

Ответы на задания заочного этапа олимпиады школьников "Ломоносов" по информатике 2015/16 учебный год

*Решения отборочного тура 5-9 классов не отличаются от решений
первого отборочного тура 10-11 классов.*

Первый тур 10-11 классы

Задание 1

вариант 1.1) 273:

bfl
blf
fbl
flb
lbf
lfb
lt
tl

вариант 1.2) 165:

bdj
bjd
dbj
djb
jbd
jdb
jn
nj

вариант 1.3) 230:

adv
avd
dav
dva
iv
vad
vda

vi

вариант 1.4) 238:

afp
apf
far
fra
mp
raf
pfa
pm

Схема решения:

- 1) Следует найти все различные разложения заданного числа на множители, величина которых от 2 до 27. Каждое найденное разложение даёт набор букв, из которых можно составить пароль.
- 2) Для каждого найденного набора букв следует построить все перестановки. Каждая перестановка определяет порядок букв в пароле, т. е. одну из строк ответа. Если множитель дважды участвует в разложении, будут порождаться дублирующие строки в ответе.
- 3) Все строки ответа следует упорядочить по алфавиту. Дублирующие строки ответа, если таковые будут, можно отсеять на этом этапе.

Задание 2

Задача аналогично задаче 1, но требовалось отправить все варианты пароля и привести обоснование, например, программу генерации всех паролей.

2.1) 2348346:

$$2*3*7*11*13*17*23 = 2348346 \quad (7! + 5*6! + 2*5! = 8880 \text{ вариантов})$$

2.2) 1677390:

$$2*3*5*11*13*17*23 = 1677390 \quad (7! + 5*6! + 2*5! = 8880 \text{ вариантов})$$

2.3) 3913910:

$$2*5*7*11*13*17*23 = 3913910 \quad (7! + 4*6! = 7920 \text{ вариантов})$$

2.4) 1118260:

$$2*2*5*11*13*17*23 = 1118260 \quad (7!/2 + 4*6! + 4*5! = 5880 \text{ вариантов})$$

Решение:

Программу, обосновывающую решение, можно составить, опираясь на схему из решения задачи 1.

Задание 3

Пример решения на языке C++, в задаче нельзя было хранить весь куб в памяти:

```
#include <cstdio>
using namespace std;

int a[250][250];
int main(int argc, char *argv[])
{
    int n;
    scanf("%d\n", &n);

    for (int t = 0; t < n; ++t) {
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < n; ++j) {
                a[i][j] |= (getc(stdin) - '0') ^ 1;
            }
        }
    }

    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            printf("%d", (a[i][j] ^ 1));
        }
    }
    return 0;
}
```

Дополнение к решению: если вести подсчёт непрозрачных “пикселей” в проекции, то это позволит останавливаться досрочно, если по текущим вычислениям станет ясно, что вся проекция непрозрачна.

Задание 4

Во всех вариантах используется шифр замены
вариант 4.1:

Far out in the uncharted backwaters of the unfashionable end of the western spiral arm of the Galaxy lies a small unregarded yellow sun. Orbiting this at a distance of roughly ninety-two million miles is an utterly insignificant little blue green planet whose ape-descended life forms are so amazingly primitive that they still think digital watches are a pretty neat idea. This planet has - or rather had - a problem, which was this: most of the people on it were unhappy for pretty much of the time. Many solutions were

suggested for this problem, but most of these were largely concerned with the movements of small green pieces of paper, which is odd because on the whole it wasn't the small green pieces of paper that were unhappy.

вариант 4.2:

Mr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense. Mr. Dursley was the director of a firm called Grunnings, which made drills. He was a big, beefy man with hardly any neck, although he did have a very large mustache. Mrs. Dursley was thin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent so much of her time craning over garden fences, spying on the neighbors. The Dursleys had a small son called Dudley and in their opinion there was no finer boy anywhere.

вариант 4.3:

Every inch of wall space is covered by a bookcase. Each bookcase has six shelves, going almost to the ceiling. Some bookshelves are stacked to the brim with hardback books: science, maths, history, and everything else. Other shelves have two layers of paperback science fiction, with the back layer of books propped up on old tissue boxes or lengths of wood, so that you can see the back layer of books above the books in front. And it still isn't enough. Books are overflowing onto the tables and the sofas and making little heaps under the windows. This is the living-room of the house occupied by the eminent Professor Michael Verres-Evans, and his wife, Mrs. Petunia Evans-Verres, and their adopted son, Harry James Potter-Evans-Verres.

