

**Московский государственный университет имени М. В. Ломоносова**  
**Факультет вычислительной математики и кибернетики**  
**Олимпиада по прикладной математике и информатике для школьников**  
**31 марта 2012 г.**

В задачах №№1, 2 разрешается пользоваться любой моделью алгоритма (программа на языке программирования, блок-схема или словесное описание). Помните, что **правильный алгоритм без обоснования его корректности засчитывается не полностью!**

Для экономии места решения задач №№1, 2 (для всех классов) записаны на языке программирования Python, главная визуальная особенность которого — в отсутствии явных операторных скобок. Группа операторов с одинаковым отступом является блоком, например, телом цикла, условного оператора или определения функции.

В Python котором есть встроенные типы «строка», «список» и «словарь». Список содержит индексированные данные любых типов (в том числе может содержать списки и словари); обращение к элементу списка производится по индексу, например *Spisok*[5]. Операция *Spisok*[*Nachalo* : *Konec*] возвращает подсписок (не включая элемент с индексом *Konec*), при этом оба параметра можно опускать, чтобы подсписок выбирался с нулевого и/или до последнего элемента соответственно.

Словарь — структура данных, также хранящая данные любых типов, но при этом в роли индекса выступают константы любых типов; например, *Slovar*["*AaBb*"], *Slovar*[3] и т. п.

Строка — неизменяемый список символов, который для удобства записывается в виде

```
soderzhimoe_stroki
```

“. Python имеет множество встроенных строковых методов, в частности, вызов вида *Stroka.split()* возвращает список слов, из которых состояла строка.

Более подробные сведения о языке программирования Python можно получить, например, из учебника: [http://ru.wikibooks.org/wiki/Учебник\\_Python\\_2.6](http://ru.wikibooks.org/wiki/Учебник_Python_2.6).

Прилагаемые программы можно рассматривать в отрыве от конкретного языка программирования: это своего рода компактный псевдокод, определяющий реализацию алгоритма.

## 8 класс

**№1. Гравицаппа.** Гравицаппу можно изготовить из нескольких деталей по прилагающейся инструкции. Если соответствующей детали нет, её, в свою очередь, можно изготовить из других деталей по другому разделу той же инструкции. Известно, что для любой пары деталей  $\langle k, i \rangle$  верно следующее: если для изготовления детали  $i$  используется деталь  $k$ , то для изготовления детали  $k$  не может использоваться деталь  $i$ . Написать программу, которая проверяет, можно ли из имеющегося набора деталей изготовить гравицаппу (при анализе должно учитываться количество требуемых деталей).

**Ввод:** Первая строка — набор имеющихся деталей (разделённые пробелом натуральные числа, необязательно различные, в диапазоне от 1 до 999 числом не более 10000). Вторая и последующие строки содержат правила из инструкции вида «ЦелеваяДеталь Деталь1 Деталь2 ...», задающие способ сборки ЦелевойДетали. Последняя строка определяет правило сборки самой гравицанпы в виде «0 Деталь1 Деталь2 ...» (ноль вместо номера детали). Каждая составная деталь собирается единственным способом.

**Вывод:** Yes или No

**Пример: Ввод:**

```
2 3 5 5 5 6 7 7 7 8 8 8 9 9
4 3 5 5
3 6 7 8 8
2 7 9 7
1 2
0 1 2 3 4
```

**Вывод:** Yes

**Ввод:**

```
4 4 4 4 6
4 6 6 6
2 4 4
0 2 4 6 6
```

**Вывод:** No

**Решение.** Первый пришедший в голову алгоритм: взять список деталей для гравицанпы (последняя строка ввода) и выкидывать из него имеющиеся в наборе детали, а если некоторой детали в наборе нет, подставить в список вместо неё её составляющие части. Если список опустеет — гравицанпа собрана. Если же некоторой детали нет в наборе (никогда не было или кончились), и собрать её нельзя, потому что она несборная, гравицанпу собрать нельзя.

