

# Второй отборочный этап

## Индивидуальная часть

### Программирование

#### *Задача П.1.1.1. Непонятные оценки (10 баллов)*

Предполагаемая роль: программист электронных устройств.

Зачастую числа, с которыми работают программы, являются недостаточно наглядными для пользователей. Данное задание является одним из случаев, с которыми приходится сталкиваться при реализации систем с широким охватом пользователей.

Для умного города выбирают компанию, которая будет разрабатывать беспилотные автобусы. Для этого все компании-участники представили свои прототипы, а жюри оценивало проделанную работу каким-либо целым неотрицательным количеством баллов. Члены жюри сложили все свои оценки, чтобы получить итоговую оценку прототипа. Однако из-за большого количества членов жюри числа получились слишком крупными для комфортного восприятия.

Ваша задача состоит в том, чтобы сделать оценки более удобными для восприятия человеком. Это подразумевает под собой выполнение следующих пунктов:

- Все оценки должны оставаться целыми числами.
- Соотношения между оценками разных прототипов не должны измениться.
- Числа должны быть минимально возможными с учетом выполнения всех остальных пунктов.
- Если оценки упростить не получается, то следует оставить их в первоначальном виде.

#### *Формат входных данных*

В первой строке идет целое число  $N$  — количество оцененных прототипов ( $1 \leq N \leq 500$ ).

В следующей строке находятся  $N$  оценок компонентов до приведения к удобному виду. Каждая оценка не превышает 10 000 000 000 баллов и больше или равна 0.

#### *Формат выходных данных*

Строка с  $N$  целыми числами — оценками компонентов после приведения к удобному виду в соответствующем порядке.

*Примеры**Пример №1*

<b>Стандартный ввод</b>
5 10 20 30 40 50
<b>Стандартный вывод</b>
1 2 3 4 5

*Пример №2*

<b>Стандартный ввод</b>
4 16 0 12 8
<b>Стандартный вывод</b>
4 0 3 2

*Решение*

```

1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <iostream>
4  #include <vector>
5  #include <algorithm>
6
7  #include <stdio.h>
8  #include <stdint.h>
9
10 int64_t gcd(int64_t a, int64_t b) {
11     while (b) {
12         a %= b;
13         std::swap(a, b);
14     }
15     return a;
16 }
17
18 int main(int argc, const char** argv) {
19     int n = 0;
20     std::cin >> n;
21
22     std::vector<int64_t> grades(n);
23     for (int i = 0; i < n; ++i) {
24         std::cin >> grades[i];
25     }
26
27     int64_t global_gcd = grades[0];
28     for (int i = 1; i < n; ++i) {
29         global_gcd = gcd(global_gcd, grades[i]);
30     }
31
32     if (!global_gcd) {
33         global_gcd = 1;
34     }
35

```

```
36     for (int i = 0; i < n; ++i) {
37         std::cout << grades[i] / global_gcd;
38         if (i != n - 1) {
39             std::cout << " ";
40         }
41     }
42
43     return 0;
44 }
```

### ***Задача II.1.1.2. Тесты. (10 баллов)***

Предполагаемая роль: программист взаимодействия электронных устройств.

Тестирование — один из основных этапов жизненного цикла ПО, и в его процессе возникает множество различных непростых задач. Ниже представлена реальная задача с небольшими упрощениями из этой области.

Для города разрабатывают новую систему. У компании есть  $N$  версий продукта, каждую из которых автоматически протестировали.

Компоненты, в которых тесты нашли не более  $K$  ошибок, считаются стабильными. Тем не менее, отчеты о каждой ошибке следует хранить для дальнейшего разбирательства в причинах. Каждый отчет об **одной** ошибке весит 1 МБ.

Требуется написать программу, которая будет подсчитывать суммарный объем отчетов для всех компонентов, признанных стабильными.

#### ***Формат входных данных***

В первой строке находится числа  $N$  и  $Q$  ( $1 \leq N \leq K \leq 10000$ ), которые обозначают количество компонентов и количество запросов к системе соответственно.

В следующей строке идет  $N$  чисел, разделенных пробелами, которые обозначают количество проваленных тестов для каждого из компонентов. Количество ошибок — целое неотрицательное число, меньше или равное 100000.

Далее следует  $Q$  строк с запросами. Каждый запрос представляет собой число  $K$  — максимальное количество ошибок в компоненте, при котором он может считаться стабильным.

#### ***Формат выходных данных***

$Q$  строк с ответами на запросы. Каждый ответ содержит суммарный объем отчетов об ошибках в стабильных компонентах в МБ (без указания единиц измерения).

#### ***Примеры***

##### *Пример №1*

Стандартный ввод
10 3
2 6 8 4 5 1 9 0 3 7
3
8
0
Стандартный вывод
6
36
0

*Пример №2*

Стандартный ввод
9 1
1 1 1 4 1 0 1 1 1
1
Стандартный вывод
7

*Решение*

```

1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <iostream>
4  #include <vector>
5  #include <algorithm>
6  #include <string>
7  #include <utility>
8
9  #include <stdio.h>
10
11 int get_last_idx_in_array(std::vector<int>& data, int value) {
12     int64_t low = 0;
13     int64_t high = data.size() - 1;
14
15     while (low <= high) {
16         int64_t mid = (low + high + 1) >> 1;
17         int mid_val = data[mid];
18
19         if (mid_val < value)
20             low = mid + 1;
21         else if (mid_val > value)
22             high = mid - 1;
23         else {
24             if (high != mid)
25                 low = mid;
26             else
27                 return mid;
28         }
29     }
30     return low;
31 }
32
33 int main(int argc, const char** argv) {
34     int n = 0;

```

```

35     int q = 0;
36     std::cin >> n >> q;
37
38     std::vector<int> comp(n);
39     for (int i = 0; i < n; ++i) {
40         std::cin >> comp[i];
41     }
42     std::sort(comp.begin(), comp.end());
43
44     std::vector<int> sum(n + 1);
45     sum[0] = 0;
46     for (int i = 1; i <= n; ++i) {
47         sum[i] = sum[i - 1] + comp[i - 1];
48     }
49
50     for (int i = 0; i < q; ++i) {
51         int k = 0;
52         std::cin >> k;
53
54         int stable_count = get_last_idx_in_array(comp, k);
55         if (stable_count == n) {
56             std::cout << sum[n] << std::endl;
57             continue;
58         }
59
60         if (k == comp[stable_count]) {
61             ++stable_count;
62         }
63         std::cout << sum[stable_count] << std::endl;
64     }
65 }

```

### ***Задача II.1.1.3. Переводчик К.И.Р. (10 баллов)***

Предполагаемая роль: программист верхнего уровня.

Не секрет, что взаимодействие между программами по сети осуществляется с помощью данных, структурированных по определенным правилам (например, JSON, XML, YAMЛ и огромное число других). Часто эти данные нужно сконвертировать из одного формата в другой. Об этом и будет следующая задача.

Сервисы в умном городе обмениваются данными между собой. Однако сервисов стало много, и форматов сообщений — тоже. Более того, системы зачастую даже не давали информации о своем формате данных. Налаживание обмена информацией становилось все затруднительнее, пока не внедрили систему Кодировщик Интегрированных Разметок Автоматический (сокращенно К.И.Р.). К.И.Р. принимает сообщения и переводит их в понятный для получателя вид.

Список языков разметки и их элементы представлен в таблице ниже. Строкой ТЕХТ может быть любая последовательность символов английского алфавита или пробелов.

жирный шрифт	<b>**ТЕХТ**</b>	{ТЕХТ}}	.ТЕХТ.	##ТЕХТ#
курсивный шрифт	*ТЕХТ*	{ТЕХТ}	..ТЕХТ.	#ТЕХТ##
жирный курсивный шрифт	<b>***ТЕХТ***</b>	{ТЕХТ}}}	...ТЕХТ.	###ТЕХТ#
подчеркнутый	<b>*****ТЕХТ*****</b>	{ТЕХТ}}}}	....ТЕХТ.	#ТЕХТ###
зачеркнутый	<b>****ТЕХТ****</b>	{ТЕХТ}}}}}	.....ТЕХТ.	###ТЕХТ###

Известно, что сообщения содержат только строчные и заглавные буквы английского алфавита и пробелы, а также элементы разметки (которые могут состоять и из других символов). Известно, что элементы разметки не могут стоять рядом друг с другом и гарантированно отделены хотя бы 1 символом (например, в разметке `brace` гарантированно не встретится последовательность `{}}`, но может встретиться `{ }`). Также гарантируется отсутствие вложенных конструкций разметки.

Вам требуется написать программу, которая переводит сообщения к нужному виду. Стоит отметить, что изменениям должны подвергнуться только символы, относящиеся к элементам разметки.

### *Формат входных данных*

На вход программа получает одно из названий разметки (`star`, `brace`, `dot` или `sharp`), в которую необходимо перевести строку.

Во второй строке находится сообщение для перевода в какой-либо одной кодировке. Длина строки не превышает 1000 символов.

### *Формат выходных данных*

Строка с переведенным сообщением.

### *Примеры*

#### *Пример №1*

<b>Стандартный ввод</b>
<code>brace</code> <code>hello **guys**</code>
<b>Стандартный вывод</b>
<code>hello {guys}}</code>

### *Решение*

```

1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <iostream>
4  #include <vector>
5  #include <algorithm>
6  #include <string>
7  #include <utility>
8

```

```

9  #include <stdio.h>
10
11 static inline bool is_letter_or_space(const char c) {
12     return ((c >= 'a') && (c <= 'z')) ||
13            ((c >= 'A') && (c <= 'Z')) ||
14            (c == ' ');
15 }
16
17 enum md_type {
18     STAR,
19     BRACE,
20     DOT,
21     SHARP
22 };
23
24 enum elem_type {
25     BOLD,
26     ITALIC,
27     BOLD_ITALIC,
28     UNDERLINED,
29     CROSSED
30 };
31
32 struct element {
33     md_type    markdown;
34     std::string prefix;
35     std::string postfix;
36     elem_type  type;
37 };
38
39 static std::vector<element> convert_table = {
40     { STAR, "**",    "**",    BOLD },
41     { STAR, "*",    "*",    ITALIC },
42     { STAR, "****", "****", BOLD_ITALIC },
43     { STAR, "*****", "*****", UNDERLINED },
44     { STAR, "*****", "*****", CROSSED },
45
46     { BRACE, "{", "}" ,    BOLD },
47     { BRACE, "{", "}" ,    ITALIC },
48     { BRACE, "{", "}" ,    BOLD_ITALIC },
49     { BRACE, "{", "}" ,    UNDERLINED },
50     { BRACE, "{", "}" ,    CROSSED },
51
52     { DOT, ".",    ".",    BOLD },
53     { DOT, "..",   ".",    ITALIC },
54     { DOT, "...",  ".",    BOLD_ITALIC },
55     { DOT, "....", ".",    UNDERLINED },
56     { DOT, ".....", ".",    CROSSED },
57
58     { SHARP, "##",  "#",    BOLD },
59     { SHARP, "#",   "##",   ITALIC },
60     { SHARP, "###", "#",    BOLD_ITALIC },
61     { SHARP, "#",   "###",  UNDERLINED },
62     { SHARP, "###", "###",  CROSSED }
63 };
64
65 void convert(const md_type target,
66             std::string& prefix,
67             std::string& postfix) {
68     elem_type type = BOLD;

