1]) <=  0) {
50                                  result =  false;
51                                  break;
52                          }
53                  }
54                  return result;
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
55            }
56    }
```

小易喜欢的单词具有以下特性：
1.单词每个字母都是大写字母
2.单词没有连续相等的字母
3.单词没有形如“xyxy”(这里的 x，y 指的都是字母，并且可以相同)这样的子序列，子序列可能不连续。
例如：
小易不喜欢"ABBA"，因为这里有两个连续的'B'
小易不喜欢"THETXH"，因为这里包含子序列"THTH"
小易不喜欢"ABACADA"，因为这里包含子序列"AAAA"
小易喜欢"A","ABA"和"ABCBA"这些单词
给你一个单词，你要回答小易是否会喜欢这个单词。

```
1    import java.util.Scanner;
2    public class Main {
3
4            public static void main(String[] args) {
5                    Scanner sc =  new Scanner(System.in);
6                    while(sc.hasNext()){
7                            String word = sc.next();
8
9                            if(isAllUpCase(word) && isConEql(word) && isThrEql(word))
10                                   System.out.println("Likes");
11                           else
12                                   System.out.println("Dislikes");
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
13                  }
14          }
15          //条件 1
16            public static boolean isAllUpCase(String word){
17                  return word.matches("[A-Z]+");
18          }
19          //条件 2
20          public static boolean isConEql(String word){
21                  return !word.matches(".*(.)(\\1).*");
22          }
23          //条件 3
24          public static boolean isThrEql(String word){
25                  return !word.matches(".*(.).*(.)(.*\\1)(.*\\2).*");
26          }
27  }
```

Fibonacci 数列是这样定义的：

F[0] = 0

F[1] = 1

for each i ≥ 2: F[i] = F[i-1] + F[i-2]

因此，Fibonacci 数列就形如：0, 1, 1, 2, 3, 5, 8, 13, ...，在 Fibonacci 数列中的数我们称为 Fibonacci 数。给你一个 N，你想让其变为一个 Fibonacci 数，每一步你可以把当前数字 X 变为 X-1 或者 X+1，现在给你一个数 N 求最少需要多少步可以变为 Fibonacci 数。

```
1   #include<iostream>
2   #include<stdio.h>
3   using namespace std;
4
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
//定义 Fibonacci 数的函数
int Fibonacci(int N)
{
        if (N == 0)
                return 0;
        else if (N == 1)
                return 1;
        else
                return Fibonacci(N - 1) + Fibonacci(N - 2);
}

int main()
{
        int N;
        cin >> N;
        int num=0;
//这里 i 的值设置大一点没关系，因为 Fibonacci 函数必然在一个点大于 N
        for (int i = 0; i < 100000; i++)
        {
                if (Fibonacci(i) - N > 0)
                {
                        num = i;
                        break;
                }
        }
//看前一个还是后一个 Fibonacci 数距离 N 近一些
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

```
31          if (Fibonacci(num) - N < N - Fibonacci(num - 1))
32                  cout << Fibonacci(num) - N;
33          else
34                  cout << N - Fibonacci(num - 1);
35          return 0;
36  }
```

小易邀请你玩一个数字游戏，小易给你一系列的整数。你们俩使用这些整数玩游戏。每次小易会任意说一个数字出来，然后你需要从这一系列数字中选取一部分出来让它们的和等于小易所说的数字。 例如： 如果{2,1,2,7}是你有的一系列数，小易说的数字是 11.你可以得到方案 2+2+7 = 11.如果顽皮的小易想坑你，他说的数字是 6，那么你没有办法拼凑出和为 6 现在小易给你 n 个数，让你找出无法从 n 个数中选取部分求和的数字中的最小数。

```
1   /*
2   * Sort 后，查看有序数组的前 n 项和是否与当前项连续，
3   * 如果不连续说明存在一个空档无法通过求和得到
4   * */
5
6   import java.util.Arrays;
7   import java.util.Scanner;
8   public class Main {
9           private static int check(int[] X,  int n){
10                if (X[0]>1)  return 1;
11                else if (n ==  1)  return X[0]+1;
12                else {
13                      int sum = X[0];
14                      for (int i =  1; i < n; i++) {
15                            if (X[i]-sum>1)  break;
```

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。

金融/四大/国企/名企求职笔试面试教育-职场精英工作室，店址：https://huntjob.taobao.com/ 唯一旺旺客服：蔚蓝小小天使

```
16                          else sum += X[i];
17                    }
18                    return sum+1;
19              }
20        }
21        public static void main(String[] args) {
22              Scanner in =  new Scanner(System.in);
23              while (in.hasNext()) {
24                    int n = in.nextInt();
25                    int[] X =  new int[n];
26                    for (int i =  0; i < n; i++) {
27                          X[i] = in.nextInt();
28                    }
29                    Arrays.sort(X);
30                    System.out.println(check(X, n));
31              }
32        }
33  }
```

职场精英工作室出品，唯一淘宝旺旺客服：蔚蓝小小天使

本资料仅限购买者一个人使用，不得分享/转赠/转卖；祝各位获得心仪 offer。版权所有，违者必究。