В таком алгоритме не учитывается, что некоторую деталь можно собрать из других несколькими *принципиально различными* способами. Если же способ сборки любой составной детали только один, а в наборе присутствуют как составные детали, так и их части, не имеет значения, в какой момент будет использована составная деталь из набора, а в какой — она же, но собранная из частей. Главное, чтобы при наличии в наборе составной детали, использовалась именно она, а не части (которые, возможно понадобятся для *деталей другого вида*).

```
#!/usr/bin/python
```

```
# coding: utf8
```

```
List=raw_input().split()
```

```
# Набор имеющихся деталей
```

```
Rules={}
```

```
# Словарь вида R[ЦелеваяДеталь]=[Деталь1 Деталь2 ...]
```

```

while True:
    Rule=raw_input().split()
    Rules[Rule[0]]=Rule[1:]
    if Rule[0] == "0": break
print Rules

Work=Rules["0"]
while Work:
    Detail=Work.pop()
    if Detail in List:
        List.remove(Detail)
        continue
    if Detail not in Rules:
        print "No"
        break
    Work.extend(Rules[Detail])
else:
    print "Yes"

```

**№2. Цепь.** Цепь состоит из  $N$  прямолинейных отрезков различной целой длины, соединённых попарно в замкнутую ломаную. Написать программу, определяющую, можно ли из этих отрезков составить квадрат (при этом цепь размыкать нельзя, сгибать и ломать отрезки нельзя, все отрезки цепи без исключения должны войти в квадрат без наложений).

**Ввод:** Число  $N$ , на следующей строке —  $N$  целых чисел — длин отрезков цепи.

**Вывод:** **Yes** или **No**, в зависимости от того, можно или нельзя составить соответствующую фигуру.

**Пример 1: Ввод:**

```

6
1 3 2 2 4 4

```

**Вывод:** Yes

**Пример 2: Ввод:**

```

5
1 3 2 3 3

```

**Вывод:** No

**Решение.** Единственный «подводный камень» этой задачи — соображение о том, что первое и последнее звено цепи не обязательно должны замыкаться на углу фигуры. В самом деле, цепь вида 2, 4, 4, 4, 2 образует квадрат, при этом первое и последнее звено встречаются в середине стороны.

Рассмотрим решение задачи на примере прямоугольника (случай квадрата аналогичен). Первая сторона может начинаться почти где угодно (правда, её длина должна быть строго меньше полупериметра). Вторая сторона должна в сумме с первой составлять полупериметр, третья — быть равной первой (о четвёртой стороне заботиться уже не надо). Все стороны могут состоять из одного или более звеньев цепи.

Таким образом, внешний цикл — перебор всех возможных начал первой стороны, внутренний — перебор возможных длин первой стороны. Внутри циклов надо проверить, можно ли сложить вторую сторону длиной *Poluperimetr - PervayaStorona* и третью длиной *PervayaStorona*.

```
# !/usr/bin/python
# coding: utf8

def locate(Lnk, Start, Length): # (определение функции)
    '''Проверить, можно ли в цепи Lnk найти последовательность звеньев,
    начиная со звена Start, суммарной длины Length.
    Вернуть начало следующей стороны, в случае неуспеха - 0.'''
    h,D=Lnk[Start],Start
    while h<Length and D<N:
        D+=1
        h+=Lnk[D]
    if h==Length: return D+1
    return 0

def check(Lnk): # (определение функции)
    '''Проверить, можно ли из цепи Lnk составить прямоугольник'''
    if Perimeter%2 != 0:          # Прямоугольник не получится
        return False
    for i in xrange(N-3):        # Начало первой стороны
        L=(Lnk*2)[i:i+N]         # Переставим предыдущие звенья цепи в конец
        w,C=0,0                  # Ширина первой стороны, следующее звено
        while w<Perimeter/2:
            w+=L[C]              # Добавляем длину звена к стороне
            C+=1                 # Выбираем следующее звено
            D=locate(L,C,Perimeter/2-w) # Находим вторую сторону
            if D==0: continue    # Не нашли
            E=locate(L,D,w)      # Находим третью сторону
            if E==0: continue
            return True          # Четвёртую сторону искать не надо :)
    return False

N=input()
Links=[int(n) for n in raw_input().split()] # каждое слово ввода превратим в число
Perimeter=sum(Links)
if check(Links): print "Yes"
else: print "No"
}
```

**№3. Отрезки.** На сторонах треугольника  $ABC$  вовне треугольника построены квадраты  $ABB_1A_2$ ,  $BCC_1B_2$ ,  $CAA_1C_2$ . Доказать, что сумма длин отрезков  $BA_1$  и  $BC_2$  равна сумме длин отрезков  $CA_2$  и  $AC_1$ .