```

```

69     for (const auto& el : convert_table) {
70         if (el.prefix == prefix && el.postfix == postfix) {
71             type = el.type;
72             break;
73         }
74     }
75     for (const auto& el : convert_table) {
76         if (el.markdown == target && el.type == type) {
77             prefix = el.prefix;
78             postfix = el.postfix;
79             return;
80         }
81     }
82 }
83
84 int main(int argc, const char** argv) {
85     std::string target_markdown;
86     std::string in;
87     std::cin >> target_markdown;
88     md_type target = target_markdown == "star" ? STAR :
89         target_markdown == "brace" ? BRACE :
90         target_markdown == "dot" ? DOT :
91         SHARP;
92
93     std::getline(std::cin >> std::ws, in);
94
95     std::string out;
96     out.reserve(in.size());
97
98     size_t i = 0;
99     while (i < in.size()) {
100         if (is_letter_or_space(in[i])) {
101             out += in[i];
102             ++i;
103         } else {
104             // parse markdown
105             size_t j = i;
106
107             std::string md_prefix;
108             while (!is_letter_or_space(in[j])) {
109                 md_prefix += in[j];
110                 ++j;
111             }
112
113             std::string text;
114             while (is_letter_or_space(in[j])) {
115                 text += in[j];
116                 ++j;
117             }
118
119             std::string md_postfix;
120             while (!is_letter_or_space(in[j])) {
121                 md_postfix += in[j];
122                 ++j;
123                 if (j == in.size()) {
124                     break;
125                 }
126             }
127             convert(target, md_prefix, md_postfix);
128             out += md_prefix + text + md_postfix;

```



```

129         i = j;
130     }
131 }
132
133     std::cout << out;
134     return 0;
135 }

```

## Схемотехника

### 8–9 класс

#### Задача II.1.2.1. 5 ламп и конденсатор (20 баллов)

Пять лампочек и конденсатор подключены к постоянному источнику питания, как показано на рис. II.1.1. Известно, что сопротивления ламп практически не зависят от подаваемого на них напряжения и равны  $R_1 = 1000$  Ом,  $R_2 = 200$  Ом,  $R_3 = 40$  Ом. Емкость конденсатора  $C = 13$  мФ. В некоторый момент времени ключ  $K$  размыкают. Считая, что заряд на конденсаторе, замкнутом на некоторое сопротивление  $R$ , убывает по закону  $q = q_0 e^{-\frac{t}{\tau}}$  (где  $q_0$  — первоначальный заряд на конденсаторе,  $\tau = RC$  — время релаксации), необходимо найти, через сколько секунд мощность, потребляемая системой ламп, уменьшится в 10 раз. Ответ округлить до десятых.

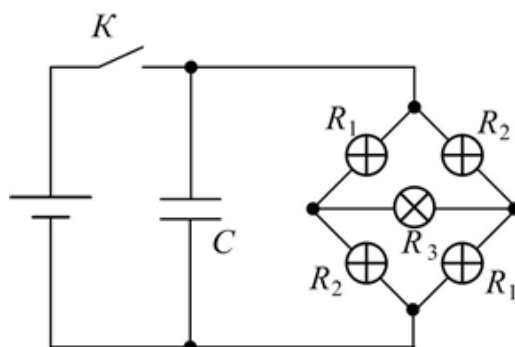


Рис. II.1.1: Схема подключения ламп и конденсатора к источнику питания

Ссылка на рисунок: <https://bit.ly/34UomnL>

Ссылка на pdf: <https://drive.google.com/file/d/12aeChZEPPUjOM-YqPZSNrtoaw8hxm1d-/view?usp=sharing>

### Решение

Из условия мощность, потребляемая системой ламп (участок  $AB$  на рис. II.1.2), должна уменьшиться в 10 раз:

$$\frac{P_{AB}}{P'_{AB}} = \eta = 10, \quad (\text{II.1.1})$$

где  $P_{AB}$  и  $P'_{AB}$  — мощности, потребляемые системой ламп в начальный момент времени и спустя время  $t$  после размыкания ключа  $K$ . Мощность может быть найдена как:

$$P_{AB} = \frac{U_{AB}^2}{R_{л}},$$

где  $U_{AB}$  и  $R_{л}$  — падение напряжения и общее сопротивление системы ламп, соответственно. Найдем  $R_{л}$ . В системе ламп расставим токи с учетом симметрии схемы как показано на рис. II.1.2.

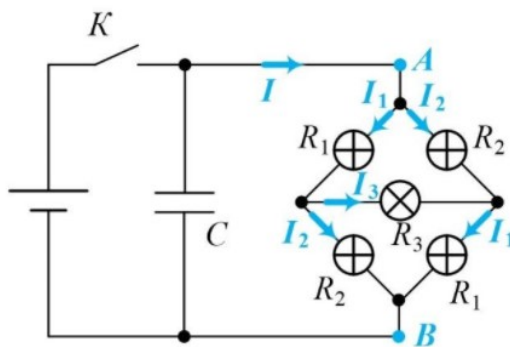


Рис. II.1.2: Схема подключения ламп и конденсатора к источнику питания с указанием направлений токов в цепи

Согласно первому и второму правилам Кирхгофа:

$$I = I_1 + I_2,$$

$$I_1 = I_2 + I_3, \quad (\text{II.1.2})$$

$$R_1 I_1 + R_3 I_3 - R_2 I_2 = 0. \quad (\text{II.1.3})$$

Общее сопротивление  $R_{л}$  системы ламп может быть найдено как:

$$R_{л} = \frac{U_{AB}}{I} = \frac{R_1 I_1 + R_2 I_2}{I_1 + I_2} = \frac{R_1 \frac{I_1}{I_2} + R_2}{\frac{I_1}{I_2} + 1}$$

Отношение  $\frac{I_1}{I_2}$  найдем из системы уравнений (II.1.2) и (II.1.3):

$$\frac{I_1}{I_2} = \frac{R_2 + R_3}{R_1 + R_3}$$

тогда:

$$R_{л} = \frac{R_1 \frac{R_2 + R_3}{R_1 + R_3} + R_2}{\frac{R_2 + R_3}{R_1 + R_3} + 1} = \frac{R_3(R_1 + R_2) + 2R_1 R_2}{R_1 + R_2 + 2R_3} = 350 \text{ Ом}$$

Напряжение  $U_{AB}$  после того, как разомкнули ключ  $K$ , будет равно согласно второму правилу Кирхгофа напряжению на обкладках конденсатора, которое определяется зарядом  $q_t$  в данный момент времени  $t$  на обкладках и емкостью  $C$ :

$$U_{AB} = \frac{q_t}{C} = \frac{q_0}{C} \exp\left(-\frac{t}{R_{л}C}\right)$$

Подставляем все полученные параметры системы в формулу (II.1.1).

$$\eta = \frac{\frac{U_{AB}^2}{R_{\pi}}}{\frac{U_{AB}^2}{R_{\pi}}} = \frac{U_{AB}^2}{U_{AB}^2}$$

или:

$$U_{AB} = \sqrt{\eta} U'_{AB}$$

Здесь  $U_{AB}$  и  $U'_{AB}$  — падения напряжения на участке  $AB$  в начальный момент времени  $t_0 = 0$  и спустя время  $t$  после размыкания ключа  $K$ , соответственно.

$$\frac{q_0}{C} = \sqrt{\eta} \frac{q_0}{C} \exp\left(-\frac{t}{R_{\pi}C}\right)$$

$$\exp\left(-\frac{t}{R_{\pi}C}\right) = \sqrt{\eta}$$

Откуда искомое время:

$$t = \frac{1}{2} R_{\pi} C \cdot \ln(\eta) \approx 5,2 \text{ с}$$

**Ответ:** 5.2 с.

## 10–11 класс

### Задача II.1.2.2. 5 ламп и реостат (20 баллов)

Пять лампочек и реостат подключены к постоянному источнику питания с внутренним сопротивлением  $r = 1100$  Ом, как показано на рис. II.1.3. Известно, что сопротивления ламп практически не зависят от подаваемого на них напряжения и равны  $R_1 = 1000$  Ом,  $R_2 = 200$  Ом,  $R_3 = 25$  Ом. В некоторый момент времени сопротивление на реостате ( $R_0 = 1200$  Ом) начинают уменьшать с постоянной скоростью  $\alpha = 80$  Ом/с. Найти, через сколько секунд мощность, потребляемая системой ламп, уменьшится в 5 раз. Ответ округлить до десятых.

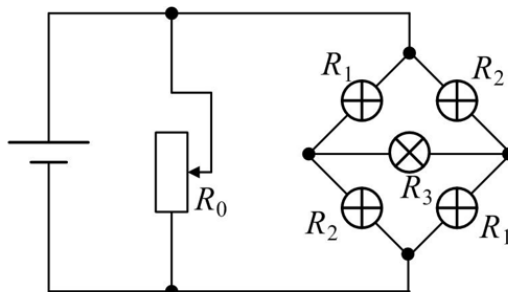


Рис. II.1.3: Схема подключения ламп и реостата к источнику питания

Ссылка на 3d-модель: <https://bit.ly/3kQGzI1>

Ссылка на pdf: <https://drive.google.com/file/d/1ys5aWuuLdGjS2noBz5vdSpW95kdxkiBU/view?usp=sharing>

### Решение

Из условия мощность, потребляемая системой ламп (участок АВ на рис. II.1.4), должна уменьшиться в 5 раз:

$$P_{AB}P'_{AB} = \eta = 5, \quad (\text{II.1.4})$$

где  $P_{AB}$  и  $P'_{AB}$  — мощности, потребляемые системой ламп в начальный момент времени и спустя время  $t$  как начали уменьшать сопротивление на реостате, соответственно. Мощность может быть найдена как

$$P_{AB} = U_{AB}^2 R_{л},$$

где  $U_{AB}$  и  $R_{л}$  — падение напряжения и общее сопротивление системы ламп, соответственно. Найдем  $R_{л}$ . В системе ламп расставим токи с учетом симметрии схемы как показано на рис. II.1.4.

