

## Олимпиада «Ломоносов» по информатике 2013-2014. Третий отборочный тур.

### Задача 0 (входная). (1 балл)

#### Вариант 1.

Переведите число 51d23d из шестнадцатеричной системы счисления в восьмеричную.

Ответ: 24351075

#### Вариант 2.

Переведите число 345240 из шестнадцатеричной системы счисления в восьмеричную.

Ответ: 15051100

#### Вариант 3.

Переведите число ff7187 из шестнадцатеричной системы счисления в восьмеричную.

Ответ: 77670607

#### Вариант 4.

Переведите число 7ba707 из шестнадцатеричной системы счисления в восьмеричную.

Ответ: 36723407

### Задача 1. (11 баллов)

#### Вариант 1.

Рассмотрим смешанную систему счисления, которая используется для регистрационных номеров транспортных средств ГИБДД. Номер имеет вид: LDDDLL, где D – это десятичная цифра, L – это 12 заглавных букв, графические представления которых похожи в кириллице и латинице, например, У и Y. Буквы мы будем считать упорядоченными как в латинице (A, B, C, ...). Число записывается справа налево, то есть справа записывается младший разряд, а слева – старший разряд. Даны два числа, записанные в этой системе счисления. Выполните их перемножение и результат запишите в той же самой системе счисления. Если результат требует больше 6 разрядов, сохраните только младшие 6 разрядов, а старшие отбросьте.

A071HY\*A043YA = ?

Ответ: O595BA

#### Вариант 2.

Рассмотрим смешанную систему счисления, которая используется для регистрационных номеров транспортных средств ГИБДД. Номер имеет вид: LDDDLL, где D – это десятичная цифра, L – это 12 заглавных букв, графические представления которых похожи в кириллице и латинице, например, У и Y. Буквы мы будем считать упорядоченными как в латинице (A, B, C, ...). Число записывается справа налево, то есть справа записывается младший разряд, а слева – старший разряд. Даны два числа, записанные в этой системе счисления. Выполните их перемножение и результат запишите в той же самой системе счисления. Если результат требует больше 6 разрядов, сохраните только младшие 6 разрядов, а старшие отбросьте.

E220HA\*A000BX = ?

Ответ: X847HA

### Вариант 3.

Рассмотрим смешанную систему счисления, которая используется для регистрационных номеров транспортных средств ГИБДД. Номер имеет вид: LDDDLL, где D – это десятичная цифра, L – это 12 заглавных букв, графические представления которых похожи в кириллице и латинице, например, У и Y. Буквы мы будем считать упорядоченными как в латинице (A, B, C, ...). Число записывается справа налево, то есть справа записывается младший разряд, а слева – старший разряд. Даны два числа, записанные в этой системе счисления. Выполните их перемножение и результат запишите в той же самой системе счисления. Если результат требует больше 6 разрядов, сохраните только младшие 6 разрядов, а старшие отбросьте.

X858KY\*A025BV = ?

Ответ: E735HY

### Вариант 4.

Рассмотрим смешанную систему счисления, которая используется для регистрационных номеров транспортных средств ГИБДД. Номер имеет вид: LDDDLL, где D – это десятичная цифра, L – это 12 заглавных букв, графические представления которых похожи в кириллице и латинице, например, У и Y. Буквы мы будем считать упорядоченными как в латинице (A, B, C, ...). Число записывается справа налево, то есть справа записывается младший разряд, а слева – старший разряд. Даны два числа, записанные в этой системе счисления. Выполните их перемножение и результат запишите в той же самой системе счисления. Если результат требует больше 6 разрядов, сохраните только младшие 6 разрядов, а старшие отбросьте.

A163PE\*A294AT = ?

Ответ: O347CE

## Задача 2. (14 баллов)

### Вариант 1.

Ровно в полночь первого января 2014 года по местному времени из некоторого аэропорта стартовал перелет в другой аэропорт с тремя промежуточными остановками. График перелета приведен ниже.

	Arrival	Departure
PANC		2014-01-01 00:00:00
TNCM	2014-01-01 14:43:00	2014-01-01 16:58:00
EBBR	2014-01-02 06:15:00	2014-01-02 10:45:00
USSS	2014-01-02 20:08:00	2014-01-02 22:38:00
UHWW	2014-01-03 09:43:00	

Все времена – местные. Временем в полете между двумя пунктами считается промежуток времени от вылета из первого пункта до прилета во второй пункт.

Сколько времени суммарно было проведено в полете. Ответ запишите в виде количества минут.

Ответ: 1708

**Вариант 2.**

Ровно в полночь первого января 2014 года по местному времени из некоторого аэропорта стартовал перелет в другой аэропорт с тремя промежуточными остановками. График перелета приведен ниже.