**Решение.** Существует такой поворот плоскости относительно точки  $A$  на  $90^\circ$ , при котором точка  $A_2$  переходит в точку  $B$ , а точка  $C$  переходит в точку  $A_1$ . Значит, длины отрезков  $CA_2$  и  $BA_1$  равны. Это можно было бы установить и из равенства треугольников  $\triangle AA_2C$  и  $\triangle ABA_1$  по двум сторонам  $AA_2 = AB$ ,  $AC = AA_1$  и углу между ними:

$$\angle A_2AC = \angle A_2AB + \angle BAC = 90^\circ + \angle BAC = \angle BAC + \angle CAA_1 = \angle BAA_1.$$

Аналогично устанавливается, что  $AC_1 = BC_2$ , откуда и вытекает требуемое.

**№4. Суммы квадратов.** Натуральные числа  $a$  и  $b$  таковы, что  $(a + b)/2 = 2012$ , причем  $a = x_1^2 + x_2^2 + x_3^2$  и  $b = x_4^2 + x_5^2 + x_6^2$ . Известно также, что все числа  $x_i$  натуральны и попарно различны. В ответе указать пример такой шестерки чисел  $x_1, x_2, x_3, x_4, x_5, x_6$ .

**Решение.** Поскольку  $7^2 + 21^2 + 39^2 = 2011$ ,  $9^2 + 29^2 + 33^2 = 2011$ ,  $21^2 + 27^2 + 29^2 = 2011$ , и  $2^2 + 28^2 + 35^2 = 2013$ ,  $4^2 + 29^2 + 34^2 = 2013$ ,  $8^2 + 10^2 + 43^2 = 2013$ ,  $13^2 + 20^2 + 38^2 = 2013$  (указаны все возможности с точностью до перестановки слагаемых для  $a = 2011$ ,  $b = 2013$ ), то в качестве примера шестерки можно взять  $(7, 21, 39, 2, 28, 35)$  или  $(7, 21, 39, 8, 10, 43)$ . Для сокращения перебора полезно заметить, что квадрат числа при делении на 4 дает в остатке только 0 или 1. Так как  $2011 \equiv 3 \pmod{4}$ , то в этом случае надо искать квадраты нечетных чисел, так как  $2013 \equiv 1 \pmod{4}$ , то в этом случае надо искать квадраты двух четных и одного нечетного чисел.

**№5. Игра „Попробуй, обнули!“** Имеется выражение

$$|Ax^3 + Bx^2 + Cx + D| \cdot E + Fx^4 + Gy.$$

Двое по очереди выбирают некоторую переменную ( $A, B, C, D, E, F, G, x$  или  $y$ ) и присваивает ей некоторое вещественное значение. Первый выиграет, если в итоге (при подстановке выбранных значений переменных) получится значение 0, иначе выиграет второй. Кто выигрывает при правильной игре? Какова стратегия этого игрока?

**Ответ.** Выигрывает первый игрок.

**Решение.** Разобьем буквы на пары  $(A, B)$ ,  $(C, F)$ ,  $(D, E)$ ,  $(G, y)$ . Первый игрок первым ходом присваивает  $x$  значение 0. Далее на выбор вторым игроком значения одной из переменных первый присваивает нулевое значение второму элементу той же пары. Значение выражения в итоге неизбежно станет равным  $|A \cdot 0 + B \cdot 0 + C \cdot 0 + D| \cdot E + F \cdot 0 + Gy = |D| \cdot E + Gy = 0$ , то есть выиграет первый игрок.

**№6. Раскраски правильного тетраэдра.** Сколько существует попарно неравных правильных единичных тетраэдров, у которых каждая грань и каждое ребро раскрашены в один из 4 цветов (красный, оранжевый, желтый, зеленый) так, что для любой пары граней цвета этих граней и цвет ребра, разделяющего эти грани, попарно различны?

Тетраэдры считаются равными, если их можно совместить с полным совпадением цветов и граней, и ребер.

**Ответ.** 128.

**Решение.** Легко видеть, что любые две грани тетраэдра смежны, а всего граней — 4. Значит, для раскраски граней будут использованы все 4 цвета. Выберем в качестве основания тетраэдра плоскость красного цвета, тогда боковые грани будут окрашены в оранжевый, желтый и зеленый цвета либо по часовой, либо против часовой стрелки (всего имеется две различных раскраски граней). Осталось каждое ребро раскрасить в цвет, отличный от цветов содержащих ребро граней — имеется 2 способа окраски каждого ребра, и их можно выбирать независимо для разных ребер. Так как общее число ребер тетраэдра равно 6, то в итоге получаем  $2 \cdot 2^6 = 128$  различных раскрасок.

