

Работа победителя заключительного этапа
командной инженерной олимпиады школьников
Олимпиада Национальной технологической инициативы

Профиль «ТЕХНОЛОГИИ БЕСПРОВОДНОЙ СВЯЗИ»

Гусева Полина Александровна

Класс: 10

Город: Москва

Школа: ГБОУ Лицей №1580

Регион: Москва

Уникальный номер участника:
16570138

Команда на заключительном этапе: Огонь

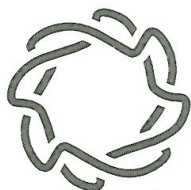
Параллель: 9-11 класс

Результаты заключительного этапа:

№	Индивидуальная часть										Командная часть		Результат
	Математика				Информатика					Итого	Макс. балл	Инж. задача	
	1	2	3	4	1. 1	1. 2	2. 1	2. 2	3. 1				3.2
16570138	15	0	0	0	7	13	7	13	7	0	200	3,613	74,902

Индивидуальная часть

Персональный лист участника с номером 16570138:



Олимпиада НТИ

ФИО Русева Полина Александровна

Город Москва

Школа № 1580

Математика

Лист 1:

Командная инженерная олимпиада «Олимпиада НТИ»

Направление Технологии Беспробудной Связи

Предмет Математика

Номер участника 165 70138

1 2 3 4 5
15 0 0 0 15

N 1.

$$\begin{cases} x+y+z=-1 & (1) \\ x^2+y^2+z^2=1 & (2) \\ x^3+y^3+z^3=-1 & (3) \end{cases}$$

$$(1) + (2) \Rightarrow x(x+1) + y(y+1) + z(z+1) = 0$$

$$\begin{cases} x=0 & \text{I. } (0; 0; 0) & \text{II. } (-1; -1; -1) \\ x=-1 & 0^3+0^3+0^3 \neq -1 & (-1)^3+(-1)^3+(-1)^3 = -1 \\ y=0 & \text{III. } (-1; 0; 0), (0; -1; 0), (0; 0; -1) \\ y=-1 & 0^3+(-1)^3+0 = -1 & \\ z=0 & \text{IV. } (-1; -1; 0), (-1; 0; -1), (0; -1; -1) \\ z=-1 & (-1)^3+0+(-1)^3 = -1 & \end{cases}$$

Ответ: $(-1; 0; 0), (0; -1; 0), (0; 0; -1)$

158

N 2.

Поскольку на одно место претендует один человек, то кол-во мест в проектах равно кол-ву людей $\Rightarrow$ переход в др проект разработчик не может сделать ни с др разработчиком. Вы получают либо выигрыш в з/п, либо нет в интересности задач. $\Rightarrow$ не существует такой пары проектов, где у одного из проектов з/п выше, и задачи интереснее.

Ответ: нет, такого быть не могло

05

N 3.

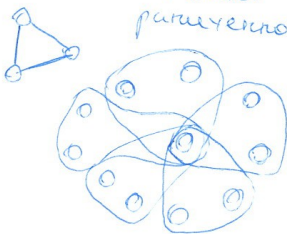
Пусть расстояние до дома первого равно S_1 . Необходимо выбрать такое место, что расстояние до домов второго и третьего не превосходит $2S_1$ и $3S_1$ соотв, иначе не имеет смысла выбирать это место, т.к. выбранное первым время будет компенсировано лишним временем второго и третьего.

Ответ: такое место, что $\min S_1 : S_2 \leq 2S_1$ и $S_3 \leq 3S_1$

06

N 4.

Если устройства из разных сетей не могут быть соединены напрямую, то такая сеть может быть только одна. Иначе, сеть может быть неограниченного кол-ва.



06

Оценка за задачу №1: 15

Комментарий к решению: задача решена верно.

Оценка за задачу №2: 0

Комментарий к решению: задача решена неверно.

Оценка за задачу №3: 0

Комментарий к решению: задача решена неверно.

Оценка за задачу №4: 0

Комментарий к решению: задача решена неверно.

Информатика

Задача 1.1:

```
#include <iostream>
#include <vector>
#include <map>

#define fi first
#define se second

using namespace std;

typedef long long ll;

bool comp(pair <ll, ll> p1, pair <ll, ll> p2) {
    return p1.fi < p2.fi;
}

int main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    ll n, k;
    cin >> n >> k;
    map <ll, ll> m;
    ll prev = 0, pw = 0;
    while (n--) {
        ll cw, cid;
        cin >> cid >> cw;
        m[cid] = prev + pw;
        prev = m[cid];
        pw = cw;
    }
    while (k--) {
        ll id;
        cin >> id;
        cout << m[id] << "\n";
    }
    system("pause");
    return 0;
}
```

Оценка за задачу №1.1: 7

Комментарий к решению: задача решена верно.

