**分糖果**

问题描述

　　有n个小朋友围坐成一圈。老师给每个小朋友随机发偶数个糖果，然后进行下面的游戏：  
  
　　每个小朋友都把自己的糖果分一半给左手边的孩子。  
  
　　一轮分糖后，拥有奇数颗糖的孩子由老师补给1个糖果，从而变成偶数。  
  
　　反复进行这个游戏，直到所有小朋友的糖果数都相同为止。  
  
　　你的任务是预测在已知的初始糖果情形下，老师一共需要补发多少个糖果。

输入格式

　　程序首先读入一个整数N(2<N<100)，表示小朋友的人数。  
　　接着是一行用空格分开的N个偶数（每个偶数不大于1000，不小于2）

输出格式

　　要求程序输出一个整数，表示老师需要补发的糖果数。

样例输入

3  
2 2 4

样例输出

4

本题的C++参考代码如下：

#include <iostream>

using namespace std;

int a[100]; // 小朋友人数<100

bool notEqual(int n)

{

for (int i = 1; i < n; ++i)

if (a[0] != a[i])

return true;

return false;

}

int main()

{

int i, n, t, s = 0;

cin >> n;

for (i = 0; i < n; ++i)

cin >> a[i];

while ( notEqual(n) )

{

a[0] /= 2;

t = a[0];

for (i = 1; i < n; ++i)

{

a[i] /= 2;

a[i-1] += a[i];

}

a[n-1] += t;

for (i = 0; i < n; ++i)

if (a[i] % 2) // 奇数个颗糖

++a[i], ++s;

}

cout << s << '\n'; // 补发糖果总数数

return 0;

}

本题的C参考代码如下：

#include <stdio.h>

#include <malloc.h>

int count=0;

void f(int n,int \*m)

{

int j,temp=m[0];

for(j=n-1;j>0;j--)

m[(j+1)%n]+=(m[j]/=2);

m[1]+=temp/2;

m[0]-=temp/2;

for(j=0;j<n;j++)

{

if(m[j]%2!=0)

{

m[j]++;

count++;

}

}

}

int jisuan(int n,int \*m)

{

int j;

for(j=0;j<n-1;j++)

if(m[j]!=m[j+1]) return 0;

return 1;

}

int main()

{

int n,j;

scanf("%d",&n);

int \*m=(int\*)malloc(sizeof(int)\*n);

for(j=0;j<n;j++)

scanf("%d",&m[j]);

do

{

f(n,m);

}

while(jisuan(n,m)!=1);

printf("%d",count);

return 0;

}

本题的Java参考代码如下：

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main

{

static int m,n;

public static void main(String[] args) throws IOException

{

int candy=0;

BufferedReader buf;

String str1,str2;

buf=new BufferedReader(new InputStreamReader(System.in));

str1=buf.readLine();

m=Integer.parseInt(str1);

int b[]=new int[m];

str2=buf.readLine();

String a[]=str2.split(" ");

for(int i=0;i<m;i++)

{

b[i]=Integer.parseInt(a[i]);

}

if(!equal(b))

{

while(true)

{

n=b[m-1];

for(int i=m-1;i>0;i--)

{

b[i]=b[i-1]/2+b[i]/2;

}

b[0]=b[0]/2+n/2;

if(equal(b))

break;

else

{

for(int i=0;i<m;i++)

{

if(b[i]%2!=0)

{

candy++;

b[i]++;

}

}

}

}

}

System.out.println(candy);

}

public static boolean equal(int a[])

{

int num=0;

for(int i=0;i<m-1;i++)

{

if(a[i]==a[i+1])

num++;

}

if(num==m-1)

{

return true;

}

else

return false;

}

}