

**Московский государственный университет имени  
М. В. Ломоносова**

Олимпиада «Ломоносов», информатика, 2012 год, вариант 1, ответы.

**Задача 1.**

463A24+B1615A=3A01B3

**Задача 2.**

21

DJ@TDRHN@

**Задача 3.**

Первое место — Бразилия, второе место — Италия.

**Задача 4.**

1023

**Задача 5.**

$n = 8$

**Задача 6.**

$2^{55} - 1$

**Задача 7.**

703109 23466114210671566104

394775 31542316614421142346

**Задача 8.**

```
#include <stdio.h>
#include <stdlib.h>
const int coins[256] =
{
    ['f'] = 1,           // Farthing
    ['h'] = 2,           // Halfpenny
    ['d'] = 4,           // Penny
    ['t'] = 4 * 3,       // Threepence
    ['s'] = 4 * 6,       // Sixpence
    ['S'] = 4 * 12,      // Shilling
    ['F'] = 4 * 12 * 2,  // Florin
    ['H'] = 4 * 12 * 2 + 4 * 6, // Half crown
    ['C'] = 4 * 12 * 5,  // Crown
    ['D'] = 4 * 12 * 10, // Half sovereign
    ['P'] = 4 * 12 * 20, // Sovereign
    ['G'] = 4 * 12 * 21, // Guinea
    [' ', ''] = -1,
};
struct money
{
    int index;
    int sum;
```

```

};
struct money money[1024];
int sort_func(const void *p1, const void *p2)
{
    const struct money *m1 = (const struct money *) p1;
    const struct money *m2 = (const struct money *) p2;
    if (m1->sum > m2->sum) return 1;
    if (m1->sum < m2->sum) return -1;
    return 0;
}
int main(void)
{
    int count = 0, c;
    money[count].index = count + 1;
    while ((c = getchar()) != EOF && coins[c]) {
        if (coins[c] < 0) {
            ++count;
            money[count].index = count + 1;
        } else {
            money[count].sum += coins[c];
        }
    }
    ++count;
    qsort(money, count, sizeof(money[0]), sort_func);
    for (int i = 0; i < count; ++i) {
        if (i > 0) printf(",");
        printf("%d", money[i].index);
    }
    printf("\n");
    return 0;
}

```

**Московский государственный университет имени  
М. В. Ломоносова**

Олимпиада «Ломоносов», информатика, 2012 год, вариант 2, ответы.

**Задача 1.**

Решений нет.

**Задача 2.**

25

ELIQATETA

**Задача 3.**

Первое место — Франция, второе место — Бразилия.

**Задача 4.**

960 или 1020.

**Задача 5.**

$n = 8$

**Задача 6.**

$2^{55} - 1$

**Задача 7.**

294653 21066346235421542344

563961 61063304611631047306

**Задача 8.**

```
#include <stdio.h>
#include <stdlib.h>
const int coins[256] =
{
    ['f'] = 1,           // Farthing
    ['h'] = 2,           // Halfpenny
    ['d'] = 4,           // Penny
    ['t'] = 4 * 3,       // Threepence
    ['s'] = 4 * 6,       // Sixpence
    ['S'] = 4 * 12,      // Shilling
    ['F'] = 4 * 12 * 2,  // Florin
    ['H'] = 4 * 12 * 2 + 4 * 6, // Half crown
    ['C'] = 4 * 12 * 5,  // Crown
    ['D'] = 4 * 12 * 10, // Half sovereign
    ['P'] = 4 * 12 * 20, // Sovereign
    ['G'] = 4 * 12 * 21, // Guinea
    [' ', ''] = -1,
};
struct money
{
    int index;
    int sum;
};
struct money money[1024];
```

```

int sort_func(const void *p1, const void *p2)
{
    const struct money *m1 = (const struct money *) p1;
    const struct money *m2 = (const struct money *) p2;
    if (m1->sum > m2->sum) return 1;
    if (m1->sum < m2->sum) return -1;
    return 0;
}

int main(void)
{
    int count = 0, c;
    money[count].index = count + 1;
    while ((c = getchar()) != EOF && coins[c]) {
        if (coins[c] < 0) {
            ++count;
            money[count].index = count + 1;
        } else {
            money[count].sum += coins[c];
        }
    }
    ++count;
    qsort(money, count, sizeof(money[0]), sort_func);
    for (int i = 0; i < count; ++i) {
        if (i > 0) printf(",");
        printf("%d", money[i].index);
    }
    printf("\n");
    return 0;
}

```

Московский государственный университет имени  
М. В. Ломоносова

Олимпиада «Ломоносов», информатика, 2012 год, вариант 2.