**№7. Кто есть кто?** На факультете учатся 2012 студентов. Каждый из них либо рыцарь (всегда говорит правду), либо лжец (всегда лжет). Декан факультета может за один шаг выбрать произвольного студента и спросить у него, является ли такой-то студент факультета лжецом. Сможет ли декан факультета выяснить, кто из 2012 студентов факультета рыцарь, а кто — лжец, если а) больше ничего не известно, б) известно, что на факультете рыцарей больше, чем лжецов. Какое наименьшее число шагов гарантированно дает декану необходимую информацию?

**Ответ.** а) Нет. б) Да, 2011 шагов.

**Решение.** Нетрудно заметить, что студенты с номерами  $i$  и  $j$  дадут друг о друге одинаковые ответы декану: „нет“, если они оба лжецы или оба рыцари, и „да“, если один из них лжец, а другой — рыцарь. Поэтому декану нет смысла задавать одной и той же паре студентов больше одного вопроса. Более того, при отсутствии у декана дополнительной информации всегда будет иметься (по крайней мере) два противоположных способа распределения меток „рыцарь“, „лжец“ по всем студентам, и, следовательно, в пункте а) ответ отрицательный. В пункте б) ответ положительный: декан может спросить мнение одного студента обо всех остальных, и, сравнив количества ответов „да“ и ответов „нет“, определить (зная, что ответов про рыцарей хотя бы на 1 больше), рыцарю или лжецу задавал он вопросы и, следовательно, правдивыми или лживыми были все ответы. Заметим, что при этом декану понадобится сделать 2011 шагов. Покажем, что это и есть минимальное количество шагов, гарантированно дающее декану необходимую информацию. Пусть декан задал студентам некоторое количество вопросов. Назовем компонентой связности каждое максимальное по включению студентов множество студентов, о которых у декана уже есть зависимая информация (так, например, если декан задал вопрос одному из студентов  $i, j$  о другом и задал вопрос одному из студентов  $i, k$  о другом, то студенты  $i, j, k$  оказываются в одной компоненте связности). В начальный момент (пока декан не задавал никому вопросов), имеется 2012 компонент связности, содержащих каждая по одному студенту. С каждым вопросом декан уменьшает количество компонент связности не более чем на 1. Значит за менее чем 2011 вопросов он уменьшит число компонент связности не более чем на 2010, и число компонент связности будет не меньше 2. Рассмотрим следующую ситуацию. Пусть на каждый вопрос, после которого имеется не менее трех компонент связности, декан слышит ответ „нет“. Тогда, получив ответ на этот вопрос, декан не сможет в точности сказать, кто из студентов

рыцарь, а кто — лжец, так как в минимальной по числу студентов компоненте связности все студенты могут быть как рыцарями, так и лжецами. Рассмотрим теперь такой вопрос декана, после которого имеется в точности две компоненты связности. Ясно, что до этого вопроса имелось в точности три компоненты связности, и после этого вопроса студенты каких-то двух компонент связности  $K_1$  и  $K_2$  (из трех имеющихся компонент связности) оказались в одной компоненте связности  $K$ . Пусть в  $K$  не менее 1007 студентов. Тогда, услышав на свой последний вопрос ответ „нет“, декан не сможет определить, являются ли все студенты из отличной от  $K$  компоненты рыцарями или они все являются лжецами (оба варианта возможны, так как в каждом число лжецов меньше числа рыцарей). Пусть теперь в  $K$  не более 1006 студентов. Тогда, услышав на свой последний вопрос ответ „да“, декан не сможет определить, являются ли все студенты из множества  $K_1$  рыцарями, а все студенты из множества  $K_2$  — лжецами, или же все студенты из множества  $K_2$  являются рыцарями, а все студенты из множества  $K_1$  — лжецами (оба варианта возможны, так как в каждом число лжецов меньше числа рыцарей). Таким образом, задав не более 2010 вопросов, декан не сможет во всех случаях определить, кто из студентов лжец, а кто — рыцарь, и 2011 вопросов и составляют искомое минимальное число шагов, гарантирующее декану успех.