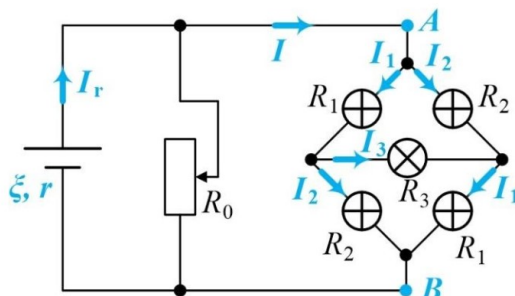


Рис. II.1.4: Схема подключения ламп и реостата к источнику питания с указанием направлений токов в цепи

Согласно первому и второму правилам Кирхгофа:

$$I = I_1 + I_2,$$

$$I_1 = I_2 + I + 3, \quad (\text{II.1.5})$$

$$R_1 I_1 + R_3 I + 3 - R_2 I_2 = 0. \quad (\text{II.1.6})$$

Общее сопротивление  $R_{л}$  системы ламп может быть найдено как:

$$R_{л} = \frac{U_{AB}}{I} = \frac{R_1 I_1 + R_2 I_2}{I_1 + I_2} = \frac{R_1 \frac{I_1}{I_2} + R_2}{\frac{I_1}{I_2} + 1}$$

Отношение  $\frac{I_1}{I_2}$  найдем из системы уравнений (II.1.5) и (II.1.6):

$$\frac{I_1}{I_2} = \frac{R_2 + R_3}{R_1 + R_3}$$

тогда:

$$R_{л} = \frac{R_1 \frac{R_2 + R_3}{R_1 + R_3} + R_2}{\frac{R_2 + R_3}{R_1 + R_3} + 1} = \frac{R_3(R_1 + R_2) + 2R_1 R_2}{R_1 + R_2 + 2R_3} = 344 \text{ Ом}$$

Напряжение  $U_{AB}$  найдем из уравнения, составленного согласно второму правилу Кирхгофа для внешнего контура на рис. II.1.4.

$$rI_r + U_{AB} = \xi,$$

в котором  $I_r$  выразим из закона Ома для полной цепи через общее сопротивление цепи  $R_{\text{общ}}$ :

$$R_{\text{общ}} = r + \frac{R_t R_{\text{л}}}{R_t + R_{\text{л}}}$$

$$I_r = \frac{\xi}{R_{\text{общ}}}$$

где  $R_t = R_0 - \alpha t$  — сопротивление реостата в произвольный момент времени  $t$ . Получаем:

$$U_{AB} = \frac{\xi(R_{\text{общ}} - r)}{R_{\text{общ}}}$$

Подставляем все полученные параметры системы в формулу (II.1.4):

$$\eta = \frac{\frac{U_{AB}^2}{R_{\text{л}}}}{\frac{U_{AB}^2}{R_{\text{л}}}} = \frac{U_{AB}^2}{U_{AB}^2}$$

или:

$$U_{AB} = \sqrt{\eta} U'_{AB}$$

Здесь  $U_{AB}$  и  $U'_{AB}$  — падения напряжения на участке АВ в начальный момент времени и спустя время  $t$  как начали уменьшать сопротивление на реостате, соответственно.

$$d \frac{\xi(r + \frac{R_0 R_{\text{л}}}{R_0 + R_{\text{л}}} - r)}{r + \frac{R_0 R_{\text{л}}}{R_0 + R_{\text{л}}}} = \sqrt{\eta} \frac{\xi(r + \frac{R_t R_{\text{л}}}{R_t + R_{\text{л}}} - r)}{r + \frac{R_t R_{\text{л}}}{R_t + R_{\text{л}}}}$$

$$\frac{R_0}{R_0 r + R_{\text{л}} r + R_0 R_{\text{л}}} = \sqrt{\eta} \frac{R_t}{R_t r + R_{\text{л}} r + R_t R_{\text{л}}}$$

Из последнего уравнения выражаем  $R_t$ :

$$R_t = \frac{r R_{\text{л}}}{\sqrt{\eta} \frac{R_0 r + R_{\text{л}} r + R_0 R_{\text{л}}}{R_0} - (R_{\text{л}} + r)} \approx 152 \text{ Ом}$$

Тогда искомое время:

$$t = \frac{R_0 - R_t}{\alpha} = 13,1 \text{ с}$$

**Ответ:** 13,1.

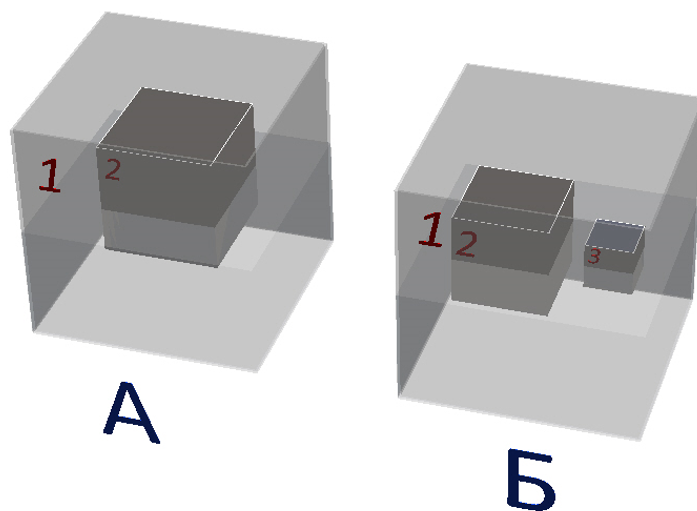
## Классическая механика

### 8–9 класс

#### Задача II.1.3.1. Три ящика (20 баллов)

Три титановых ящика ( $\rho_T = 4540 \text{ кг/м}^3$ ) вложены друг в друга. Ящики имеют форму открытых сверху кубов и герметичны снизу и с боков. В первый — самый

большой — налили 50 л воды ( $\rho_B = 1000 \text{ кг/м}^3$ ) так, что второй ящик стал плавать вместе с вложенным в него третьим, масса которого равна 3 кг. Определите, как и на сколько изменится уровень воды в первом ящике, если вынуть третий ящик из второго и опустить его в воду рядом со вторым. Ребро первого ящика (внутреннее) имеет длину 50 см, а ребро третьего (внешнее) — в 5 раз меньше. Донные грани ящиков параллельны. Ответ привести в мм, округлить до десятых.



**Пример ввода ответа:**

- если уровень воды увеличился на 60.29 мм, введите 60,3;
- если уровень воды уменьшился на 480.11 мм, введите -480,1.

Ссылка на 3d-модель: <https://bit.ly/38olMIx>

Ссылка на pdf: [https://drive.google.com/file/d/1r-GJdfXVt6vmS6F9WBJ1qzpc\\_\\_Y1VWlW/view?usp=sharing](https://drive.google.com/file/d/1r-GJdfXVt6vmS6F9WBJ1qzpc__Y1VWlW/view?usp=sharing)

**Решение**

Пусть  $H$  — уровень воды в первом ящике после того, как в него налили  $V_0$  воды и второй ящик с третьим внутри всплыл. Тогда:

$$a_1^2 H = V_0 + V_{23},$$

где  $a_1$  — внутренняя сторона первого ящика,  $V_{23}$  — объем, вытесненный вторым ящиком вместе с третьим внутри. Так как второй и третий ящики находятся в покое, то сила тяжести уравновешивается силой Архимеда:

$$\rho_B g V_{23} = (m_2 + m_3)g,$$

где  $m_2$  и  $m_3$  — массы второго и третьего ящиков,  $\rho_B$  — плотность воды, а  $g$  — ускорение свободного падения. Из этих формул можно выразить начальный уровень воды:

$$H = \frac{V_0 + \frac{(m_2 + m_3)}{\rho_B}}{a_1^2}.$$

Рассмотрим теперь случай, когда третий ящик вынули из второго и опустили в воду. Оценим среднюю плотность третьего ящика с воздухом внутри:

$$\rho_3 = \frac{m_3}{a_3^3} = \frac{3}{\left(\frac{0,5}{5}\right)^3} = 3000 \text{ кг/м}^3,$$

где  $a_3$  — внешняя сторона третьего ящика. Так как  $\rho_3 > \rho_{\text{в}}$  то третий ящик будет стоять на дне. Если бы второго и третьего ящика не было, то уровень воды был бы равен:

$$H'' = \frac{V_0}{a_1^2} = \frac{50 \cdot 10^{-3}}{0,5^2} = 0,2 \text{ м.}$$

При погружении в воду любых предметов ее уровень будет только расти. Следовательно, третий ящик, высота которого  $a_3 = \frac{0,5}{5} = 0,1 < H''$ , будет полностью скрыт водой и она залется внутрь него. Результирующий уровень воды  $H'$  после этого будет определяться уравнением:

$$a_1^2 H' = V_0 + V_2' + V_3',$$

где  $V_2', V_3'$  — объемы, вытесненные вторым и третьим ящиками соответственно. Объем  $V_2'$  можно найти из условия равновесия второго ящика:

$$\rho_{\text{в}} g V_2' = m_2 g.$$

$V_3'$  является объемом титановых стенок, поэтому:

$$V_3' = \frac{m_3}{\rho_{\text{т}}}.$$

Следовательно:

$$H' = \frac{V_0 + \frac{m_2}{\rho_{\text{в}}} + \frac{m_3}{\rho_{\text{т}}}}{a_1^2}.$$

Выразим изменение уровня:

$$\Delta H = H' - H = \frac{m_3(\rho_{\text{в}} - \rho_{\text{т}})}{a_1^2 \rho_{\text{т}} \rho_{\text{в}}} = \frac{3 \cdot 1000 - 4540}{0,5^2 \cdot 1000 \cdot 4540} \approx -0,00936 \text{ м} \approx -9,4 \text{ мм}$$

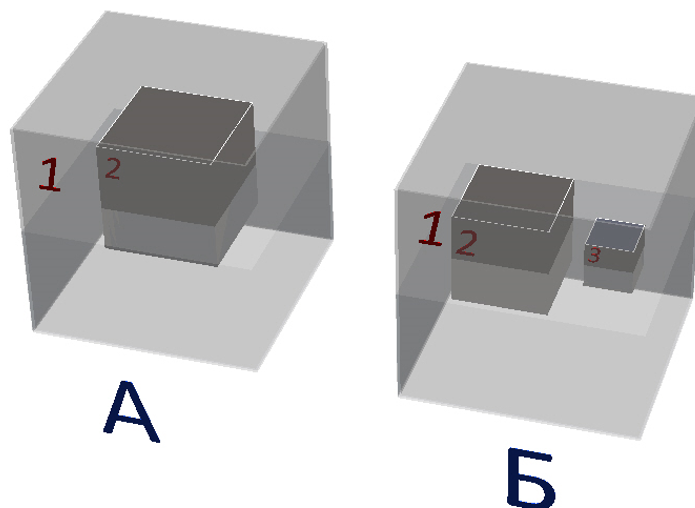
Таким образом, уровень уменьшится на 9,4 мм.