Таблица сложения.

	B	A	0	1	2	3	4	5	6
B	A5	A6	B	A	0	1	2	3	4
A	A6	B	A	0	1	2	3	4	5
0	B	A	0	1	2	3	4	5	6
1	A	0	1	2	3	4	5	6	1B
2	0	1	2	3	4	5	6	1B	1A
3	1	2	3	4	5	6	1B	1A	10
4	2	3	4	5	6	1B	1A	10	11
5	3	4	5	6	1B	1A	10	11	12
6	4	5	6	1B	1A	10	11	12	13

Задача 1. Решение.

B46132+A12155=A66A3B

Задача 2. Решение.

Char	Dec	Bin	Mod. bin	Mod. char
E	69	01000101	01000101	E
L	76	01001100	01001100	L
I	73	01001001	01001001	I
S	83	01010011	01010001	Q
A	65	01000001	01000001	A
V	86	01010110	01010100	T
E	69	01000101	01000101	E
T	84	01010100	01010100	T
A	65	01000001	01000001	A

Answer: 25

Задача 3. Решение. Обозначим высказывания:

A = «в 1998 году чемпионом стала Хорватия»,

B = «в 1998 второе место заняла сборная Бразилии»,

C = «в 1998 первое место заняла сборная Франции»,

D = «в 1998 второе место заняла сборная Хорватии».

Так как брат был наполовину прав, то  $A \& \text{not } B = 1$  или  $\text{not } A \& B = 1$ ; сестра была наполовину неправа, то  $C \& \text{not } D = 1$  или  $\text{not } C \& D = 1$ .

Пусть  $A=1$ , тогда  $B=0$ . Так как  $A=1$ , то первое место заняла Хорватия, значит  $C=1$  и  $D=0$ , но тогда получаем противоречие: две команды заняли первое место.

Пусть  $A=0$ , тогда  $B=1$ . Так как второе место заняла сборная Бразилии, то  $D=0$ , значит,  $C=1$ .

Ответ. Первое место Франция, второе место Бразилия.

Задача 4. Решение. Пусть в четвертый час Дед поймал  $x$  рыбок, тогда во второй час он поймал 3 рыбок, а в третий час поймал  $2 + 4$  рыбки. Всего Дед поймал  $6 + 4$  рыбки, при этом — целое число. Пусть Дед поймал  $y$  рыбок, тогда — четное, в двоичной записи числа  $y$  нет незначащих нулей, значит, запись начинается с 1. Пусть 10 одинаковых цифр в записи — это 1, тогда запись заканчивается 1 не может, так как это будет нечетное число, а  $y$  — четное, значит, запись заканчивается 0 и это  $1111111110_2 = 2046$ .  $6 + 4 = 2046$ ,  $6x = 2042$ ,  $x$  — нецелое. Тогда 10 одинаковых цифр в записи числа — это 0 и это  $10000000000_2 = 1024$ .  $6 + 4 = 1024$ ,  $6x = 1020$ ,  $x$  — целое. Таким образом, Дед поймал 1024 рыбки.

Лиса выкинула сумму первых 10 членов геометрической последовательности с частным 2 и первым членом 1.  $S_{10} = 2^{10} - 1 = 1023$ . Всего Дед привез домой  $1024 - 1023 = 1$  рыбку.

Ответ. 1

Задача 5. Решение. Число чисел  $K_n = 3 \cdot (4^{n-1} - 3^{n-1})$ .

$n$	$4^{n-1}$	$3^{n-1}$	$4^{n-1} - 3^{n-1}$	$3(4^{n-1} - 3^{n-1})$
2	4	3	1	3
3	16	9	7	21
4	64	27	37	111
5	256	81	175	525
6	1024	243	781	2343
7	4096	729	3367	10101
8	16384	2187	14197	42591

Таким образом,  $n = 8$ .