вариант 4.4:

Let's set the existence-of-God issue aside for a later volume, and just stipulate that in some way, self-replicating organisms came into existence on this planet and immediately began trying to get rid of each other, either by spamming their environments with rough copies of themselves, or by more direct means which hardly need to be belabored. Most of them failed, and their genetic legacy was erased from the universe forever, but a few found some way to survive and to propagate. After about three billion years of this sometimes zany, frequently tedious fugue of carnality and carnage, Godfrey Waterhouse IV was born, in Murdo, South Dakota, to Blanche, the wife of a Congregational preacher named Bunyan Waterhouse. Like every other creature on the face of the earth, Godfrey was, by birthright, a stupendous badass, albeit in the somewhat narrow technical sense that he could trace his ancestry back up a long line of slightly less highly evolved stupendous badasses to that first

self-replicating gizmo-which, given the number and variety of its descendants, might justifiably be described as the most stupendous badass of all time. Everyone and everything that wasn't a stupendous badass was dead.

Задание 5

Пример решения на C++:

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

const int maxN = 1e6 + 1000;

typedef long long ll;

ll a[maxN];

int main(void) {
    ios_base::sync_with_stdio(0);
    ll n;
    int m;
    cin >> n >> m;
    ll sum = 0;
    for (int i = 1; i <= m; i++) {
        cin >> a[i];
        sum += a[i];
    }
    if (sum <= n) {
        cout << m << endl;
        return 0;
    }
    ll p = sum - n;
    int ans = 1;
    int r = 1;
    ll cur = 0;
    for (int i = 2; i <= m; i++) {
        if (r < i) {
            cur = 0;
            r = i;
        } else {
            cur -= a[r];
        }
    }
```

```

        while (cur < p && r <= m) {
            cur += a[r];
            r++;
        }
        if (cur < p) {
            break;
        }
        ans = max(ans, m - (r - i));
        r--;
        cur -= a[i];
    }
    cout << ans << endl;
    return 0;
}

```

Задание 6

Пример решения:

```

]?( {} )? ) } ! { { ( [ ] ] { ( ~ ] { > ( [ ] } ! ) ) [ ( ~ } } ! ) [ ] ] { } ) } [ . @ . { { } "

```

Задание 7

Для анализа архива с текстами программ требовалось написать программу, которая являлась обоснованием получения результата.

Задание 8

При чтении последовательности запоминаем индексы элементов, где нарушается порядок. Так получаем те элементы, которые требуется изменить.

Входная последовательность считывается поэлементно. Текущее считанное число следует сравнивать с предыдущим и запоминать номера пар, в которых числа не по порядку. Таких пар может быть одна или две. Если найдена лишь одна пара, и последовательность дочитана до конца, то ответ состоит из номера пары и номера следующего за ним числа. Если пар две, то ответ состоит из номера первой пары и увеличенного на 1 номера второй пары. Если встречено две пары, то ввод можно не дочитывать.

```

program task8(input, output);
var    endFlag: boolean;
        aPrev, aCurr, i, n, num1, num2, a1st, a2nd: integer;
begin
    num1:= 0;
    num2:= 0;

```