**Московский государственный университет имени М. В. Ломоносова**  
**Факультет вычислительной математики и кибернетики**  
**Олимпиада по прикладной математике и информатике для школьников**  
**31 марта 2012 г.**  
**9 класс**

В задачах №№1, 2 разрешается пользоваться любой моделью алгоритма (программа на языке программирования, блок-схема или словесное описание). Помните, что **правильный алгоритм без обоснования его корректности засчитывается не полностью!**

**№1. Гравицаппа.** Гравицаппу можно изготовить из нескольких деталей по прилагающейся инструкции. Если соответствующей детали нет, её, в свою очередь, можно изготовить из других деталей по другому разделу той же инструкции. Известно, что для любой пары деталей  $\langle k, i \rangle$  верно следующее: если для изготовления детали  $i$  используется деталь  $k$ , то для изготовления детали  $k$  не может использоваться деталь  $i$ . Написать программу, которая проверяет, можно ли из имеющегося набора деталей изготовить гравицаппу (при анализе должно учитываться количество требуемых деталей).

**Ввод:** Первая строка — набор имеющихся деталей (разделённые пробелом натуральные числа, необязательно различные, в диапазоне от 1 до 999 числом не более 10000). Вторая и последующие строки содержат правила из инструкции вида «ЦелеваяДеталь Деталь1 Деталь2 . . .», задающие способ сборки ЦелевойДетали. Последняя строка определяет правило сборки самой гравицаппы в виде «0 Деталь1 Деталь2 . . .» (ноль вместо номера детали). Каждая составная деталь собирается единственным способом.

**Вывод:** Yes или No

**Пример: Ввод:**

```
2 3 5 5 5 6 7 7 7 8 8 8 9 9
4 3 5 5
3 6 7 8 8
2 7 9 7
1 2
0 1 2 3 4
```

**Вывод: Yes**

**Ввод:**

```
4 4 4 4 6
4 6 6 6
2 4 4
0 2 4 6 6
```



**Вывод:** No

**Решение:** см. решение задачи №1 для 8 класса.

**№2. Цепь.** Цепь состоит из  $N$  прямолинейных отрезков различной целой длины, соединённых попарно в замкнутую ломаную. Написать программу, определяющую, можно ли из этих отрезков составить прямоугольник (при этом цепь размыкать нельзя, сгибать и ломать отрезки нельзя, все отрезки цепи без исключения должны войти в прямоугольник без наложений).

**Ввод:** Число  $N$ , на следующей строке —  $N$  целых чисел — длин отрезков цепи.

**Вывод:** Yes или No, в зависимости от того, можно или нельзя составить соответствующую фигуру.

**Пример 1: Ввод:**

6  
1 3 2 3 4 5

**Вывод:** Yes

**Пример 2: Ввод:**

5  
1 3 2 3 3

**Вывод:** No

**Решение:** см. решение задачи №2 для 8 класса.

**№3. Отрезки.** На сторонах треугольника  $ABC$  вовне треугольника построены квадраты  $ABB_1A_2$ ,  $BCC_1B_2$ ,  $CAA_1C_2$ . Доказать, что сумма длин отрезков  $BA_1$  и  $BC_2$  равна сумме длин отрезков  $CA_2$  и  $AC_1$ .

**Решение:** см. решение задачи №3 для 8 класса.

**№4. Суммы квадратов.** Натуральные числа  $a$  и  $b$  таковы, что  $(a + b)/2 = 2012$ , причем  $a = x_1^2 + x_2^2 + x_3^2$  и  $b = x_4^2 + x_5^2 + x_6^2$ . Известно также, что все числа  $x_i$  натуральны и попарно различны. В ответе указать пример такой шестерки чисел  $x_1, x_2, x_3, x_4, x_5, x_6$ .

**Решение:** см. решение задачи №4 для 8 класса.

**№5. Раскраски куба.** Сколько существует попарно неравных правильных единичных кубов, у которых каждая грань и каждое ребро раскрашены в один из 4 цветов так, что для любой пары смежных граней цвета этих граней и цвет ребра, разделяющего эти грани, попарно различны? Кубы считаются равными, если их можно совместить с полным

совпадением цветов и граней, и ребер. Для раскраски граней обязательно использовать все 4 цвета

**Ответ.** 12480.

**Решение.** По условию для раскраски граней использованы все 4 цвета. Т. к. каждого цвета не более 2 граней (для каждой грани только одна грань с ней не смежна), получаем, что имеется по 2 грани каких-то двух цветов (всегда можно положить куб на горизонтальную плоскость так, чтобы они были боковыми) и по одной грани оставшихся двух цветов (они тогда будут сверху и снизу). Понятно, что таких раскрасок будет ровно столько, сколько способов выбрать 2 цвета из 4-х, т.е. ровно 6 штук.