Задача 6. Решение. Необходимо посчитать, сколько можно написать различных программ для вычислителя, который умеет прибавлять 1 или умножать на 5 число на экране, которые из 1 получают 101. Рассмотрим количество программ для получения каждого числа от 1 до 101. Пусть  $n$  — это число,  $P(n)$  — это количество различных программ.  $P(1) = 1$ , так как из 1 можно получить 1 пустой программой. Если  $n$  не делится на 5, то  $P(n) = P(n-1)$ , если  $n$  делится на 5, то  $P(n) = P(n-1) + P(n/5)$ .

001 001  
006 002  
011 003  
016 004  
021 005  
026 007  
031 009

036 011  
041 013  
046 015  
051 018  
056 021  
061 024  
066 027  
071 030  
076 034  
081 038  
086 042  
091 046  
096 050  
101 055  
101 055

Таким образом, число программ равно 55. Выплачиваемая сумма равна  $2^{55} - 1$ .

**Ответ.**  $2^{55} - 1$

**Задача 7. Решение.**

Был использован следующий алгоритм шифрования:

1. Входные цифры разбиваются на пары. Например,

931001 -> 93 10 01

2. В каждой паре цифре { 0 ... 9 } и позиции в паре { 1, 2 } ставится в соответствие последовательность из 5 цифр согласно таблице.

	1	2
0	00220	11331
1	20002	31113
2	02002	13113
3	22000	33111
4	00202	11313
5	20200	31311
6	02200	13311
7	00022	11133
8	20020	31131
9	02020	13131

9 -> 02020

3 -> 33111

3. Для каждой пары строится последовательность из 10 цифр где на нечетных местах находятся цифры, соответствующие первой исходной цифре, а на четных местах — цифры, соответствующие второй исходной цифре.

93 -> 0323012101

4. Получившиеся последовательности из 10 цифр конкатенируются.

931001 -> 032301210121010303210301212103

5. Полученная последовательность переводится из 4-ричной системы счисления в 8-ричную и записывается в обратном порядке.

931001 61462306630431062354

Таким образом,

294653 21066346235421542344

563961 61063304611631047306

Ключевые моменты при расшифровке:

1. Перевод из 8-ричной системы счисления в 4-ричную и изменения порядка записи.

2. Разбиение на группы по 10 цифр и сопоставление с группами по две цифры в незашифрованном сообщении.

3. Примеры подобраны так, что в незашифрованном сообщении каждая цифра присутствует и на нечетной, и на четной позиции.

**Ответ:**

294653 21066346235421542344

563961 61063304611631047306

**Задача 8. Этапы решения.**

1. Анализ кодов монет и суммирование значений в описи.

2. Сортировка описей вместе с индексами.

3. Вывод результата.

```
#include <stdio.h>
#include <stdlib.h>
const int coins[256] =
{
    ['f'] = 1,           // Farthing
    ['h'] = 2,           // Halfpenny
    ['d'] = 4,           // Penny
    ['t'] = 4 * 3,       // Threepence
    ['s'] = 4 * 6,       // Sixpence
    ['s'] = 4 * 12,      // Shilling
    ['f'] = 4 * 12 * 2,  // Florin
    ['h'] = 4 * 12 * 2 + 4 * 6, // Half crown
    ['c'] = 4 * 12 * 5,  // Crown
    ['d'] = 4 * 12 * 10, // Half sovereign
    ['p'] = 4 * 12 * 20, // Sovereign
    ['g'] = 4 * 12 * 21, // Guinea
    [''] = -1,
};
struct money
```

```

{
    int index;
    int sum;
};
struct money money[1024];
int sort_func(const void *p1, const void *p2)
{
    const struct money *m1 = (const struct money *) p1;
    const struct money *m2 = (const struct money *) p2;
    if (m1->sum > m2->sum) return 1;
    if (m1->sum < m2->sum) return -1;
    return 0;
}
int main(void)
{
    int count = 0, c;
    money[count].index = count + 1;
    while ((c = getchar()) != EOF && coins[c]) {
        if (coins[c] < 0) {
            ++count;
            money[count].index = count + 1;
        } else {
            money[count].sum += coins[c];
        }
    }
    ++count;
    qsort(money, count, sizeof(money[0]), sort_func);
    for (int i = 0; i < count; ++i) {
        if (i > 0) printf(",");
        printf("%d", money[i].index);
    }
    printf("\n");
    return 0;
}

```