Задача 1.2:

```

#include <iostream>
#include <vector>
#include <map>

#define fi first
#define se second

using namespace std;

typedef long long ll;

bool comp(pair <ll, ll> p1, pair <ll, ll> p2) {
    return p1.fi < p2.fi;
}

int main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    ll n, k;
    cin >> n >> k;
    map <ll, ll> m;
    ll prev = 0, pw = 0;
    while (n--) {
        ll cw, cid;
        cin >> cid >> cw;
        m[cid] = prev + pw;
        prev = m[cid];
        pw = cw;
    }
    while (k--) {
        ll id;
        cin >> id;
        cout << m[id] << '\n';
    }
    system("pause");
    return 0;
}

```

Оценка за задачу №1.2: 13

Комментарий к решению: задача решена верно.

Задача 2.1:

```

#include <bits/stdc++.h>

```

```

#define fi first
#define se second

using namespace std;

typedef long long ll;

int main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    ll n, t;
    cin >> n >> t;
    vector <pair<ll, ll>> tasks(n);
    for (int i = 0; i < n; i++)
        cin >> tasks[i].first >> tasks[i].second;
    int dp[1010][1010];
    memset(dp, 0, sizeof(dp));
    for (ll i = 1; i <= n; i++)
        for (ll j = 1; j <= t; j++) {
            dp[i][j] = dp[i - 1][j];
            if (j - tasks[i - 1].fi >= 0 && dp[i - 1][j - tasks[i - 1].first] + tasks[i - 1].second > dp[i][j])
                dp[i][j] = dp[i - 1][j - tasks[i - 1].first] + tasks[i - 1].se;
        }
    cout << dp[n][t];
    system("pause");
    return 0;
}

```

Оценка за задачу №2.1: 7

Комментарий к решению: задача решена верно.

Задача 2.2:

```
#include <bits/stdc++.h>

#define fi first
#define se second

using namespace std;

typedef long long ll;

int main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    ll n, t;
    cin >> n >> t;
    vector <pair<ll, ll>> tasks(n);
    for (int i = 0; i < n; i++)
        cin >> tasks[i].first >> tasks[i].second;
    int dp[1010][1010];
    memset(dp, 0, sizeof(dp));
    for (ll i = 1; i <= n; i++)
        for (ll j = 1; j <= t; j++) {
            dp[i][j] = dp[i - 1][j];
            if (j - tasks[i - 1].fi >= 0 && dp[i - 1][j - tasks[i - 1].first] + tasks[i - 1].second > dp[i][j])
                dp[i][j] = dp[i - 1][j - tasks[i - 1].first] + tasks[i - 1].se;
        }
    cout << dp[n][t];
    system("pause");
    return 0;
}
```

Оценка за задачу №2.2: 13

Комментарий к решению: задача решена верно.

Задача 3.1:

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

#define fi first
#define se second

using namespace std;

typedef long long ll;

bool comp(pair <ll, ll> p1, pair <ll, ll> p2) {
    return p1.se < p2.se;
}

int main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    ll n, k;
    cin >> n >> k;
    if (k == 1) {
        ll sum = 0;
        while (n--) {
            ll x;
            cin >> x;
            sum ^= x;
        }
        cout << sum;
    }
    system("pause");
    return 0;
}
```

Оценка за задачу №3.1: 7

Комментарий к решению: задача решена верно.

Задача 3.2:

```
#include <bits/stdc++.h>

using namespace std;

int main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    int n, k;
    cin >> n >> k;
    unordered_map <int, int> mp;
    while (n--) {
        ll x;
        cin >> x;
        mp[x]++;
    }
    vector <ll> out;
    for (auto key : mp)
        if (key.second % 2)
            out.push_back(key.first);
    cout << min(out[1], out[0]) << ' ' << max(out[1], out[0]);
    system("pause");
    return 0;
}
```

Оценка за задачу №3.2: 0

Комментарий к решению: задача решена неверно.

Командная часть

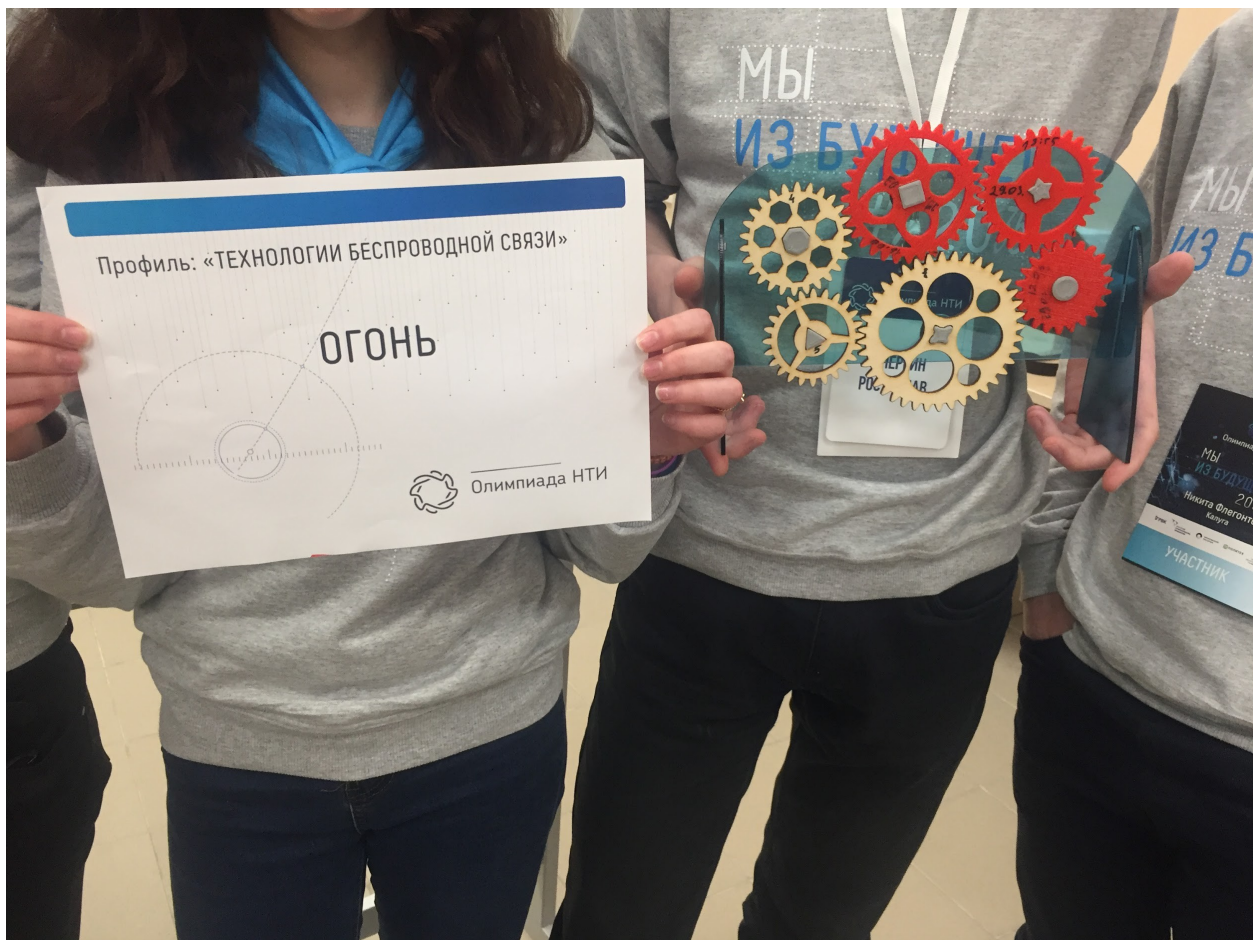
Результаты были получены в рамках выступления команды: Огонь

Личный состав команды:

- Густырь Валерия Александровна
- Епишина Анастасия Витальевна
- Ильина Елизавета Евгеньевна
- Кочергин Ростислав Алексеевич
- Флегонтов Никита
- Чемерев Илья Денисович

Описание решений командной задачи заключительного этапа:

1. Фото результата:



2. Код решения:

`h_utils.py`