```

endFlag:= false;
read(n, aCurr);
i:= 1;
repeat
    aPrev:= aCurr;
    read(aCurr);
    if (aPrev > aCurr) then begin {var 2: aPrev < aCurr}
        if (num1 = 0) then begin
            num1 := i;
            a1st:= aCurr;
            a2nd:= aPrev
        end
        else begin
            num2:= i+1;
            a1st:= aCurr;
            endFlag:= true;
        end;
    end;
    i:= i + 1;
until (i = n) or endFlag;
if endFlag then begin
    { writeln(num2, ' ', num1); var 2}
    writeln(a1st, ' ', a2nd) {var 1}
end
else begin
    { writeln(num1+1, ' ', num1); var 2}
    writeln(a1st, ' ', a2nd) {var 1}
end;
end.

```

Второй тур 10-11 классы

Задание 1

вариант 1.1

ayyz
ayzb
azaz
azbb
bayz
bazb
bbaz
bbbb

вариант 1.2

cyyz
cyzb
czaz
czbb
dayz
dazb
dbaz
dbbb

вариант 1.3

aaaaz
aaabb
yaaz
yabb
xyaz
xybb
xxyz
xxzb

вариант 1.4

gyyz
gyzb
gzaz
gzbb
hayz
hazb
hbaz
hbbb

Решение: Можно заметить, что некоторые числа могут быть представлены двумя или более разными полиномами. Так происходит потому, что $s[i] \cdot 24^{n-i} + s[i+1] \cdot 24^{n-i-1} = (s[i]-1) \cdot 24^{n-i} + (24+s[i+1]) \cdot 24^{n-i-1}$ а $s[i]$, $(s[i]-1)$, $s[i+1]$ и $(24+s[i+1])$ могут быть кодами букв при $1 \leq s[i] < 27$, $0 \leq s[i+1] < 3$. Если $i=1$, то следует рассматривать варианты из обеих частей тождества при $1 \leq s[1] < 27$, $0 \leq s[2] < 3$, а букву с 0-м номером считать "пустым местом" (см. ответы длиной 5 и длиной 4 в варианте 3).

Схема решения:

- 1) Если $X > 0$, то goto 2, иначе ответа нет.
- 2) Находим p – наибольшую степень 24, не превосходящую X .
- 3) Если $p > 1$, то goto 4, иначе ответ – строка из одной X -ой буквы.
- 4) Находим s – частное и r – остаток при делении нацело X на p .
- 5) Находим s_{next} – частное и r_{next} – остаток при делении r на $(p/24)$.
- 6) Если $1 \leq s < 27$ и $0 \leq s_{next} < 3$, то в ответ надо включить строки двух видов:

а) начинающиеся с s -ой буквы и продолжающиеся строками, полученными при расшифровке остатка r ;

б) начинающиеся с $(s-1)$ -ой буквы и $(s_{next}+24)$ -ой буквы за ней, и при $r_{next}>0$ продолжающиеся строками, полученными при расшифровке r_{next} , при $r_{next}=0$ и $p=24$ расшифровка состоит лишь из двух указанных букв.

7) Если $s=1$ и $0 \leq s_{next} < 3$ и мы определяем первую букву ответа для заданного числа (а не для его остатков), то также есть два вида строк: а) аналогичные указанному в б)а); и б) начинающиеся с $(s_{next}+24)$ -ой буквы, и продолжающиеся строками, полученными при расшифровке остатка r_{next} .

8) Во всех остальных случаях буква ответа определяется однозначно, это s -ая буква, и расшифровка продолжается для остатка r . По указанной схеме ищется множество слов, затем упорядочивается по алфавиту.

Задание 2

Кроме ответа требовалось привести то, как он был получен, например, программу.

вариант 2.1. $X = 9571766450$

Вывод (128 строк, для экономии места слова выписаны подряд через пробелы):

```
ayyyyyyz ayyyyyzb ayyyzaz ayyyzabb ayyyzayz ayyyzazb
ayyyzbaz ayyyzbbb ayyzayyz ayyzayzb ayyzazaz ayyzazbb ayyzbayz
ayyzbazb ayyzbbaz ayyzbbbb ayzayyyz ayzayyzb ayzayzaz ayzayzbb
ayzazayz ayzazazb ayzazbaz ayzazbbb ayzbayyz ayzbayzb ayzbazaz
ayzbazbb ayzbbayz ayzbbazb ayzbbbbaz ayzbbbbbb azayyyyz azayyyzb
azayyzaz azayyzbb azayzayz azayzazb azayzbaz azayzbbb azazayyz
azazayzb azazazaz azazazbb azazbayz azazbazb azazbbaz azazbbbb
azbayyyz azbayyzb azbayzaz azbayzbb azbazayz azbazazb azbazbaz
azbazbbb azbbayyz azbbayzb azbbazaz azbbazbb azbbbayz azbbbazb
azbbbbaz azbbbbbb bayyyyyz bayyyyzb bayyyzaz bayyyzbb bayyzayz
bayyzazb bayyzbaz bayyzbbb bayzayyz bayzayzb bayzazaz bayzazbb
bayzbayz bayzbazb bayzbbaz bayzbbbb bazayyyz bazayyzb bazayzaz
bazayzbb bazazayz bazazazb bazazbaz bazazbbb bazbayyz bazbayzb
bazbazaz bazbazbb bazbbayz bazbbazb bazbbbaz bazbbbbbb bbayyyyz
bbayyyzb bbayyzaz bbayyzbb bbayzayz bbayzazb bbayzbaz bbayzbbb
bbazayyz bbazayzb bbazazaz bbazazbb bbazbayz bbazbazb bbazbbaz
bbazbbbb bbbayyyz bbbayyzb bbbayzaz bbbayzbb bbbazayz bbbazazb
bbbazbaz bbbazbbb bbbbayyz bbbbayzb bbbbazaz bbbbazbb bbbbbayz
bbbbbazb bbbbbbaz bbbbbbbb
```

вариант 2.2 $X = 18744709298$

Вывод (128 строк, для экономии места слова выписаны подряд через пробелы):