**Ответ:** -9,4.

## 10–11 класс

### Задача П.1.3.2. Три ящика (20 баллов)

Три титановых ящика ( $\rho_{\text{т}} = 4540 \text{ кг/м}^3$ ) вложены друг в друга. Ящики имеют форму открытых сверху кубов и герметичны снизу и с боков. В первый — самый большой — налили 22 л воды ( $\rho_{\text{в}} = 1000 \text{ кг/м}^3$ ). Определите, как и на сколько изменится уровень воды в первом ящике, если вынуть третий ящик из второго и опустить его в воду рядом со вторым. Ребро первого ящика (внутреннее) имеет длину 50 см, а ребра второго и третьего (внешние) меньше в 2 и 5 раз соответственно. Масса второго ящика равна 5 кг, масса третьего равна 3 кг. Донные грани ящиков параллельны. Ответ привести в мм, округлить до десятых.



### Пример ввода ответа:

- если уровень воды увеличился на 60.29 мм, введите 60,3;
- если уровень воды уменьшился на 480.11 мм, введите -480,1.

Ссылка на 3d: <https://bit.ly/2IfFRGb>

Ссылка на pdf: <https://drive.google.com/file/d/1SpfeqjI16IZd9Rr4FwfMzhc6FsPOhCN/view?usp=sharing>

### Решение

Рассмотрим сначала третий ящик внутри второго. Предположим, что второй ящик плавает после того, как в первый налили воду. Пусть  $h$  — уровень воды в первом ящике, тогда:

$$a_1^2 h = V_0 + a_2^2 h_2,$$

где  $a_1$  — внутренняя сторона первого ящика,  $a_2$  — внешняя сторона второго ящика,  $h_2$  — глубина погружения второго ящика, а  $a_2^2 h_2$  — объем, вытесненный вторым ящиком. Так как второй ящик находится в покое, то сила тяжести уравновешивается силой Архимеда:

$$\rho_{\text{в}} g (a_2^2 h_2) = (m_2 + m_3) g,$$

где  $m_2$  и  $m_3$  — массы второго и третьего ящиков,  $\rho_{\text{в}}$  — плотность воды, а  $g$  — ускорение свободного падения. Выразим из этих уравнений глубину погружения и уровень воды:

$$h_2 = \frac{m_2 + m_3}{\rho_{\text{в}} a_2^2} = \frac{5 + 3}{1000 \cdot (0,5/2)^2} = 0,128 \text{ м}$$

$$h = \frac{V_0 + (m_2 + m_3)/\rho_{\text{в}}}{a_1^2} = \frac{22 \cdot 10^{-3} + (5 + 3)/1000}{0,5^2} = 0,12 \text{ м.}$$

Таким образом,  $h_2 > h$  и второй ящик должен стоять на дне, а не плавать. Обозначим  $H$  уровень воды в первом ящике в этом случае, тогда:

$$a_1^2 H = V_0 + a_2^2 H,$$

откуда:

$$H = \frac{V_0}{a_1^2 - a_2^2} = \frac{22 \cdot 10^{-3}}{0,5^2 - (0,5/2)^2} \approx 0,117333 \text{ м.}$$



Найдем уровень воды  $h'$ , когда третий ящик уже вынули из второго, но еще не опустили в воду. Предположим, что второй ящик всплыл, тогда ситуация отличается от рассмотренной в самом начале только суммарной массой и, следовательно, имеем:

$$h'_2 = \frac{m_2}{\rho_{\text{в}} a_2^2} = \frac{5}{1000 \cdot (0,5/2)^2} = 0,08 \text{ м},$$

$$h' = \frac{V_0 + m_2/\rho_{\text{в}}}{a_1^2} = \frac{22 \cdot 10^{-3} + 5/1000}{0,5^2} = 0,108 \text{ м},$$

где  $h'_2$  — глубина погружения второго ящика в этом случае. Так как  $h'_2 < h'$ , то второй ящик действительно будет плавать после того, как из него вынут третий.

Оценим среднюю плотность третьего ящика с воздухом внутри:

$$\rho_3 = \frac{m_3}{a_3^3} = \frac{3}{(0,5/5)^3} = 3000 \text{ кг/м}^3,$$

где  $a_3$  — внешняя сторона третьего ящика. Так как  $\rho_3 > \rho_{\text{в}}$  то третий ящик будет стоять на дне. При погружении в воду любых предметов ее уровень будет только расти. Следовательно, третий ящик, высота которого  $a_3 = 0,5/5 = 0,1 < h'$ , будет полностью скрыт водой и она залется внутрь него.

Результирующий уровень воды  $H'$  после этого будет определяться уравнением:

$$a_1^2 H' = V_0 + a_2^2 h'_2 + V'_3,$$

где  $a_2^2 h'_2$ ,  $V'_3$  — объемы, вытесненные вторым и третьим ящиками соответственно. Объем  $V'_3$  является объемом титановых стенок, поэтому:

$$V'_3 = \frac{m_3}{\rho_{\text{т}}}.$$

Следовательно:

$$H' = \frac{V_0 + m_2/\rho_{\text{в}} + m_3/\rho_{\text{т}}}{a_1^2} = \frac{22 \cdot 10^{-3} + \frac{5}{1000} + \frac{3}{4540}}{0,5^2} \approx 0,110643 \text{ м}.$$

Выразим изменение уровня:

$$\Delta H = H' - H = 0,110643 - 0,117333 = -0,006690 \text{ м} \approx -6,7 \text{ мм}$$

Таким образом, уровень уменьшится на 6,7 мм.

**Ответ:** -6,7.

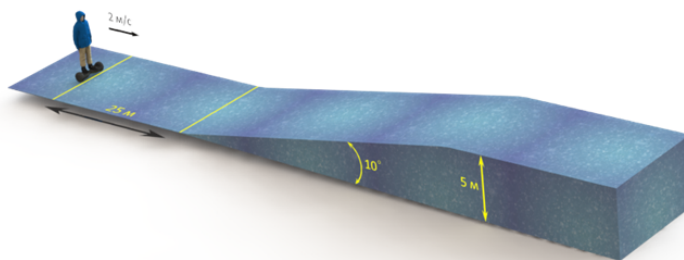
## Преобразование энергии

### 8–9 класс

#### Задача II.1.4.1. Зимний скутер (20 баллов)

Катаясь зимой на гироскутере, мальчик решил заехать на пологую горку высотой 5 м, склон которой можно представить в виде участка наклонной под углом

$10^\circ$  к горизонту плоскости. За 25 м до начала горки мальчик приступил к разгону по гладкой горизонтальной поверхности с начальной скоростью 2 м/с в сторону горки, причем колеса скутера начали пробуксовывать. Оцените, на сколько градусов нагреются резиновые шины скутера диаметром 25 см и суммарной массой 250 г за время разгона и подъема до максимальной высоты, если колеса пробуксовывали все это время с постоянной частотой 12 об/сек, а коэффициент трения скольжения равен 0,05. Суммарная масса мальчика и скутера равна 50 кг. Удельная теплоемкость материала шин 800 Дж/(кг·град). Передачей тепла пренебречь. При переходе от горизонтальной поверхности к наклонной скорость не теряется. Значение ускорения свободного падения принять равным  $9,81 \text{ м/с}^2$ . Ответ привести в кельвинах, округлить до десятых.



Ссылка на pdf: <https://drive.google.com/file/d/1tUhIM-tC5ThQ-DpDgs126xPBjRqSSMNR/view?usp=sharing>

### Решение

Так как колеса проскальзывают, то на скутер будет действовать сила трения скольжения помимо силы тяжести и силы реакции опоры (см. рис. II.1.5).

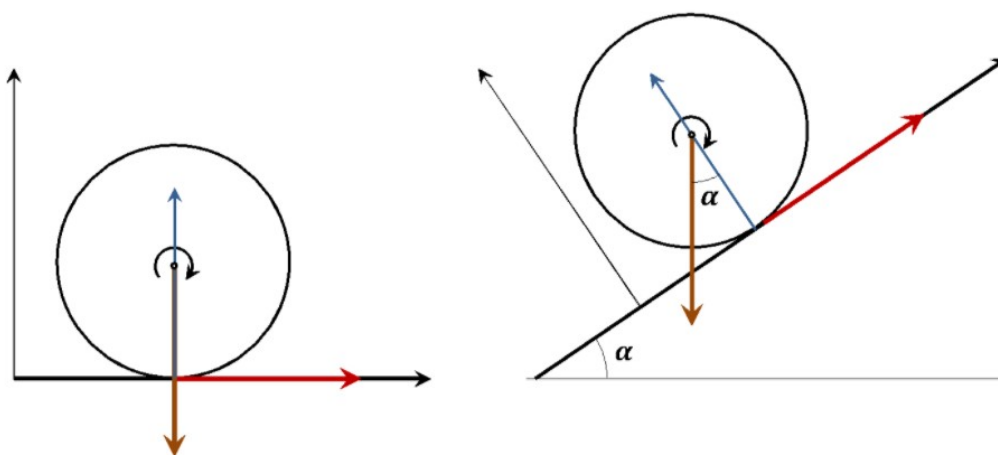


Рис. II.1.5: Силы, действующие на тело при движении по а) горизонтальной и б) наклонной поверхностям

Рассмотрим движение по горизонтальной поверхности. Направим ось  $x$  горизонтально, а ось  $y$  — вертикально. Запишем динамические уравнения в проекции на

оси:

$$OX : Ma_1 = F_{\text{тр}}$$

$$OY : 0 = N - Mg$$

где  $M$  — суммарная масса мальчика и скутера. Сила трения скольжения:

$$F_{\text{тр1}} = \nu N = \nu Mg,$$

Поэтому ускорение  $a_1 = \nu g$ . Запишем выражение для длины горизонтального участка:

$$l = \frac{v_1^2 - v_0^2}{2a_1}.$$

Здесь  $v_0$  — начальная скорость, а  $v_1$  — скорость в конце горизонтального участка. Выразим скорость  $v_1$ :

$$v_1 = \sqrt{v_0^2 + 2\nu gl}.$$

С другой стороны,  $v_1 = v_0 + a_1 t_1$ , где  $t_1$  — время движения по горизонтальной поверхности. Тогда:

$$t_1 = \frac{v_1 - v_0}{a_1} = \frac{\sqrt{v_0^2 + 2\nu gl} - v_0}{\nu g} \approx 6,81 \text{ с.}$$