```

import random

# длина блока кодирования
CHUNK_LENGTH = 8

# проверка длины блока
assert not CHUNK_LENGTH % 8, 'Длина блока должна быть кратна 8'

# вычисление контрольных бит
CHECK_BITS = [i for i in range(1, CHUNK_LENGTH + 1) if not i & (i - 1)]

def to_bin(chars):
    """
    Преобразование символов в бинарный формат
    """
    return ".join([bin(byte)[2:].zfill(8) for byte in chars])

def chunk_iterator(text_bin, chunk_size=CHUNK_LENGTH):
    """
    Поблочный вывод бинарных данных
    """
    for i in range(len(text_bin)):
        if not i % chunk_size:
            yield text_bin[i:i + chunk_size]

def get_check_bits_data(value_bin):
    """
    Получение информации о контрольных битах из бинарного блока данных
    """
    check_bits_count_map = {k: 0 for k in CHECK_BITS}
    for index, value in enumerate(value_bin, 1):
        if int(value):
            bin_char_list = list(bin(index)[2:].zfill(8))
            bin_char_list.reverse()
            for degree in [2 ** int(i) for i, value in enumerate(bin_char_list) if int(value)]:
                check_bits_count_map[degree] += 1
    check_bits_value_map = {}
    for check_bit, count in check_bits_count_map.items():
        check_bits_value_map[check_bit] = 0 if not count % 2 else 1
    return check_bits_value_map

```

```

def set_empty_check_bits(value_bin):
    """
    Добавить в бинарный блок "пустые" контрольные биты
    """
    for bit in CHECK_BITS:
        value_bin = value_bin[:bit - 1] + '0' + value_bin[bit - 1:]
    return value_bin

def set_check_bits(value_bin):
    """
    Установить значения контрольных бит
    """
    value_bin = set_empty_check_bits(value_bin)
    check_bits_data = get_check_bits_data(value_bin)
    for check_bit, bit_value in check_bits_data.items():
        value_bin = '{0}{1}{2}'.format(
            value_bin[:check_bit - 1], bit_value, value_bin[check_bit:])
    return value_bin

def get_check_bits(value_bin):
    """
    Получить информацию о контрольных битах из блока бинарных данных
    """
    check_bits = {}
    for index, value in enumerate(value_bin, 1):
        if index in CHECK_BITS:
            check_bits[index] = int(value)
    return check_bits

def exclude_check_bits(value_bin):
    """
    Исключить информацию о контрольных битах из блока бинарных данных
    """
    clean_value_bin = ""
    for index, char_bin in enumerate(list(value_bin), 1):
        if index not in CHECK_BITS:
            clean_value_bin += char_bin

    return clean_value_bin

def check_and_fix_error(encoded_chunk):

```

```
"""
```

Проверка и исправление ошибки в блоке бинарных данных

```
"""
```

```
check_bits_encoded = get_check_bits(encoded_chunk)
check_item = exclude_check_bits(encoded_chunk)
check_item = set_check_bits(check_item)
check_bits = get_check_bits(check_item)
if check_bits_encoded != check_bits:
    invalid_bits = []
    for check_bit_encoded, value in check_bits_encoded.items():
        if check_bits[check_bit_encoded] != value:
            invalid_bits.append(check_bit_encoded)
    num_bit = sum(invalid_bits)
    encoded_chunk = '{0}{1}{2}'.format(
        encoded_chunk[:num_bit - 1],
        int(encoded_chunk[num_bit - 1]) ^ 1,
        encoded_chunk[num_bit:])
    return encoded_chunk
```

```
def get_diff_index_list(value_bin1, value_bin2):
```

```
"""
```

Получить список индексов различающихся битов

```
"""
```

```
diff_index_list = []
for index, char_bin_items in enumerate(zip(list(value_bin1), list(value_bin2)), 1):
    if char_bin_items[0] != char_bin_items[1]:
        diff_index_list.append(index)
return diff_index_list
```

```
def encode(source):
```

```
"""
```

Кодирование данных

```
"""
```

```
text_bin = to_bin(source)
result = ""
for chunk_bin in chunk_iterator(text_bin):
    chunk_bin = set_check_bits(chunk_bin)
    result += chunk_bin
return result
```

```
def decode(encoded, fix_errors=True):
```

```
"""
```

Декодирование данных

"""