```
сyyyyyyz сyyyyyzb сyyyzaz сyyyzabb сyyyzayz сyyyzazb
сyyyzbaz сyyyzbbb сyyzayyz сyyzayzb сyyzazaz сyyzazbb сyyzbayz
сyyzbazb сyyzbbaz сyyzbbbb сызayyyz сызayyzb сызayzaz сызayzbb
```

cyzazayz cyzazazb cyzazbaz cyzazbbb cyzbayyz cyzbayzb cyzbazaz
 cyzbazbb cyzbbayz cyzbbazb cyzbbbaz cyzbbbbb czayyyyz czayyyzb
 czayyzaz czayyzbb czayzayz czayzazb czayzbaz czayzbbb czazayyz
 czazayzb czazazaz czazazbb czazbayz czazbazb czazbbaz czazbbbb
 czbayyyy czbayyzb czbayzaz czbayzbb czbazayz czbazazb czbazbaz
 czbazbbb czbbayyz czbbayzb czbbazaz czbbazbb czbbbayz czbbbazb
 czbbbbbaz czbbbbbbb dayyyyyz dayyyyzb dayyyzaz dayyyzbb dayyzayz
 dayyzazb dayyzbaz dayyzbbb dayzayyz dayzayzb dayzazaz dayzazbb
 dayzbayz dayzbazb dayzbbaz dayzbbbb dazayyyz dazayyzb dazayzaz
 dazayzbb dazazayz dazazazb dazazbaz dazazbbb dazbayyz dazbayzb
 dazbazaz dazbazbb dazbbayz dazbbazb dazbbbaz dazbbbbbb dbayyyyz
 dbayyyzb dbayyzaz dbayyzbb dbayzayz dbayzazb dbayzbaz dbayzbbb
 dbazayyz dbazayzb dbazazaz dbazazbb dbazbayz dbazbazb dbazbbaz
 dbazbbbb dbbayyyz dbbayyzb dbbayzaz dbbayzbb dbbazayz dbbazazb
 dbbazbaz dbbazbbb dbbbayyz dbbbayzb dbbbazaz dbbbazbb dbbbbayz
 dbbbbazb dbbbbbaz dbbbbbbbb

вариант 2.3 X = 27917652146

Вывод (128 строк, для экономии места слова выписаны подряд через пробелы):

euyuyuyz euyuyyzb euyuyzaz euyuyzbb euyyzayz euyyzazb
 euyyzbaz euyyzbbb euyzayyz euyzayzb euyzazaz euyzazbb euyzbayz
 euyzbazb euyzbbaz euyzbbbb eyzayyyz eyzayyzb eyzayzaz eyzayzbb
 eyzazayz eyzazazb eyzazbaz eyzazbbb eyzbayyz eyzbayzb eyzbazaz
 eyzbazbb eyzbbayz eyzbbazb eyzbbbaz eyzbbbbb ezayyyyz ezayyyzb
 ezayyzaz ezayyzbb ezayzayz ezayzazb ezayzbaz ezayzbbb ezazayyz
 ezazayzb ezazazaz ezazazbb ezazbayz ezazbazb ezazbbaz ezazbbbb
 ezbayyyyz ezbayyzb ezbayzaz ezbayzbb ezbazayz ezbazazb ezbazbaz
 ezbazbbb ezbbayyz ezbbayzb ezbbazaz ezbbazbb ezbbbayz ezbbbazb
 ezbbbbbaz ezbbbbbbb fayyyyyz fayyyyzb fayyyzaz fayyyzbb fayyzayz
 fayyzazb fayyzbaz fayyzbbb fayzayyz fayzayzb fayzazaz fayzazbb
 fayzbayz fayzbazb fayzbbaz fayzbbbb fazayyyz fazayyzb fazayzaz