	Arrival	Departure
PHNL		2014-01-01 00:00:00
KMCO	2014-01-01 14:11:00	2014-01-01 16:51:00
EIDW	2014-01-02 05:43:00	2014-01-02 12:33:00
UUEE	2014-01-02 19:53:00	2014-01-02 22:13:00
UEEE	2014-01-03 10:05:00	

Все времена – местные. Временем в полете между двумя пунктами считается промежуток времени от вылета из первого пункта до прилета во второй пункт.

Сколько времени суммарно было проведено в полете. Ответ запишите в виде количества минут.

Ответ: 1575

**Вариант 3.**

Ровно в полночь первого января 2014 года по местному времени из некоторого аэропорта стартовал перелет в другой аэропорт с тремя промежуточными остановками. График перелета приведен ниже.

	Arrival	Departure
KSFO		2014-01-01 00:00:00
KJFK	2014-01-01 08:00:00	2014-01-01 10:25:00
LROP	2014-01-02 02:36:00	2014-01-02 11:16:00
UNNT	2014-01-02 21:14:00	2014-01-02 23:04:00
UHPP	2014-01-03 09:43:00	

Все времена – местные. Временем в полете между двумя пунктами считается промежуток времени от вылета из первого пункта до прилета во второй пункт.

Сколько времени суммарно было проведено в полете. Ответ запишите в виде количества минут.

Ответ: 1488

**Вариант 4.**

Ровно в полночь первого января 2014 года по местному времени из некоторого аэропорта стартовал перелет в другой аэропорт с тремя промежуточными остановками. График перелета приведен ниже.

	Arrival	Departure
PHNL		2014-01-01 00:00:00
BGGH	2014-01-01 17:22:00	2014-01-01 19:57:00
UMMS	2014-01-02 07:13:00	2014-01-02 14:58:00
UNKL	2014-01-03 00:44:00	2014-01-03 02:54:00
RJAA	2014-01-03 08:58:00	

Все времена – местные. Временем в полете между двумя пунктами считается промежуток

времени от вылета из первого пункта до прилета во второй пункт.

Сколько времени суммарно было проведено в полете. Ответ запишите в виде количества минут.

Ответ: 1528

### Задача 3. (15 баллов)

#### Вариант 1.

Очередное  $X_{k+1}$  число последовательности получается из  $X_k$  с помощью формулы  $X_{k+1} = (a X_k + c) \bmod m$ , где  $a, c, m$  - некоторые целочисленные коэффициенты. По известным числам  $a, c, m, X_{k+1}$  найдите  $X_k$ .

$$a = 462648, c = 37248, m = 39916801, X_{k+1} = 4$$

Ответ: 33614466

#### Вариант 2.

Очередное  $X_{k+1}$  число последовательности получается из  $X_k$  с помощью формулы  $X_{k+1} = (a X_k + c) \bmod m$ , где  $a, c, m$  - некоторые целочисленные коэффициенты. По известным числам  $a, c, m, X_{k+1}$  найдите  $X_k$ .

$$a = 37847242, c = 7625, m = 54018521, X_{k+1} = 1232999$$

Ответ: 46686399

#### Вариант 3.

Очередное  $X_{k+1}$  число последовательности получается из  $X_k$  с помощью формулы  $X_{k+1} = (a X_k + c) \bmod m$ , где  $a, c, m$  - некоторые целочисленные коэффициенты. По известным числам  $a, c, m, X_{k+1}$  найдите  $X_k$ .

$$a = 4839, c = 6439277, m = 20996011, X_{k+1} = 42$$

Ответ: 4315890

#### Вариант 4.

Очередное  $X_{k+1}$  число последовательности получается из  $X_k$  с помощью формулы  $X_{k+1} = (a X_k + c) \bmod m$ , где  $a, c, m$  - некоторые целочисленные коэффициенты. По известным числам  $a, c, m, X_{k+1}$  найдите  $X_k$ .

$$a = 23473622, c = 16638284, m = 24036583, X_{k+1} = 88$$

Ответ: 9431306

### Задача 4. (12 баллов)

#### Вариант 1.

Рассмотрим десятичную запись дробной части числа  $\phi$  (золотое сечение). Позиции цифр в нумеруются от 1 (то есть первая цифра дробной части - '6', имеет номер 1). Сколько цифр '7' находится на позициях от 1991 до 2036 включительно.

Ответ: 3

**Вариант 2.**

Рассмотрим десятичную запись дробной части числа  $\phi$  (золотое сечение). Позиции цифр в нумеруются от 1 (то есть первая цифра дробной части – '6', имеет номер 1). Сколько цифр '5' находится на позициях от 1564 до 1611 включительно.