```
fixed_encoded_list = []
for encoded_chunk in chunk_iterator(encoded, CHUNK_LENGTH +
len(CHECK_BITS)):
    if fix_errors:
        encoded_chunk = check_and_fix_error(encoded_chunk)
        fixed_encoded_list.append(encoded_chunk)

clean_chunk_list = []
for encoded_chunk in fixed_encoded_list:
    encoded_chunk = exclude_check_bits(encoded_chunk)
    clean_chunk_list.append(encoded_chunk)
ans = bytearray(b'')
for clean_chunk in clean_chunk_list:
    for clean_char in [clean_chunk[i:i + 8] for i in range(len(clean_chunk)) if not i %
8]:
        ans.append(int(clean_char, 2))
return ans
```

humming_encode.py

```
import h_utils

def byte_to_char(s):
    c = chr(int(s, 2))
    return c

inp = open("gear6.7z", "br")
outp = open("encoded.txt", "bw")

cur = inp.read(2)
while cur:
    byt = bytearray(b'')
    if len(cur) == 1:
        cur = b"\x00"
    st = h_utils.encode(cur)
    byt.append(int(st[:8], 2))
    byt.append(int(st[8:16], 2))
    byt.append(int(st[16:24], 2))
    cur = inp.read(2)
    outp.write(byt)
```

```
outp.close()
inp.close()
```

humming_decode.py

```
import h_utils

inp = open("encoded_inv.txt", "rb")
outp = open("out.7z", "wb")

cur = inp.read(3)

while len(cur):
    out = h_utils.decode(h_utils.to_bin(cur))
    outp.write(out)
    cur = inp.read(3)

inp.close()
outp.close()
```

invert.py

```
inp = open("encoded1.txt", "br")
outp = open("encoded_inv.txt", "bw")

fin = inp.read(1024)

switch = 0

while fin:
    if switch == 0:
        outp.write(fin)
    else:
        out = bytearray(b"")
        for byte in fin:
            out.append(255 - int(byte))
        outp.write(out)
    fin = inp.read(1024)
    switch = (switch + 1) % 2