Фиксируем раскраску граней. И положим куб так, как указано выше. Заметим, что раскраска граней переходит в себя при повороте на  $180^\circ$  относительно вертикальной оси симметрии куба. Раскрасим правильным образом 4 ребра передней грани и 2 горизонтальных ребра левой боковой грани. Таких раскрасок ровно  $2^6$ , т. к. каждое ребро можно раскрасить в один из двух цветов, „не занятых“ инцидентными гранями. При указанном повороте эти 6 ребер переходят в оставшиеся 6 ребер куба. Назовем раскраску симметричной, если при таком повороте раскраска ребер куба переходит в себя. Очевидно, имеется  $2^6$  симметричных раскрасок ребер и  $2^6 \cdot (2^6 - 1)/2$  несимметричных раскрасок ребер.

Итак, всего имеется различных раскрасок граней и ребер куба, удовлетворяющих условию задачи, ровно  $6 \cdot (2^6 + 2^6 \cdot (2^6 - 1)/2) = 6 \cdot 2^5 \cdot (2 + 2^6 - 1) = 6 \cdot 32 \cdot 65 = 12480$ .

**№6. Уравнение в целых числах.** При каждом значении параметра  $a$  ( $a \in \mathbf{R}$ ) решить в целых числах уравнение

$$x^3 + 3a^2x + 3a^2 - 5a = 0.$$

**Ответ.** Если  $a = -\frac{1}{5}$ , то  $x = -1$ , если  $a = 0$  или  $a = \frac{5}{3}$ , то  $x = 0$ , если  $a = \frac{1}{3}$  или  $a = 12$ , то  $x = 1$ , при других  $a$  целочисленных решений нет.

**Решение.** Перегруппируем уравнение по степеням  $a$ :

$$(3x + 3)a^2 - 5a + x^3 \tag{1}$$

Рассмотрим два случая.

1)  $x = -1$ , уравнение (1) — линейное,  $a = -\frac{1}{5}$ .

2)  $x \neq -1$ , уравнение (1) — квадратное. Его дискриминант должен быть неотрицателен:

$$D = 25 - 12x^4 - 12x^3 \geq 0,$$

$$x^4 + x^3 \leq \frac{25}{12}. \tag{2}$$

Обозначим  $f(x) = x^4 + x^3$ ,  $g(x) = f(-x) = x^4 - x^3$ .

Легко видеть, что функция  $f(x)$  строго возрастает при неотрицательных целых  $x$ , так что среди таких  $x$  условию (2) удовлетворяют лишь  $x = 0$ ,  $x = 1$ .

Функция  $g(x)$  строго возрастает при натуральных  $x$ :

$$g(x + 1) - g(x) = 4x^3 + 3x^2 + x > 0.$$

Поэтому функция  $f(x)$  строго убывает при отрицательных целых  $x$ , так что среди таких  $x$  условию (2) удовлетворяет лишь  $x = -1$ , но это значение было исключено в рассматриваемом случае.

Подставим  $x = 0$  в уравнение (1):  $3a^2 - 5a = 0$ ,  $a \in \{0, \frac{5}{3}\}$ .

Подставим  $x = 1$  в уравнение (1):  $6a^2 - 5a + 1 = 0$ ,  $a \in \{\frac{1}{3}, \frac{1}{2}\}$ .

Выписывая найденные целочисленные решения в зависимости от параметра  $a$ , получаем ответ.

**№7. Кто есть кто?** На факультете учатся 2012 студентов. Каждый из них либо рыцарь (всегда говорит правду), либо лжец (всегда лжет). Декан факультета может за один шаг выбрать произвольного студента и спросить у него, является ли такой-то студент факультета лжецом. Сможет ли декан факультета выяснить, кто из 2012 студентов факультета рыцарь, а кто - лжец, если а) больше ничего не известно, б) известно, что на факультете рыцарей больше, чем лжецов. Какое наименьшее число шагов гарантированно дает декану необходимую информацию?