Ответ: 8

**Вариант 3.**

Рассмотрим десятичную запись дробной части числа  $\phi$  (золотое сечение). Позиции цифр в нумеруются от 1 (то есть первая цифра дробной части – '6', имеет номер 1). Сколько цифр '3' находится на позициях от 1901 до 1999 включительно.

Ответ: 7

**Вариант 4.**

Рассмотрим десятичную запись дробной части числа  $\phi$  (золотое сечение). Позиции цифр в нумеруются от 1 (то есть первая цифра дробной части – '6', имеет номер 1). Сколько цифр '9' находится на позициях от 2001 до 2999 включительно.

Ответ: 4

**Задача 5. (13 баллов)**

**Вариант 1.**

Рассмотрим упрощенный вариант игры “Маджонг”. Игра ведется комплектом из 16 пар костей (всего 32 кости). Каждую кость для удобства обозначим шестнадцатеричной цифрой (0-9, A-F). Кости выкладываются в два слоя размера 4x4, один строго над другим. За один ход можно убрать одну пару костей, обозначенных одной и той же цифрой, если каждую из костей можно убрать независимо от другой. Кость можно убрать, если на ней не лежит другая кость, и либо слева, либо справа, либо с обеих сторон соседнее место не занято другой костью (в частности, из-за края доски).

Последовательно удаляя пары костей в результате получится терминальная конфигурация, в которой нельзя удалить ни одну пару. Возможно, что в результате удаления пар костей все поле будет очищено.

Начальную конфигурацию будем записывать в виде 32 шестнадцатеричных цифр следующим образом:

Верхний слой

ABCDEF  
DE98  
1234  
5670

Нижний слой

0123  
4567

89AB  
CDEF

Запись конфигурации: ABCFDE98123456700123456789ABCDEF

Решите задачу:

Дана конфигурация: 760BF84BD325FAE7E16D9A243C95081C

Какова сумма значений всех оставшихся костей в терминальной конфигурации?

Ответ: 46

### Вариант 2.

Рассмотрим упрощенный вариант игры “Маджонг”. Игра ведется комплектом из 16 пар костей (всего 32 кости). Каждую кость для удобства обозначим шестнадцатеричной цифрой (0-9, A-F). Кости выкладываются в два слоя размера 4x4, один строго над другим. За один ход можно убрать одну пару костей, обозначенных одной и той же цифрой, если каждую из костей можно убрать независимо от другой. Кость можно убрать, если на ней не лежит другая кость, и либо слева, либо справа, либо с обеих сторон соседнее место не занято другой костью (в частности, из-за края доски).

Последовательно удаляя пары костей в результате получится терминальная конфигурация, в которой нельзя удалить ни одну пару. Возможно, что в результате удаления пар костей все поле будет очищено.

Начальную конфигурацию будем записывать в виде 32 шестнадцатеричных цифр следующим образом:

Верхний слой

ABCF  
DE98  
1234  
5670

Нижний слой

0123  
4567  
89AB  
CDEF

Запись конфигурации: ABCFDE98123456700123456789ABCDEF

Решите задачу:

Дана конфигурация: B41A8D389120EC6AE7FCD0F957B42356

Какова сумма значений всех оставшихся костей в терминальной конфигурации?

Ответ: 54

### Вариант 3.

Рассмотрим упрощенный вариант игры “Маджонг”. Игра ведется комплектом из 16 пар костей (всего 32 кости). Каждую кость для удобства обозначим шестнадцатеричной цифрой (0-9, A-F). Кости выкладываются в два слоя размера 4x4, один строго над другим. За один ход можно

убрать одну пару костей, обозначенных одной и той же цифрой, если каждую из костей можно убрать независимо от другой. Кость можно убрать, если на ней не лежит другая кость, и либо слева, либо справа, либо с обеих сторон соседнее место не занято другой костью (в частности, из-за края доски).

Последовательно удаляя пары костей в результате получится терминальная конфигурация, в которой нельзя удалить ни одну пару. Возможно, что в результате удаления пар костей все поле будет очищено.

Начальную конфигурацию будем записывать в виде 32 шестнадцатеричных цифр следующим образом:

**Верхний слой**

ABCF

DE98

1234

5670

**Нижний слой**

0123

4567

89AB

CDEF

Запись конфигурации: ABCFDE98123456700123456789ABCDEF

Решите задачу:

Дана конфигурация: 98AD6E300F17D5CB832C61E9BA27454F

Какова сумма значений всех оставшихся костей в терминальной конфигурации?