Рассмотрим теперь движение по наклонной поверхности. Направим ось  $x'$  вдоль нее, а ось  $y'$  — перпендикулярно  $O'X'$ . Запишем динамические уравнения в проекции на оси:

$$O'X' : Ma_2 = F_{\text{тр2}} - Mgs\sin\alpha,$$

$$O'Y' : 0 = N_2 - Mg\cos\alpha,$$

Сила трения скольжения  $F_{\text{тр2}} = N_2 = \nu Mg\cos\alpha$ , следовательно, ускорение:

$$a_2 = g(\nu\cos\alpha - \sin\alpha).$$

Если склон достаточно длинный, то до остановки скутер пройдет расстояние:

$$s = \frac{0 - v_1^2}{2a_2} = \frac{v_0^2 + 2\nu gl}{g(\sin\alpha - \nu\cos\alpha)} = \frac{2^2 + 2 \cdot 0,05 \cdot 9,81 \cdot 25}{9,81\sin 10^\circ - 0,05\cos 10^\circ} = 11,87 \text{ м,}$$

где учтено, что скорость не теряется при переходе от горизонтальной поверхности к наклонной. Если высота горки  $H$ , то полная длина склона:

$$s' = \frac{H}{\sin\alpha} = \frac{H}{\sin 10^\circ} = 28,79 \text{ м.}$$

Так как  $s < s'$ , то скутер остановится на склоне, не поднявшись на вершину. Найдем время движения по склону до остановки:

$$t_2 = \frac{v_1}{a_2} = \frac{\sqrt{v_0^2 + 2\nu gl}}{g(\nu\cos\alpha - \sin\alpha)} \approx 4,38 \text{ с}$$

Нагрев колес определяется работой силы трения:

$$A_{\text{тр}} = Q = cm\Delta T,$$

где  $Q$  — количество теплоты,  $c$  — удельная теплоемкость,  $m$  — масса колес,  $\Delta T$  — изменение температуры. На каждом из участков сила трения постоянная, следовательно:

$$A_{\text{тр}} = A_1 + A_2 = F_{\text{тр1}}L_1 + F_{\text{тр2}}L_2,$$

где  $A_1$  и  $A_2$  — вклады горизонтального и наклонного участков, а  $L_1$  и  $L_2$  — пути, которые проходят колеса на этих участках. Так как колеса проскальзывают, то эти пути пропорциональны частоте вращения, времени движения и диаметру колес  $d$ :

$$L_1 = \pi d v t_1,$$

$$L_2 = \pi d v t_2.$$

Выразим изменение температуры:

$$\Delta T = \frac{A_{\text{гр}}}{cm} = \frac{\pi d v M g}{cm} (t_1 + \cos \alpha t_2)$$

Подстановка значений дает:

$$\Delta T = \frac{3,14159 \cdot 0,25 \cdot 12 \cdot 0,05 \cdot 50 \cdot 9,81}{800 \cdot 0,25} (6,81 + 4,38 \cos 10^\circ) \approx 12,9 \text{ К}$$

**Ответ:** 12,9.

## 10–11 класс

### Задача II.1.4.2. Суточная выработка (20 баллов)

Определите суточную выработку электроэнергии ветрогенератором, если зависимость скорости ветра от времени можно представить кусочно-линейной функцией (см. рис. II.1.6), генератор не вырабатывает энергию при скорости ветра  $v < v_m = 5$  м/с, а при  $v > v_m$  электрическую мощность генератора с учетом всех потерь можно представить следующим образом:

$$P(v) = P_0(v - v_m)$$

где  $P_0 = 250$  Вт · с / м. Координаты точек, в которых зависимость  $v = v(t)$  терпит излом приведены в таблице II.1.1 ниже. Ответ привести в кВт·час, округлить до целых.

Таблица II.1.1: Координаты точек

№ точки	1	2	3	4	5
t, ч	0	6	10	16	24
v, м/с	25	18	1	4	22

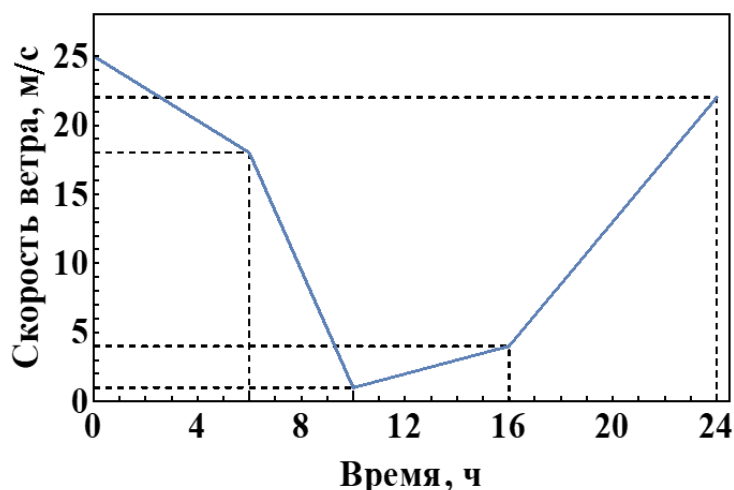


Рис. П.1.6: Зависимость скорости ветра от времени

Ссылка на 3d: <https://bit.ly/3eCK1E6>

Ссылка на pdf: <https://drive.google.com/file/d/1Dzt6cHUHLLCskdVAA3zWEvDNORB0rKvE/view?usp=sharing>



### Решение

Генератор не работает при скорости ветра  $v < v_m = 5$  м/с. Найдём моменты времени, соответствующие  $v = v_m$ . В первый раз функция  $v = v(t)$  пересекает прямую  $v = 5$  м/с при  $t = t'$  ( $6 \text{ ч} < t' < 10 \text{ ч}$ ), а во второй раз при  $t = t''$  ( $16 \text{ ч} < t'' < 24 \text{ ч}$ ). Уравнение прямой, проходящей через точки  $v_1, t_1$  и  $v_2, t_2$  можно записать следующим образом:

$$v = v_1 + \frac{(t - t_1)(v_2 - v_1)}{(t_2 - t_1)}.$$

Следовательно, имеем два уравнения для  $t'$  и  $t''$ .

$$5 = 18 + \frac{(t' - 6)(1 - 18)}{(10 - 6)},$$

$$5 = 4 + \frac{(t'' - 16)(22 - 4)}{(24 - 16)},$$

откуда получим  $t' = \frac{154}{17} \approx 9,06$  ч,  $t'' = \frac{148}{9} \approx 16,44$  ч. Мощность также как и скорость будет задаваться кусочно-линейной функцией. В интервале  $t' < t < t''$  она будет равна нулю, а в других точках ее можно вычислить по формуле:

$$P(v) = P_0(v - v_m).$$

Вычислим значения мощности в точках излома и запишем в таблицу.

№ точки	1	2	3	4	5
t, ч	0	6	9.06	16.44	24
P, кВт	5	3.25	0	0	4.25

График этой функции приведен на рис. II.1.7.

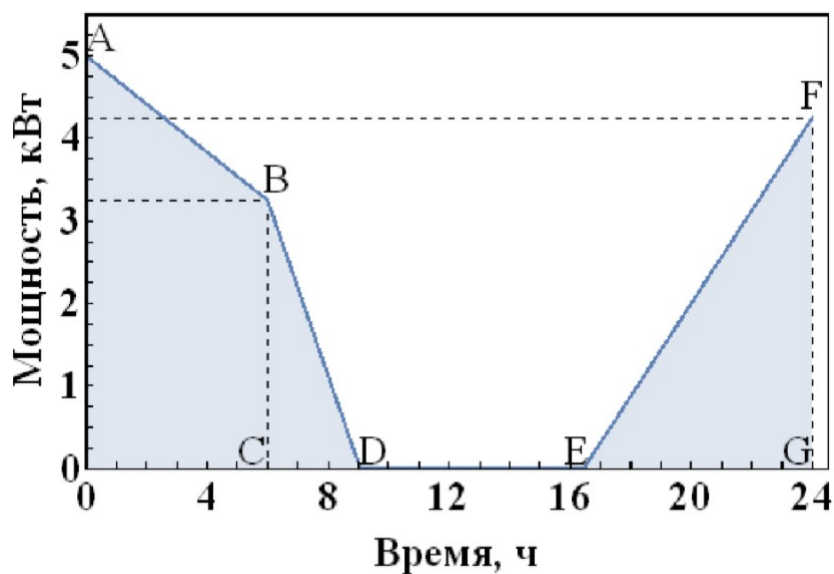


Рис. II.1.7: График функции

Суточная выработка энергии  $W$  определяется площадью под кривой  $P = P(t)$ . Ее можно найти как сумму площадей трапеции ABCO и треугольников BCD и EFG:

$$W = \left( 6 \cdot \frac{5 + 3,25}{2} + \frac{1}{2} \cdot 3,25 \cdot (9,06 - 6) + \frac{1}{2} \cdot 4,25 \cdot 24 - 16,44 \right) \approx 46 \text{ кВт} \cdot \text{час}$$

**Ответ:** 46.

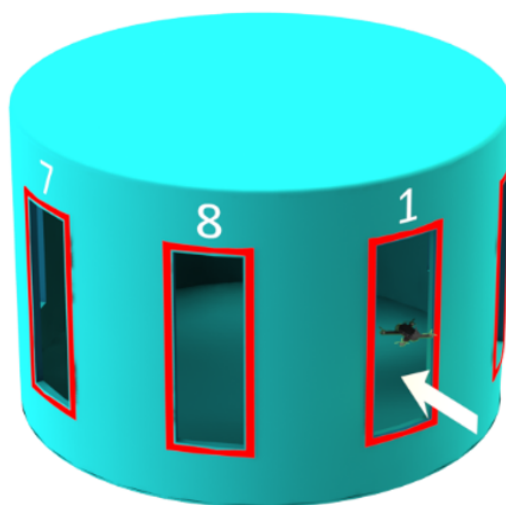
## Классическая механика

8–9 класс

### Задача П.1.5.1. Дрон-рейсинг (20 баллов)

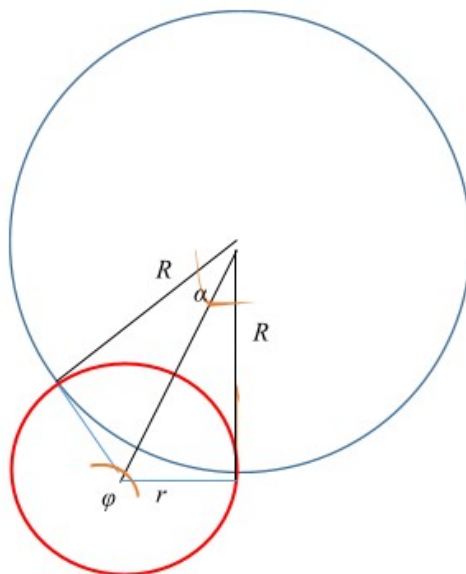
На соревнованиях по дрон-рейсингу необходимо провести квадрокоптер через барабан диаметром 5 м с восемью равномерно расположенными окнами. Дрон влетел внутрь барабана через окно номер 1 со скоростью 2,5 м/с перпендикулярно плоскости окна. Оцените ускорение, необходимое дрону, чтобы вылететь через окно номер 8, если дрон внутри барабана движется с постоянной по модулю скоростью по дуге окружности. Кроме того, определите время, затраченное на полет внутри барабана. Размеры окон и дрона считать достаточно близкими. Полученное ускорение привести в  $\text{м/с}^2$ , округлить до десятых. Время привести в секундах, округлить до сотых.