**Решение:** см. решение задачи №7 для 8 класса.

**Московский государственный университет имени М. В. Ломоносова**  
**Факультет вычислительной математики и кибернетики**  
**Олимпиада по прикладной математике и информатике для школьников**  
**31 марта 2012 г.**  
**10 класс**

В задачах №№1, 2 разрешается пользоваться любой моделью алгоритма (программа на языке программирования, блок-схема или словесное описание). Помните, что **правильный алгоритм без обоснования его корректности засчитывается не полностью!**

**№1. Гравицаппа.** Гравицаппу можно изготовить из нескольких деталей по прилагающейся инструкции. Если соответствующей детали нет, её, в свою очередь, можно изготовить из других деталей по другому разделу той же инструкции. Известно, что для любой пары деталей  $\langle k, i \rangle$  верно следующее: если для изготовления детали  $i$  используется деталь  $k$ , то для изготовления детали  $k$  не может использоваться деталь  $i$ . Написать программу, которая проверяет, можно ли из имеющегося набора деталей изготовить гравицаппу (при анализе должно учитываться количество требуемых деталей).

**Ввод:** Первая строка — набор имеющихся деталей (разделённые пробелом натуральные числа, необязательно различные, в диапазоне от 1 до 999 числом не более 10000). Вторая и последующие строки содержат правила из инструкции вида «ЦелеваяДеталь Деталь1 Деталь2 . . .», задающие способ сборки ЦелевойДетали. Последняя строка определяет правило сборки самой гравицаппы в виде «0 Деталь1 Деталь2 . . .» (ноль вместо номера детали). Каждая составная деталь собирается единственным способом.

**Вывод:** Yes или No

**Пример: Ввод:**

```
2 3 5 5 5 6 7 7 7 8 8 8 9 9
4 3 5 5
3 6 7 8 8
2 7 9 7
1 2
0 1 2 3 4
```

**Вывод:** Yes

**Ввод:**

```
4 4 4 4 6
4 6 6 6
2 4 4
0 2 4 6 6
```

**Вывод:** No

**Решение:** см. решение задачи №1 для 8 класса.

**№2. Цепь.** Цепь состоит из  $N$  прямолинейных отрезков различной целой длины, соединённых попарно в замкнутую ломаную. Написать программу, определяющую, можно ли из этих отрезков составить прямоугольник (при этом цепь размыкать нельзя, сгибать и ломать отрезки нельзя, все отрезки цепи без исключения должны войти в прямоугольник без наложений).

**Ввод:** Число  $N$ , на следующей строке —  $N$  целых чисел — длин отрезков цепи.

**Вывод:** Yes или No, в зависимости от того, можно или нельзя составить соответствующую фигуру.

**Пример 1: Ввод:**

6  
1 3 2 3 4 5

**Вывод:** Yes

**Пример 2: Ввод:**

5  
1 3 2 3 3

**Вывод:** No

**Решение:** см. решение задачи №2 для 8 класса.

**№3. Отрезки.** На сторонах выпуклого четырехугольника  $ACEG$  вовне четырехугольника построены правильные треугольники  $ABC$ ,  $CDE$ ,  $EFG$ ,  $GHA$ . Доказать, что суммарные длины ломаных  $HCFAD$  и  $DGBEH$  равны.

**Решение.** Существует такой поворот плоскости относительно точки  $A$  на  $60^\circ$ , при котором точка  $B$  переходит в точку  $C$ , а точка  $G$  переходит в точку  $H$ . Значит, длины отрезков  $BG$  и  $CH$  равны. Это можно было бы установить и из равенства треугольников  $\triangle ABG$  и  $\triangle ACH$  по двум сторонам  $AB = AC$ ,  $AG = AH$  и углу между ними:

$$\angle BAG = \angle BAC + \angle CAG = 60^\circ + \angle CAG = \angle CAG + \angle GAH = \angle CAH.$$

Аналогично устанавливается, что  $BE = AD$ ,  $DG = CF$ ,  $HE = AF$ , откуда и вытекает требуемое.

**№4. Игра „Попробуй, обнули!“** Имеется выражение

$$A \sin ax + B \cos bx + Cx + D.$$

Двое по очереди выбирают некоторую переменную ( $A$ ,  $a$ ,  $B$ ,  $b$ ,  $C$ ,  $D$  или  $x$ ) и присваивает ей некоторое вещественное значение. Первый выигрывает, если в итоге (при подстановке

выбранных значений переменных) получится значение 0, иначе выиграет второй. Кто выигрывает при правильной игре? Какова стратегия этого игрока?

**Ответ.** Выигрывает первый игрок.

**Решение.** Разобьем буквы на пары  $(A, a)$ ,  $(b, C)$ ,  $(B, D)$ . Первый игрок первым ходом присваивает  $x$  значение 0. Далее на выбор вторым игроком значения одной из переменных первый присваивает противоположное значение второму элементу той же пары. Поскольку  $\cos 0 = 1$ , значение выражения в итоге неизбежно станет равным  $A \cdot 0 + B \cdot 1 + C \cdot 0 + D = B + D = 0$ , то есть выиграет первый игрок.