outp.close()
inp.close()
```

3. Протокол действий

Имя файла	Размер, КВ	MD5	Имя сохр. файла	MD5 сохр. файла	Начало	Окончание	Длительность	Скорость, kbit/s
output.txt	9.9	44ebd774a49e2a7d474a6fd572d7e3d2	output.txt	09aaf02a3ca7a075315b626ecb2032f0	2017-03-26 11:38:03	2017-03-26 11:38:03	00:00:00	MAX
test.txt	0.8	2548646e95f29f68b9c2dffcf19136c54	test.txt	0a2a11d278b1409b18b08674ae68ea5d	2017-03-26 12:29:57	2017-03-26 12:29:57	00:00:00	MAX
gear_6_out.txt	54.3	7d93ab89cbc53aaf9613001923767859	gear_6_out.txt	343835cb06dad2cac084b555ffdf2e3a1	2017-03-26 14:11:47	2017-03-26 14:11:54	00:00:07	62.1
gear_6_one_in.txt	54.3	5a3eda3598a17d407e916e52d8412673	gear_6_one_in.txt	6419f4765ec60530e0c8594cb039ea46	2017-03-26 15:07:39	2017-03-26 15:07:46	00:00:07	62.1
gear_6_ml.txt	69.0	f5a2c395e8096852bff89fc749ef030f	gear_6_ml.txt	bde66603f79b219c8fa4a358a3ad9e	2017-03-26 15:07:55	2017-03-26 15:08:04	00:00:09	61.3
gear_6_one_in_3.txt	54.3	5a3eda3598a17d407e916e52d8412673	gear_6_one_in_3.txt	417023f771ab16b849239f106066b034	2017-03-26 15:32:43	2017-03-26 15:32:43	00:00:15	29.0

						2:58		
gear_6_one ln_2.txt	54.3	5a3eda3598a17 d407e916e52d8 412673	gear_6_on eln_2.txt	b0acc0289b7d4 d723e862a02f2 7fc9d6	2017- 03-26 15:33: 05	201 7- 03- 26 15:3 3:21	00: 00: 16	27.1
gear_6_one ln_1.txt	54.3	5a3eda3598a17 d407e916e52d8 412673	gear_6_on eln_1.txt	b61e508c0e870 8c7b7c2aa03cef 1b466	2017- 03-26 15:33: 36	201 7- 03- 26 15:3 3:51	00: 00: 15	29.0
gear_6_ml _3.txt	69.0	f5a2c395e8096 852bff89fc749e f030f	gear_6_ml _3.txt	01a9612d82c6b db852c42142c8 48f713	2017- 03-26 15:34: 01	201 7- 03- 26 15:3 4:20	00: 00: 19	29.1
gear_6_ml _2.txt	69.0	f5a2c395e8096 852bff89fc749e f030f	gear_6_ml _2.txt	3b96037d8f408 84ed1077ca34c 99f678	2017- 03-26 15:34: 29	201 7- 03- 26 15:3 4:47	00: 00: 18	30.7
gear_6_ml _1.txt	69.0	f5a2c395e8096 852bff89fc749e f030f	gear_6_ml _1.txt	a3a462ba64576 3c7b0365f61d7 92fa9f	2017- 03-26 15:34: 57	201 7- 03- 26 15:3 5:17	00: 00: 20	27.6
gear_6_hou t.txt	790.3	a898741268f32 5ea0816ba9ff9 1955b9	gear_6_ho ut.txt	b715225ff8238 b97d8c8fa6178 6dc3c6	2017- 03-27 17:15: 18	201 7- 03- 27 17:1 7:01	00: 01: 43	61.4
gear_6_hou t.bin	790.3	a898741268f32 5ea0816ba9ff9 1955b9	gear_6_ho ut.bin	52c4232f4d9dbf 744fbd8cf39fcb dc77	2017- 03-27 17:41: 04	201 7- 03- 27 17:4 2:46	00: 01: 42	62.0
gear_	127.1	92cd51620d8de	gear_	f1eb06d97ac74	2017-	201	00:	63.5

6_hou t.txt		b03ec19875185 dc1062	6_ho ut(1). txt	804d0d541d852 6cd8c5	03-28 15:05: 24	7- 03- 28 15:0 5:40	00: 16	
gear_ 6_fou t.txt	98.7	e3284614135a6 4c8cfbab92ef3a 7a58d	gear_ 6_fou t.txt	9f6e03ea7d634 d7753aaa89175 7d2375	2017- 03-28 16:41: 09	201 7- 03- 28 16:4 1:22	00: 00: 13	60.7
gear_ 6_fou t.txt	65.9	6f23abbc78fcd 4ad57d0780be5 59aa9d	gear_ 6_fou t(1).t xt	20ea0391758af 925f1190e7810 60b64c	2017- 03-28 17:08: 55	201 7- 03- 28 17:0 9:04	00: 00: 09	58.6
gear_ 6_fou t.txt	65.9	afdfc4da2b34b ab910b993777 6997301	gear_ 6_fou t(2).t xt	4452e4726556c ca8cd9b37e215 2a0497	2017- 03-28 17:49: 26	201 7- 03- 28 17:4 9:35	00: 00: 09	58.6
gear_ 6.zip	53.7	fadc88b36de88 17037273fbc64 729e09	gear_ 6.zip	9b0ac73b6d14f eba5a3ae3c0a3c f42d6	2017- 03-29 10:23: 38	201 7- 03- 29 10:2 3:53	00: 00: 15	28.6
encod ed.txt	11.1	5b00adfc8214 c4cec49fcecfa e5cf1	encod ed.txt	e51711e5e0244 d3b525dad6a39 df7434	2017- 03-29 11:26: 57	201 7- 03- 29 11:2 6:57	00: 00: 00	MA X
encod ed.txt	11.1	5b00adfc8214 c4cec49fcecfa e5cf1	encod ed(1). txt	6e7a04aff9e1ad 1d1f1db698c24 bf6d4	2017- 03-29 11:34: 55	201 7- 03- 29 11:3 4:55	00: 00: 00	MA X
encod ed.txt	20.3	12752fb8452fb 820ea707ae427 ed43ef	encod ed(2). txt	8bd784769dd5c c8ba5c3d8fda59 c7232	2017- 03-29 12:35:	201 7- 03-	00: 00: 01	162. 4

					25	29 12:3 5:26		
encod ed.txt	27.7	90025092682c 2ec6a70a653c8 40b51dd	encod ed(3). txt	1616bade058dd 77f1c4ffc26fe1f f2ca	2017- 03-29 13:44: 29	201 7- 03- 29 13:4 4:32	00: 00: 03	73.9
encod ed.txt	27.8	011146c118ce2 0efd25718b35a bb5fc0	encod ed(4). txt	eeCBC276b4933 5966cc84e3a8e bcc079	2017- 03-29 14:04: 00	201 7- 03- 29 14:0 4:03	00: 00: 03	74.1

Общее время: 00:06:40