В качестве ответа вывести сначала ускорение без единиц измерения, а затем время, затраченное на полет внутри барабана, также без единиц измерения. Ответы отделить друг от друга пробелом, дробную часть отделять запятой. Если дробная часть равна 0, то ее все равно следует писать с требуемой точностью. Пример ответа для ускорения  $5,54 \text{ м/с}^2$  и времени 2,002 секунд: 5,5 2,00.



Ссылка на 3d-рисунок: <https://bit.ly/36i0FmJ>

Ссылка на pdf: [https://drive.google.com/file/d/1XP\\_N9TKKzzPIg0IHLWvtnPNBEJFXpwVH/view?usp=sharing](https://drive.google.com/file/d/1XP_N9TKKzzPIg0IHLWvtnPNBEJFXpwVH/view?usp=sharing)

*Решение*

Чтобы дрон вылетел из барабана, он должен двигаться по дуге радиуса  $r$ , а угол  $\alpha$  на рисунке должен быть равен  $\frac{\pi}{4}$ . Из прямоугольного треугольника:

$$\operatorname{tg} \frac{\alpha}{2} = \operatorname{tg} \frac{\pi}{8} = \frac{r}{R}.$$

Внутри барабана дрон движется постоянным ускорением:

$$a_{\text{ц}} = \frac{v^2}{r} = \frac{v^2}{R \operatorname{tg} \frac{\pi}{8}} = 6.04 \text{ м/с}^2.$$

Время движения можно найти из формулы для угла поворота  $\phi = \omega t = \frac{v}{r} t$ . Из рисунка найдем:

$$\phi = 2\pi - \frac{\pi}{4} - \pi = \frac{3\pi}{4}$$

Подставляем в формулу для угла и получаем ответ.

**Ответ:** 6,0 0,98.

**10–11 класс****Задача II.1.5.2. Дружные кубики (20 баллов)**

На столе лежат в ряд плотно прижатых друг к другу 6 одинаковых дубовых кубиков (плотность дуба  $\rho = 700 \text{ кг/м}^3$ , коэффициент трения друг о друга  $\nu = 0,26$ ). Какую минимальную силу  $F$  нужно приложить к центрам крайних граней кубиков (см. рисунок II.1.8), чтобы вся система кубиков отрывалась от стола как целое? Ребро кубиков имеет длину 5 см. Ускорение свободного падения  $g = 9,8 \text{ м/с}^2$ . Ответ дать в Ньютонах, округлить до десятых.



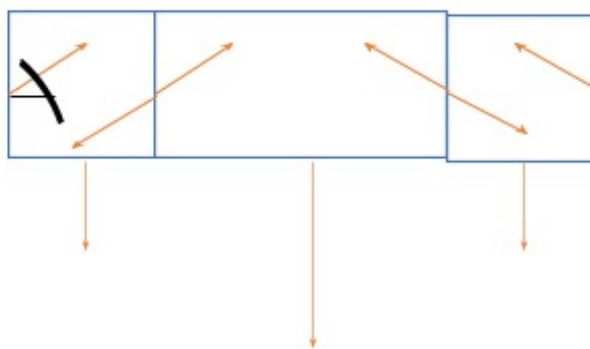


Рис. II.1.8: Кубики на столе

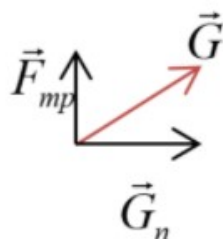
Ссылка на 3d: <https://bit.ly/3kfBsja>

Ссылка на pdf: <https://drive.google.com/file/d/1J1J7VyWdSTzq9ixBgsHk0LuyWVL5ooW/view?usp=sharing>

### Решение



Чтобы поднять кубики, 4 «внутренних» кубика должны двигаться вместе без проскальзывания относительно друг друга. Все силы взаимодействия между этими 4 кубиками в сумме дадут ноль. Поэтому 4 «внутренних» кубика можно заменить одним телом массой  $4m$ ,  $m$  — масса одного кубика. На рисунке показаны силы, действующие на каждый кубик.  $G$  — сила давления внешнего кубика на внутренний,  $R$  — сила реакции. По третьему закону Ньютона  $R = G$ ,  $\vec{R} \uparrow \downarrow \vec{G}$ . На внешние кубики действует сила  $F$ , сила реакции  $R$  и сила тяжести  $mg$ . На внутренние кубики действуют силы давления  $G$  и сила тяжести  $4mg$ . Сила давления  $D = F$ ,  $\vec{F} \uparrow \uparrow \vec{G}$ . Силу давления и реакции раскладываем на силу нормального давления и силу трения.



Сила  $F$  будет минимальна, если кубики будут двигаться равномерно. Тогда уравнение движения левого кубика:

$$F \sin \alpha = mg + F_{\text{тр}}$$

Уравнение движения внутренних кубиков:

$$2F_{\text{тр}} = 4mg$$

$$F_{\text{тр}} = \nu G_n = \nu F \cos \alpha$$

Решая систему уравнений, находим  $tg \alpha = \frac{3}{2} \nu$ .

$$F = \frac{2mg}{\nu \cos \alpha}$$

Ответ: 7,1.

## Теплотехника

### 8–9 класс

#### Задача II.1.6.1. Парилка в доме (20 баллов)

Парилка имеет размеры  $2 \text{ м} \times 4 \text{ м} \times 2,4 \text{ м}$  (см. рисунок II.1.9). Три стены парилки толщиной 30 см сделаны из пенобетона и деревянной обшивки (коэффициенты теплопроводности дерева и пенобетона считать одинаковыми и равными  $0,14 \text{ Вт}/(\text{м} \cdot \text{К})$ ). Одна из этих стен является внутренней и граничит с помещением с температурой воздуха  $t_1 = 24^\circ \text{C}$ . Две другие являются внешними (температура воздуха на улице  $t_2 = -30^\circ \text{C}$ ). Четвертая стена толщиной 3 см с выходом в помещение сделана из специального закаленного стекла, теплопроводность которого в диапазоне температур от  $0^\circ \text{C}$  до  $100^\circ \text{C}$  зависит от абсолютной температуры  $T$  (в Кельвинах) как  $(\alpha T + \beta)$ , где  $\alpha = 0,003 \text{ Вт}/(\text{м} \cdot \text{К}^2)$ ,  $\beta = 0,41 \text{ Вт}/(\text{м} \cdot \text{К})$ . Парилка отапливается печью с мощностью  $11,7 \text{ кВт}$ . Необходимо оценить максимальную температуру воздуха в парилке, которую можно поддерживать в данных условиях.

При решении задачи учесть, что теплопередача через стены осуществляется по закону:

$$\frac{Q}{\tau \cdot S} = \nu \frac{\Delta T}{h}$$

где  $Q$  — количество тепла, отводимое за время  $\tau$  через стену площадью  $S$  и толщиной  $h$ ;  $\nu$  — коэффициент теплопроводности стены;  $\Delta T$  — перепад температур по разные стороны стены. Считать, что температура стеклянной стены плавно меняется от температуры воздуха в парилке до температуры воздуха в помещении. Потолок и пол парилки считать полностью теплоизолированными. Ответ представить в градусах Цельсия и округлить до целого числа.

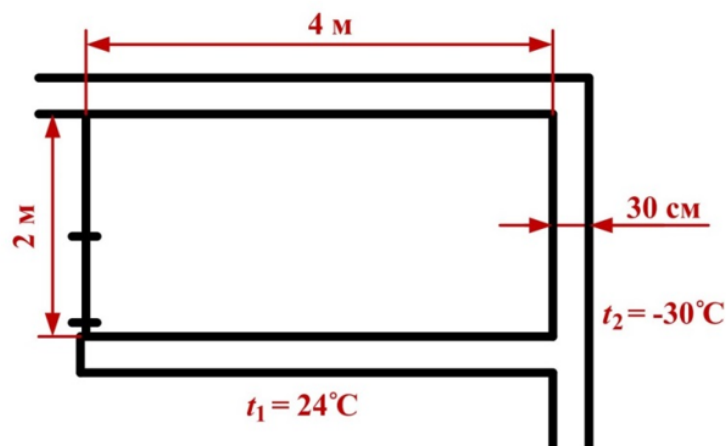
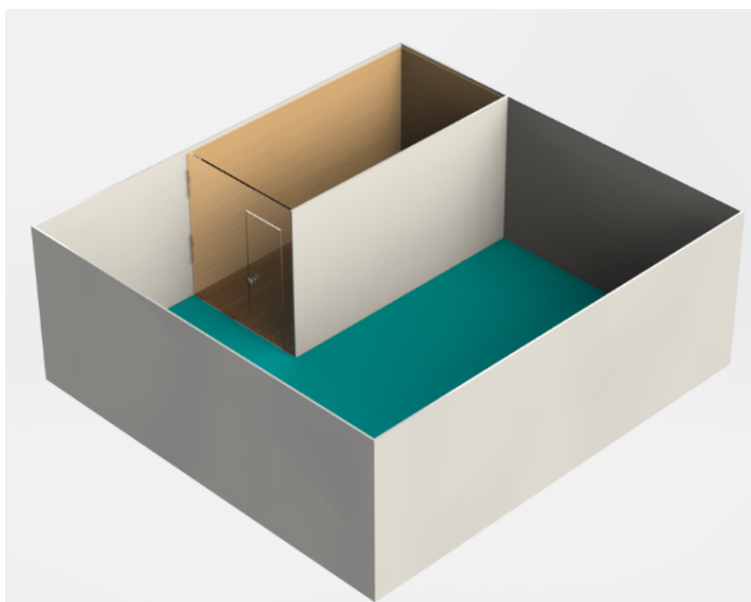