Ответ: 36

#### **Вариант 4.**

Рассмотрим упрощенный вариант игры “Маджонг”. Игра ведется комплектом из 16 пар костей (всего 32 кости). Каждую кость для удобства обозначим шестнадцатеричной цифрой (0-9, A-F). Кости выкладываются в два слоя размера 4x4, один строго над другим. За один ход можно убрать одну пару костей, обозначенных одной и той же цифрой, если каждую из костей можно убрать независимо от другой. Кость можно убрать, если на ней не лежит другая кость, и либо слева, либо справа, либо с обеих сторон соседнее место не занято другой костью (в частности, из-за края доски).

Последовательно удаляя пары костей в результате получится терминальная конфигурация, в которой нельзя удалить ни одну пару. Возможно, что в результате удаления пар костей все поле будет очищено.

Начальную конфигурацию будем записывать в виде 32 шестнадцатеричных цифр следующим образом:

**Верхний слой**

ABCF

DE98

1234

5670

Нижний слой

0123

4567

89AB

CDEF

Запись конфигурации: ABCFDE98123456700123456789ABCDEF

Решите задачу:

Дана конфигурация: 1A3414D7FBA6680982EC5DB97C350E2F

Какова сумма значений всех оставшихся костей в терминальной конфигурации?

Ответ: 62

### Задача 6. (10 баллов)

#### Вариант 1.

Предположим, что переменная типа CHAR занимает 1 байт, переменная типа INTEGER занимает два байта, а переменная типа REAL занимает 4 байта. Информация о школьнике описывается следующим образом:

```
name : ARRAY [1..16] OF CHAR;  
gender, grade : CHAR;  
year : INTEGER;  
avg : REAL;
```

Определите объем памяти в байтах, который необходим для хранения информации о 32 школьниках.

Ответ: 768

#### Вариант 2.

Предположим, что переменная типа CHAR занимает 1 байт, переменная типа INTEGER занимает два байта, а переменная типа REAL занимает 4 байта. Информация о книге описывается следующим образом:

```
weight : REAL;  
author : ARRAY [1..12] OF CHAR;  
name : ARRAY [1..12] OF CHAR;  
year : INTEGER;  
pages : INTEGER;
```

Определите объем памяти в байтах, который необходим для хранения информации о 64 книгах.

Ответ: 2048

### **Вариант 3.**

Предположим, что переменная типа CHAR занимает 1 байт, переменная типа INTEGER занимает два байта, а переменная типа REAL занимает 4 байта. Информация об аудиозаписи описывается следующим образом:

```
price : REAL;
name : ARRAY [1..12] OR CHAR;
artist : ARRAY [1..16] OF CHAR;
year : INTEGER;
number : INTEGER;
```

Определите объем памяти в байтах, который необходим для хранения информации о 50 записях.

Ответ: 1800

### **Вариант 4.**

Предположим, что переменная типа CHAR занимает 1 байт, переменная типа INTEGER занимает два байта, а переменная типа REAL занимает 4 байта. Информация о картине описывается следующим образом:

```
name : ARRAY [1..20] OR CHAR;
artist : ARRAY [1..12] OF CHAR;
price : REAL;
width : INTEGER;
height : INTEGER;
```

Определите объем памяти в байтах, который необходим для хранения информации о 100 картинах.

Ответ: 4000

### **Задача 7. (24 балла)**

#### **Вариант 1.**

Напишите программу, решающую следующую задачу.

Результат письменной работы оценивается исходя из 100 баллов (0 --- минимальная оценка, 100 --- максимальная). Сначала программе задается 101 целое число --- число человек, которые получили за письменную работу соответствующее число баллов. Затем программе задается число баллов за работу, которое набрал некоторый человек (этот человек уже учтен в статистике, заданной в начале).

Программа должна напечатать процент людей (от общего количества писавших работу), которые написали работу не хуже этого человека. Процент людей должен выводиться как целое число в диапазоне от 0 до 100. При необходимости дробная часть отбрасывается. Для вычислений используйте только целочисленные операции.

Программа может быть написана на языках C, C++, Java, C#, QuickBasic, VisualBasic .NET,

Pascal ABC, Delphi, FreePascal, Python2, Python3, PHP. Программа должна считывать входные данные со стандартного потока ввода (терминала, консоли) и выводить результат на стандартный поток вывода (терминал, консоль).

### **Вариант 2.**

Напишите программу, решающую следующую задачу.

Результат письменной работы оценивается исходя из 100 баллов (0 --- минимальная оценка, 100 --- максимальная). Сначала программе задается 101 целое число --- число человек, которые получили за письменную работу соответствующее число баллов. Затем программе задается число баллов за работу, которое набрал некоторый человек (этот человек уже учтен в статистике, заданной в начале).