 fazayzbb fazazayz fazazazb fazazbaz fazazbbb fazbayyz fazbayzb
 fazbazaz fazbazbb fazbbayz fazbbazb fazbbbaz fazbbbbbb fbayyyyz
 fbayyyzb fbayyzaz fbayyzbb fbayzayz fbayzazb fbayzbaz fbayzbbb
 fbazayyz fbazayzb fbazazaz fbazazbb fbazbayz fbazbazb fbazbbaz
 fbazbbbb fbbayyyz fbbayyzb fbbayzaz fbbayzbb fbbazayz fbbazazb
 fbbazbaz fbbazbbb fbbbayyz fbbbayzb fbbbazaz fbbbazbb fbbbbayz
 fbbbbazb fbbbbbaz fbbbbbbb

вариант 2.4 X = 37090594994

Вывод (128 строк, для экономии места слова выписаны подряд через пробелы):

guyuyuyz guyuyyzb guyuyzaz guyuyzbb guyyzayz guyyzazb
 guyyzbaz guyyzbbb guyzayyz guyzayzb guyzazaz guyzazbb guyzbayz
 guyzbazb guyzbbaz guyzbbbb gyzayyyz gyzayyzb gyzayzaz gyzayzbb
 gyzazayz gyzazazb gyzazbaz gyzazbbb gyzbayyz gyzbayzb gyzbazaz
 gyzbazbb gyzbbayz gyzbbazb gyzbbbaz gyzbbbbbb gzayyyyz gzayyyzb
 gzayyzaz gzayyzbb gzayzayz gzayzazb gzayzbaz gzayzbbb gzazayyz
 gzazayzb gzazazaz gzazazbb gzazbayz gzazbazb gzazbbaz gzazbbbb

gzbayyyz gzbayyzb gzbayzaz gzbayzbb gzbazayz gzbazazb gzbazbaz
gzbazbbb gzbabayz gzbabayzb gzbazazb gzbazabb gzbabbayz gzbabbazb
gzbabbaz gzbabbbb hayyyyyz hayyyyzb hayyyzaz hayyyzbb hayyzayz
hayyzazb hayyzbaz hayyzbbb hayzayyz hayzayzb hayzazaz hayzazbb
hayzbayz hayzbazb hayzbbaz hayzbbbb hazayyyz hazayyzb hazayzaz
hazayzbb hazazayz hazazazb hazazbaz hazazbbb hazabayz hazbayzb
hazbazaz hazbazbb hazbbayz hazbbazb hazbbbaz hazbbbbbb hbayyyyyz
hbayyyzb hbayyzaz hbayyzbb hbayzayz hbayzazb hbayzbaz hbayzbbb
hbazayyz hbazayzb hbazazaz hbazazbb hbazbayz hbazbazb hbazbbaz
hbazbbbb hbbayyyz hbbayyzb hbbayzaz hbbayzbb hbbazayz hbbazazb
hbbazbaz hbbazbbb hbbbayyz hbbbayzb hbbbazaz hbbbazbb hbbbbayz
hbbbbazb hbbbbbaz hbbbbbbb

Программа, решающая задачу, может быть построена на основе схемы из решения задачи 1.

Задание 3

Решения проверялись на следующих тестах:

1 1
2 2
3 1+2
4 2*2
5 1+2*2
6 2+2*2
7 1+2+2*2
8 2*2*2
9 1+2*2*2
10 2+2*2*2
11 11
12 12
13 1+12
14 2+12
15 1+2+12
16 2+2+12
17 1+2+2+12
18 2+2+2+12
19 2*2*2+11
20 2*2*2+12
21 21
22 22
23 1+22
24 2*12
1000 112+2*2*222
1001 1+112+2*2*222
1002 2+112+2*2*222
1003 121+2*21*21