Рис. II.1.9: План парилки в помещении

Ссылка на pdf: <https://drive.google.com/file/d/1ImCCAo8o0ysb7QEHC8C4GK8axeyVAVUi/view?usp=sharing>



### Решение

Максимальная температура воздуха в парилке из условия теплового баланса будет соответствовать ситуации, когда все тепло  $Q$ , сообщаемое печью парилке в течение времени  $t$ , передается через ее стенки:

$$P = \frac{Q}{t}$$

Распишем поток тепла  $\frac{Q}{t}$  через стенки:

$$\frac{Q}{t} = \frac{Q_1}{t} + \eta \frac{T - T_1}{h} S_2 + \eta \frac{T - T_2}{h} S_3$$

где  $Q_1$  — количество тепла, отводимое за время  $t$  через стену из закаленного стекла;  $\eta$  — теплопроводность пенобетона/дерева;  $S_2 = bc$  — площадь внутренней стены из

пенобетона и деревянной обшивки,  $S_3 = (a + b)c$  — площадь внешних стен из пенобетона и деревянной обшивки;  $a = 2$  м,  $b = 4$  м,  $c = 2,4$  м — линейные размеры стен;  $h$  — толщина стен из пенобетона/дерева;  $T, T_1$  и  $T_2$  — абсолютные значения температур в парилке, помещении и на улице, соответственно. Расчет потока тепла  $\frac{Q_1}{t}$  через стену из закаленного стекла осложнен тем фактом, что температура внутри стекла меняется от температуры  $T$  воздуха в парилке до температуры  $T_1$  воздуха в помещении и коэффициент теплопроводности стены  $\eta_{ст}$  в разных ее частях различен. Рассмотрим теплопередачу через узкую часть стены толщиной  $\Delta x$  с перепадом температур  $\Delta T$ , учитывая, что температура внутри стенки меняется плавно:

$$\frac{Q_1}{t} = \eta_{ст} \frac{\Delta T}{\Delta x} S_1$$

где  $S_1 = ac$  — площадь стены из закаленного стекла. Обозначим величиной  $J_Q = \frac{Q_1}{tS_1}$  плотность потока тепла, которая в условиях равновесия является постоянной величиной, тогда:

$$J_Q \Delta x = \eta_{ст} \Delta T$$

Правая и левая часть этого уравнения при условии, что  $\Delta x$  является очень маленькой величиной, могут быть рассмотрены как площади узких областей под зависимостями  $\eta_{ст}(T)$  и  $J_Q(x)$ , соответственно (рис. II.1.10). Тогда для описания плотности потока тепла через всю стенку толщиной  $h_{ст}$ , необходимо найти площади под обеими зависимостями (площадь прямоугольника на рис. II.1.10а и площадь трапеции на рис. II.1.10б):

$$\begin{aligned} J_Q h_{ст} &= \frac{1}{2} (\eta_{ст}(T) + \eta_{ст}(T_1)) (T - T_1) = \frac{1}{2} (\alpha T + \beta + \alpha T_1 + \beta) (T - T_1) = \\ &= \left( \frac{T + T_1}{2} + \beta \right) (T - T_1) \end{aligned}$$

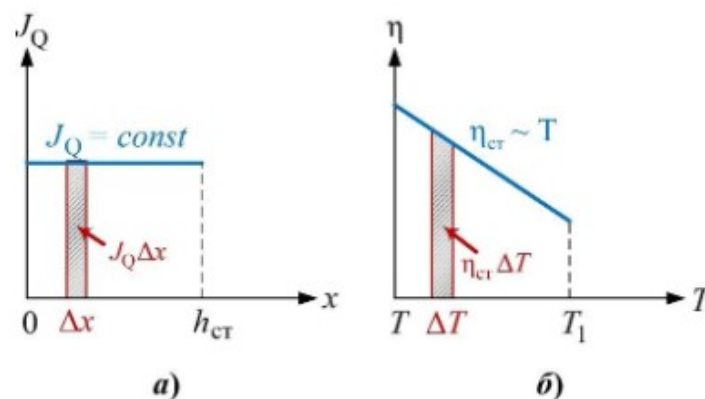


Рис. II.1.10: Зависимость: а)  $J_Q(x)$ ; б)  $\eta_{ст}(T)$ . На панели (б) учтено, что температура внутри стекла меняется от температуры  $T$  воздуха в парилке до температуры  $T_1$  воздуха в помещении

Так поток тепла через всю стену из закаленного стекла:

$$\frac{Q_1}{t} = \left( \alpha \frac{T + T_1}{2} + \beta \right) \frac{T - T_1}{h_{ст}} S_1$$

В итоге получаем:

$$P = \left( \alpha \frac{T + T_1}{2} + \beta \right) \frac{T - T_1}{h_{\text{ст}}} S_1 + \eta \frac{T - T_1}{h} S_2 + \eta \frac{T - T_2}{h} S_3$$

или:

$$T^2 \cdot \frac{\alpha S_1}{2h_{\text{ст}}} + T \cdot \left( \beta \frac{S_1}{h_{\text{ст}}} + \eta \frac{S_2 + S_3}{h} \right) - T_1^2 \frac{S_1}{2h_{\text{ст}}} - T_1 \left( \beta \frac{S_1}{h_{\text{ст}}} + \eta \frac{S_2}{h} \right) - T_2 \eta \frac{S_3}{h} - P = 0$$

Введя обозначения:

$$\frac{\alpha S_1}{2h_{\text{ст}}} = A = 0,24 \text{ Вт/К}^2$$

$$\beta \frac{S_1}{h_{\text{ст}}} + \eta \frac{S_2 + S_3}{h} = B = 76,8 \text{ Вт/К}$$

$$T_1^2 \cdot \frac{\alpha S_1}{2h_{\text{ст}}} + T_1 \cdot \left( \beta \frac{S_1}{h_{\text{ст}}} + \eta \frac{S_2}{h} \right) + T_2 \eta \frac{S_3}{h} + P = C = 55316,88 \text{ Вт}$$

имеем:

$$A \cdot T^2 + B \cdot T - C = 0$$

$$T = \frac{-B \pm \sqrt{B^2 + 4AC}}{2A}$$

$$T = -666 \text{ К}; T = 346 \text{ К} \rightarrow t = 73^\circ \text{С}$$

Температура  $T = -666 \text{ К}$  не подходит, так как абсолютное значение температуры не может быть отрицательным.

**Ответ:**  $73^\circ$ .

## 10–11 класс

### Задача II.1.6.2. Паровая пушка (20 баллов)

В прямую пробирку длиной 25 см налили немного воды и заткнули пробкой массой 3 г так, что пробка оказалась внутри (см. рис. II.1.11), и стали нагревать. Оцените максимальную скорость, которую может приобрести пробка, если процесс расширения газа считать адиабатическим, а силой тяжести, объемом жидкой воды и размерами пробки пренебречь. Сила трения скольжения пробки о пробирку постоянна и равна 5 Н. Для того, чтобы стронуть пробку с места необходимо приложить силу в 1,2 раза больше, чем сила трения. Считать, что газ внутри пробирки состоит из жестких многоатомных молекул. Ответ привести в м/с и округлить до десятых.

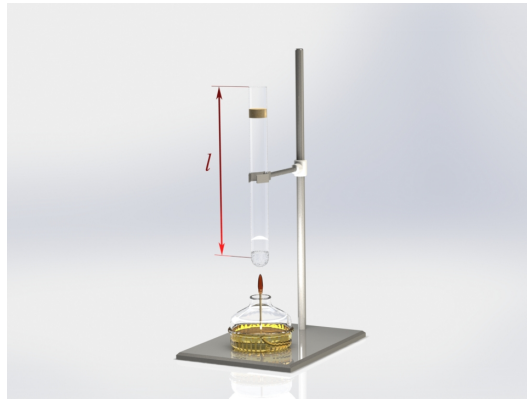


Рис. П.1.11: Установка с пробиркой

Ссылка на 3d: <https://bit.ly/363FR47>

Ссылка на pdf-файл: <https://drive.google.com/file/d/12LMS1WWawwNd6nnpGhTWM8TdYXzuu2ry/view?usp=sharing>

### Решение

При движении пробки на нее действуют сила трения скольжения и сила давления газа. Чтобы пробка начала движение, сила давления должна превысить силу трения в 1.2 раза. При расширении газа давление и сила давления будут уменьшаться. Следовательно, ускорение пробки тоже будет уменьшаться.

Скорость пробки будет максимальна, если в момент вылета пробки из пробирки ускорение пробки будет равно нулю, т. е. на всем пути она будет ускоряться. В этом случае сила давления газа равна силе трения скольжения:  $p_2 S = F_{\text{тр}}$ ,  $p_2$  — давление водяного пара в момент вылета пробки,  $S$  — площадь поперечного сечения. В момент начала движения  $p_1 S = 1.2 F_{\text{тр}}$ . Т.к. процесс расширения адиабатический,  $p_1 V_1^\gamma$ ,  $\gamma = 4/3$  — показатель адиабаты для многоатомного газа,  $V_1 = l_1 S$ ,  $V_2 = l S$  — объемы газа в момент в начальный момент движения пробки и в момент вылета из пробирки. Решая совместно уравнения, находим расстояние от пробки до дна пробирки в начальный момент времени  $l_1 = l \frac{1}{1.2^\gamma}$ .

Чтобы найти скорость пробки в момент вылета из пробирки используем закон изменения кинетической энергии:

$$\frac{mv^2}{2} = A$$

$A = A_{\text{тр}} + A_{\text{д}}$  — работа равнодействующей сил, которые действуют на пробку.

Работ силы трения  $A_{\text{тр}} = -F_{\text{тр}}(l - l_1)$ .

Работа силы давления газа — это работа идеального газа при адиабатическом расширении:

$$A_{\text{д}} = -\Delta U = \frac{i}{2} \nu R (T_1 - T_2)$$

где  $i = 6$  — число степеней свободы для одноатомного газа. Из уравнения состояния идеального газа  $\nu R T = p V$ .

$$A_{\text{д}} = \frac{i}{2} (p_1 V_1 - p_2 V_2)$$

Подставив в формулу давления и объемы, получаем ответ задачи ( $v = 2,84$  м/с).

**Ответ:** 3,2.

## Командная часть

### Программирование

#### *Задача П.2.1.1. Морской бой (20 баллов)*

Автоматическая проверка данных в настоящее время – явление нередкое. Программисту, создающему автоматический тест необходимо понять алгоритм работы тестируемой программы, понять как проверить полученный результат и сделать вывод о правильности введенных данных. Данная задача наглядно показывает принцип написания тестов.