Программа должна напечатать процент людей (от общего количества писавших работу), которые написали работу лучше этого человека. Процент людей должен выводиться как целое число в диапазоне от 0 до 100. При необходимости дробная часть отбрасывается. Для вычислений используйте только целочисленные операции.

Программа может быть написана на языках C, C++, Java, C#, QuickBasic, VisualBasic .NET, Pascal ABC, Delphi, FreePascal, Python2, Python3, PHP. Программа должна считывать входные данные со стандартного потока ввода (терминала, консоли) и выводить результат на стандартный поток вывода (терминал, консоль).

### **Вариант 3.**

Напишите программу, решающую следующую задачу. Результат письменной работы оценивается исходя из 100 баллов (0 --- минимальная оценка, 100 --- максимальная). Сначала программе задается 101 целое число --- число человек, которые получили за письменную работу соответствующее число баллов. Затем программе задается число баллов за работу, которое набрал некоторый человек (этот человек уже учтен в статистике, заданной в начале).

Программа должна напечатать процент людей (от общего количества писавших работу), которые написали работу не лучше этого человека. Процент людей должен выводиться как целое число в диапазоне от 0 до 100. При необходимости дробная часть отбрасывается. Для вычислений используйте только целочисленные операции.

Программа может быть написана на языках C, C++, Java, C#, QuickBasic, VisualBasic .NET, Pascal ABC, Delphi, FreePascal, Python2, Python3, PHP. Программа должна считывать входные данные со стандартного потока ввода (терминала, консоли) и выводить результат на стандартный поток вывода (терминал, консоль).

### **Вариант 4.**

Напишите программу, решающую следующую задачу. Результат письменной работы оценивается исходя из 100 баллов (0 --- минимальная оценка, 100 --- максимальная). Сначала программе задается 101 целое число --- число человек, которые получили за письменную работу соответствующее число баллов. Затем программе задается число баллов за работу, которое набрал некоторый человек (этот человек уже учтен в статистике, заданной в начале).

Программа должна напечатать процент людей (от общего количества писавших работу), которые написали работу хуже этого человека. Процент людей должен выводиться как целое число в диапазоне от 0 до 100. При необходимости дробная часть отбрасывается. Для вычислений используйте только целочисленные операции.

Программа может быть написана на языках C, C++, Java, C#, QuickBasic, VisualBasic .NET,

Pascal ABC, Delphi, FreePascal, Python2, Python3, PHP. Программа должна считывать входные данные со стандартного потока ввода (терминала, консоли) и выводить результат на стандартный поток вывода (терминал, консоль).

**Олимпиада Ломоносов**  
**3 тур, задача 7**

Максимальный балл: 24, выставляется за полное правильное решение, не использующее вещественной арифметики.

Вариант 1.

```
#include <iostream>

using namespace std;

int main(void)
{
    int people[101];
    int count = 0;
    int score;

    for (int i = 0; i <= 100; ++i) {
        cin >> people[i];
        count += people[i];
    }

    cin >> score;

    int count2 = -1;
    for (int i = score; i <= 100; ++i) {
        count2 += people[i];
    }

    cout << count2 * 100 / count << endl;

    return 0;
}
```

Вариант 2

```
#include <iostream>

using namespace std;

int main(void)
{
    int people[101];
    int count = 0;
    int score;

    for (int i = 0; i <= 100; ++i) {
        cin >> people[i];
        count += people[i];
    }

    cin >> score;

    int count2 = 0;
    for (int i = score + 1; i <= 100; ++i) {
        count2 += people[i];
    }

    cout << count2 * 100 / count << endl;

    return 0;
}
```

Вариант 3

```
#include <iostream>

using namespace std;

int main(void)
{
    int people[101];
    int count = 0;
    int score;

    for (int i = 0; i <= 100; ++i) {
        cin >> people[i];
        count += people[i];
    }

    cin >> score;

    int count2 = -1;
    for (int i = 0; i <= score; ++i) {
        count2 += people[i];
    }

    cout << count2 * 100 / count << endl;

    return 0;
}
```

Вариант 4

```
#include <iostream>

using namespace std;

int main(void)
{
    int people[101];
    int count = 0;
    int score;

    for (int i = 0; i <= 100; ++i) {
        cin >> people[i];
        count += people[i];
    }

    cin >> score;

    int count2 = 0;
    for (int i = 0; i < score; ++i) {
        count2 += people[i];
    }

    cout << count2 * 100 / count << endl;

    return 0;
}
```