```

1004 122+2*21*21
1005 121+2*2*221
1006 122+2*2*221
1007 1+122+2*2*221
1008 2*2*12*21
1009 1+2*2*12*21
1010 2+2*2*12*21
1011 1+2+2*2*12*21
1012 2*22+2*22*22
1013 1+2*22+2*22*22
1014 11*12+2*21*21
100000 11112+2*2*22222
100001 1+11112+2*2*22222
100002 21*121+21*21*221
100003 11*211+2*221*221
100004 1+11*211+2*221*221
239 2*2*2+11*21
179 11+2*2*2*21
12345 1+2+11*1122
222222 222222

```

Для правильного решения требовалось найти оптимальное разложение числа на множители и числа на слагаемые.

```

1) строим множество с возможных чисел:
def fill_c(s):
    if (len(s) == 0 or int(s) < 1000000):
        if len(s) > 0:
            c.append(int(s))
            fill_c("1"+s)
            fill_c("2"+s)

2) строим ответ для каждого числа i в списке b[i]
def fill_b(n):
    if b[n] != "":
        return b[n]
    b[n] = "-1"
    if n in c:
        b[n] = str(n)
    for k in s_c:
        if n % k == 0 and k > 1 and n > k:
            b[n] = best_mul(b[n], fill_b(n // k), k)
    return b[n]

```

3) строим общий ответ для каждого числа, используя очередное слагаемое из b

```

def fill_a(n):

```

```

if a_f[n]:
    return a[n]
a[n] = b[n]
a_f[n] = True
if a[n] == "-1":
    a[n] = b_l[0][1] + "+" + fill_a(n - b_l[0][0])
for k in b_l:
    if k[0] >= n:
        break
    a[n] = best_sum(a[n], k[1], fill_a(n - k[0]))
return a[n]

```

Далее можно предподсчитать часть ответов и внести их в массив.

Задание 4

Задача "Лабиринт"

Во всех 4-х вариантах правильный ответ -1 (невозможно получить последовательность действий). Обоснованием может служить выделенная часть поля, где показывается что включить все лампочки невозможно. Общее решение следующее - для каждой лампочки вводим переменную включалась она или нет. Затем составляем систему линейных уравнений и получаем, что она не имеет решения.

Задание 5

Пример решения:

```

#include <cstdio>
#include <vector>
#include <cstdlib>
#include <cctype>

using namespace std;

const int frame_size = 8;

void encode(const vector<char> &in, vector<char> &out)
{
    for (int i = 0; i < int(in.size()); ++i) {
        out.push_back(in[i]);
        out.push_back(in[i]);
        out.push_back(in[i]);
        out.push_back(in[i]);
    }
    if (int(out.size()) % frame_size != 0) {

```

```

        out.push_back('0');
        out.push_back('0');
        out.push_back('0');
        out.push_back('0');
    }
}

void decode(const vector<char> &in, vector<char> &out)
{
    for (int pos = 0; pos < int(in.size()); pos += 4) {
        int sum = in[pos] + in[pos + 1] + in[pos + 2] + in[pos
+ 3] - 4 * '0';
        if (sum <= 1) sum = 0;
        else if (sum >= 3) sum = 1;
        out.push_back(char('0' + sum));
    }
}

int main()
{
    int cmd, size;
    vector<char> data;

    scanf("%d", &cmd);
    scanf("%d", &size);

    int c;
    while ((c = getchar()) != EOF) {
        if (isspace(c)) {
        } else if (c == '0' || c == '1') {
            data.push_back(c);
        }
    }

    vector<char> result;
    if (cmd == 0) {
        encode(data, result);
    } else if (cmd == 1) {
        decode(data, result);
    }

    for (int i = 0; i < int(result.size()); ++i) {
        if (i > 0 && i % 64 == 0) {
            putchar('\n');
        }
    }
}

```

```
        putchar(result[i]);
    }
    putchar('\n');
    return 0;
}
```

Задание 6

Пример решения

```
?}}?[]{(~[]){<([~]!){}[]{}[]{}[]{}[]?){(.@.{})"
```

Задание 7

Один из вариантов решения – найти число одинаковых символов между каждым эталоном и запрашиваемой картинкой. Далее выбрать эталон с максимальным числом совпадающих символов.

Задание 8

Задача аналогична задаче 8 из первого тура – находим тройки, которые не являются возрастающими или убывающими. Требуется поменять местами элементы в этих тройках, если одна из них не содержит минимальный элемент.