**№5. Странное число.** Доказать, что расстояние между точками с только лишь целочисленными координатами на плоскости не может быть равно  $\sqrt{20122012\dots 2012}$  (под радикалом ровно 2012 цифр).

**Решение.** Предположим противное: найдутся точки декартовой плоскости  $A(x, y)$  и  $B(x', y')$  с только лишь целочисленными координатами такие, что  $|AB| = \sqrt{20122012\dots 2012}$  (под радикалом ровно 2012 цифр). Тогда  $(x - x')^2 + (y - y')^2 = \underbrace{20122012\dots 2012}_{2012 \text{ цифр}}$  (обозначим

число в правой части через  $m$ ), то есть  $p^2 + q^2 = m$  ( $p, q$  — целые). Легко видеть, что каждое целое число  $n$  можно представить в виде  $n = 16k \pm r$ , где  $r \in \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ ,  $k \in \mathbf{Z}$ . Тогда ясно, что определяемый по  $n$  однозначно остаток по делению на 16 числа  $n^2$  равен остатку по делению на 16 числа  $r^2$ , то есть лежит во множестве чисел  $M_{16} = \{0, 1, 4, 9\}$ . Следовательно, сумма двух квадратов целых чисел может давать при делении на 16 только такие остатки, какие встречаются среди остатков по делению на 16 сумм двух не обязательно различных чисел из множества  $M_{16}$ . Но среди таких остатков нет числа 12, а именно таков остаток по делению на 16 числа  $m$  (это легко следует из того, что число  $10000 = 5^4 \cdot 2^4$  кратно числу  $16 = 2^4$ , а число 2012 имеет остаток 12 по делению на 16). Получили противоречие. Утверждение доказано.

**№6. Кто есть кто?** На факультете учатся 2012 студентов. Каждый из них либо рыцарь (всегда говорит правду), либо лжец (всегда лжет). Декан факультета может за один шаг выбрать произвольного студента и спросить у него, является ли такой-то студент факультета лжецом. Сможет ли декан факультета выяснить, кто из 2012 студентов факультета рыцарь, а кто — лжец, если а) больше ничего не известно, б) известно, что на факультете рыцарей больше, чем лжецов. Какое наименьшее число шагов гарантированно дает декану необходимую информацию?

**Решение:** см. решение задачи №7 для 8 класса.

**№7. Четверки.** Все упорядоченные четверки попарно различных элементов множества  $\{0, 1, 2, 3\}$  разбили на 12 пар так, что в каждой паре оказались четверки вида  $(a, b, c, d)$  и  $(d, c, b, a)$ . Предлагается выбрать из каждой пары ровно одну четверку и занумеровать выбранные 12 четверок числами  $0, 1, \dots, 11$  в таком порядке, чтобы для всех целых  $n$  и для всех  $k = 4m + r$  ( $m$  целое,  $r = 0, 1, 2, 3$ ) нашлась перестановка  $s(r, n)$  номеров мест элементов четверки (зависящая от  $r$  и  $n$  и не зависящая от  $m$ ) такая, что четверка с номером  $k + n \bmod 12$  получалась бы из четверки с номером  $k \bmod 12$  с помощью перестановки  $s(r, n)$ . (Через  $k \bmod 12$  обозначается остаток от деления числа  $k$

на 12, то есть такое целое число  $t$  от 0 до 11, что для некоторого целого  $j$  верно равенство  $k = 12j + t$ ).

**Один из возможных вариантов ответа:**

0: (0, 1, 2, 3),

1: (1, 2, 3, 0),

2: (2, 1, 3, 0),

3: (1, 3, 0, 2),

4: (1, 3, 2, 0),

5: (3, 2, 0, 1),

6: (2, 3, 0, 1),

7: (3, 0, 1, 2),

8: (3, 0, 2, 1),

9: (0, 2, 1, 3),

10: (2, 0, 1, 3),

11: (0, 1, 3, 2).

Здесь вторая четверка получается из первой циклическим сдвигом влево, третья из второй — транспозицией (переменой мест друг с другом) первых двух элементов, четвертая четверка из третьей — опять циклическим сдвигом влево, пятая четверка из четвертой — транспозицией последних двух элементов, и далее процесс повторяется, причем бесконечно долго, если считать, что после 12-й четверки опять следует первая.