**买不到的数目**

问题描述

小明开了一家糖果店。他别出心裁：把水果糖包成4颗一包和7颗一包的两种。糖果不能拆包卖。

小朋友来买糖的时候，他就用这两种包装来组合。当然有些糖果数目是无法组合出来的，比如要买 10 颗糖。

你可以用计算机测试一下，在这种包装情况下，最大不能买到的数量是17。大于17的任何数字都可以用4和7组合出来。

本题的要求就是在已知两个包装的数量时，求最大不能组合出的数字。

输入格式

两个正整数，表示每种包装中糖的颗数(都不多于1000)

输出格式

一个正整数，表示最大不能买到的糖数

样例输入1

4 7

样例输出1

17

样例输入2

3 5

样例输出2

7

锦囊1

数论或动态规划。

本题的C++参考代码如下：

//2013第四届蓝桥杯全国软件大赛省级选拔赛C/C++A组第8题

//天农计算机系许晓华老师出品

#include<iostream>

using namespace std;

int main()

{

int a,b;

cin>>a>>b;

cout<<a\*b-a-b<<endl;

return 0;

}

本题的C参考代码如下：

#include<stdio.h>

#define MAXSIZE 1000

int main() {

int flag[MAXSIZE] = {0};

int maxunuse[MAXSIZE] = {0};

int num1, num2, minNum, maxNum, temp, i, count = 0;

scanf("%d%d", &num1, &num2);

minNum = ((num1 < num2) ? num1 : num2);

maxNum = num1 + num2 - minNum;

for(i = 1; ;i++){

temp = i \* maxNum % minNum;

if((temp) && flag[temp] == 0){

flag[temp] = 1;

maxunuse[temp] = i \* maxNum -minNum;

count++;

if(count == minNum - 1){

break;

}

}

}

printf("%d\n", maxunuse[temp]);

return 0;

}

本题的Java参考代码如下：

import java.io.BufferedReader;

import java.io.InputStreamReader;

public class Main {

public static void main(String[] args) throws Throwable {

BufferedReader buf = new BufferedReader(

new InputStreamReader(System.in));

String strNum = buf.readLine();

String[] num = strNum.split(" ");

buf.close();

int a, b;

a = Integer.valueOf(num[0]);

b = Integer.valueOf(num[1]);

if (a > b) {

int tem = a;

a = b;

b = tem;

}

int c = a \* b;

int tem = c;

while (tem > 0) {

if (tem % a == 0)

tem = --c;

else if (tem % b == 0)

tem = --c;

else

tem -= b;

}

System.out.println(c);

}

}