Учащихся начальных классов обучают игре в морской бой с помощью онлайн-курса. Они играют друг с другом в режиме онлайн. Мальчик Вася почти победил — он полностью уничтожил все корабли Пети, кроме одного. Но Петя схитрил — последний корабль он просто не разместил на поле. Может ли Вася понять на текущем ходу, жульничает его соперник или нет?

Напоминание о правилах игры «Морской бой»:

1. У каждого игрока есть 10 кораблей, представляющих из себя прямую линию толщиной в 1 клетку и различной длины:
  - 4 корабля длиной 1.
  - 3 корабля длиной 2.
  - 2 корабля длиной 3.
  - 1 корабль длиной 4.
2. Корабли могут располагаться в любом месте поля, но не должны соприкасаться с другими кораблями ни бортами, ни углами.

#### *Формат входных данных*

На вход программе подается 10 строк, каждая из которых состоит из 10 символов. (клетка, по которой Вася еще не стрелял), «x» (клетка, в которую Вася стрелял, но в корабль не попал) или «#» (клетка с пораженной частью корабля). Эта таблица демонстрирует поле боя, которое видит Вася. Гарантируется, что все пораженные корабли уничтожены полностью.

#### *Формат выходных данных*

На выход программа должна выдать yes, если имеющейся у Васи информации достаточно, чтобы понять на текущем ходу, что его соперник жульничает, и no в противном случае.

## Примеры

### Пример №1

Стандартный ввод
<pre> .....x..#. .#.####.#. .#.....x .#.#.##. x...x... ..x#.#.##. ..... ..x..... .x.x..... ..... </pre>
Стандартный вывод
no

### Пример №2

Стандартный ввод
<pre> #.x..##.x ...xx....# x.xx.xxx.# #.x...x..x ...x...x.. ..x.xx...# #...x.x..# ..x...x... .x.....x. #.###x.### </pre>
Стандартный вывод
yes

## Решение

```

1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <iostream>
4  #include <vector>
5  #include <algorithm>
6  #include <string>
7  #include <utility>
8
9  #include <stdio.h>
10
11 int main(int argc, const char **argv) {
12     std::vector<std::string> field(10);
13     for (int i = 0; i < 10; ++i) {
14         std::cin >> field[i];
15     }
16
17     int lost_ship_size = 0;

```



```

18     std::vector<std::vector<int> > map(10, std::vector<int>(10, 255));
19     for (int y = 0; y < 10; ++y) {
20         for (int x = 0; x < 10; ++x) {
21             if (field[y][x] == 'x') {
22                 map[y][x] = 0;
23             } else if (field[y][x] == '#') {
24                 ++lost_ship_size;
25                 int l = std::max(0, x - 1);
26                 int r = std::min(9, x + 1);
27                 int t = std::max(y - 1, 0);
28                 int b = std::min(y + 1, 9);
29                 for (int j = t; j <= b; ++j) {
30                     for (int i = l; i <= r; ++i) {
31                         map[j][i] = 0;
32                     }
33                 }
34             }
35         }
36     }
37     lost_ship_size = ((1 * 4) + (2 * 3) + (3 * 2) + (4 * 1)) - lost_ship_size;
38
39     for (int is_hor = 0; is_hor < 2; ++is_hor) {
40         for (int y = 0; y < 10; ++y) {
41             for (int x = 0; x < 10; ++x) {
42                 if (map[y][x] > 0) {
43                     const int prev_val = is_hor ?
44                         (x > 0 ? map[y][x - 1] : 0) :
45                         (y > 0 ? map[y - 1][x] : 0);
46                     map[y][x] = prev_val + 1;
47                     if (map[y][x] == lost_ship_size) {
48                         std::cout << "no";
49                         return 0;
50                     }
51                 }
52             }
53         }
54     }
55     std::cout << "yes";
56     return 0;
57 }

```

### Задача II.2.1.2. Кварталы и трубы (20 баллов)

Во многих программах, связанных с финансами, требуется написание алгоритма, который будет находить наибольшую выгоду. Однако полным перебором такую задачу за короткое время не решить — слишком много вариантов. Тогда приходится придумывать решения, которые позволят сократить время обработки данных. Задача ниже — как раз из таких.

Город Нижний Квадратовск представляет собой прямоугольник, разделенный на  $M \times N$  квадратных кварталов. Коммунальные службы хотят провести из углового северо-западного квартала в угловой юго-восточный новую водопроводную трубу. Они начинают с северо-западного квартала и могут прокладывать ее только под кварталами и двигаться только на юг или на восток. Стоимость прокладки трубы под кварталом может различаться, поэтому коммунальным службам нужен совет — какая будет минимально возможная стоимость прокладки нового трубопровода?

### Формат входных данных

В первой строке находятся два числа —  $M$  и  $N$  ( $1 \leq M \leq 100$ ,  $1 \leq N \leq 100$ ), которые обозначают количество кварталов в городе по горизонтали и вертикали.

Далее следует  $N$  строк по  $M$  чисел в каждом, разделенных пробелами — стоимость проведения трубопровода под соответствующим кварталом. Каждая цена является целым числом и находится в диапазоне от 1 до 2000000000 рублей включительно.

Стоит отметить, что угловой северо-западный квартал — это квартал находится в верхнем левом углу таблицы, а юго-восточный — в нижнем правом. Стоимость трубопровода в этих кварталах также учитывается.

### Примеры

#### Пример №1

Стандартный ввод
4 3
1 6 1 2
1 1 3 8
9 2 5 4
Стандартный вывод
14

#### Пример №2

Стандартный ввод
6 8
1 6 1 2 8 9
1 1 3 8 5 6
9 2 5 4 7 5
8 5 4 5 6 9
7 5 1 6 5 2
8 5 6 9 5 4
1 1 3 8 5 6
9 2 5 4 7 5
Стандартный вывод
43

### Решение

```

1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <iostream>
4  #include <vector>
5  #include <algorithm>
6
7  #include <stdio.h>
8
9  int main(int argc, const char **argv) {

```

```

10     int m = 0;
11     int n = 0;
12     std::cin >> m >> n;
13
14     std::vector<std::vector<long long> > dp(n, std::vector<long long>(m));
15
16     for (int y = 0; y < n; ++y) {
17         for (int x = 0; x < m; ++x) {
18             std::cin >> dp[y][x];
19             if (y > 0 && x > 0) {
20                 dp[y][x] += std::min(dp[y - 1][x], dp[y][x - 1]);
21             } else {
22                 if (y > 0) {
23                     dp[y][x] += dp[y - 1][x];
24                 } else if (x > 0) {
25                     dp[y][x] += dp[y][x - 1];
26                 }
27             }
28         }
29     }
30
31     std::cout << dp[n - 1][m - 1];
32
33     return 0;
34 }

```

### *Задача II.2.1.3. Потерявшиеся автомобили (30 баллов)*

Сейчас все популярнее становятся темы беспилотных транспортных средств и геолокации. Ниже представлена упрощенная версия одной из систем по нахождению собственного положения при наличии заранее известной карты местности. Эта задача потребует слаженной командной работы на всех этапах — от продумывания решения до реализации и тестирования.

В городе испытывают новые автомобили с искусственным интеллектом, которые умеют самостоятельно передвигаться по городу. Как это часто случается с новыми технологиями, автомобили иногда теряется. Тогда они останавливаются, отправляют в центр информацию о текущем окружении и ждут информацию о возможном местоположении.

В центре имеется карта города, поэтому каждый раз, когда автомобиль теряется, его местоположение сверяется с картой, а также проверяется траектория с последними перемещениями автомобиля. Если текущее положение совпадает с каким-то местом на карте, а последние передвижения возможны, то координаты отправляются автомобилю как возможные. Стоит отметить, что возможных мест в городе может быть несколько, и автомобиль получает их все.

Ваша задача — написать систему, которая по карте города сможет отвечать на запросы от потерявшихся автомобилей и сообщать им возможные места.

#### *Формат входных данных*

В первой строке находится два числа  $W$  и  $H$  — высота и ширина карты города соответственно. ( $3 \leq W \leq 500$ ) ( $3 \leq H \leq 500$ ).

В следующих  $H$  строках находятся  $W$  символов, где «.» обозначает дорогу, а «#»

— здание, через которое автомобиль проехать не мог. Координаты города начинаются в левом верхнем углу, который имеет координаты  $(0,0)$ . Ось  $Y$  направлена вниз,  $X$  — вправо. Гарантируется, что первый и последний символ каждой строки — «#», а также первая и последняя строки полностью состоят из символа «#».

Затем в отдельной строке идет число  $K$  — количество запросов ( $1 < K < 500$ ).

Далее следует  $K$  запросов, имеющих следующую структуру:

1. строка из символов (длина строки не меньше 1 символа и не больше 200), задающих последний путь пройденный автомобилем до момента потери (символами могут быть «n»1 (вверх), «s»1 (вниз), «e»1 (вправо) и «w»1 (влево)).
2. 3 строки по 3 символа в каждой, показывающих окружение автомобиля на момент потери (то есть на момент времени после прохождения последней команды в пути). Автомобиль не указывается и находится в центре представленной матрицы.

### *Формат выходных данных*

$K$  строк, в каждой из которых — список возможных координат автомобиля на момент потери для соответствующего запроса, каждая пара координат  $x, y$  — в круглых скобках, через запятую, без пробелов. Запросы отделяются друг от друга запятой и пробелом. Координаты должны следовать в порядке возрастания сначала координаты  $y$ , затем  $x$  (проще говоря, координаты обходятся сверху вниз и слева направо). Для лучшего понимания следует рассмотреть представленные примеры.

### *Примеры*

#### *Пример №1*

Стандартный ввод
5 6
#####
#.##
#.##
#...#
#...#
#####
2
n
###
##
##
swwn
##
##
#..
Стандартный вывод
(1,1), (3,1)
(1,2)

*Решение*

```

1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <iostream>
4  #include <string>
5  #include <list>
6  #include <vector>
7  #include <utility>
8
9  #include <stdio.h>
10
11 inline int get_mask(const char* top_line,
12                   const char* mid_line,
13                   const char* bot_line) {
14     return ((top_line[0] == '#') << 7) |
15            ((top_line[1] == '#') << 6) |
16            ((top_line[2] == '#') << 5) |
17            ((mid_line[0] == '#') << 4) |
18            ((mid_line[2] == '#') << 3) |
19            ((bot_line[0] == '#') << 2) |
20            ((bot_line[1] == '#') << 1) |
21            ((bot_line[2] == '#') << 0);
22 }
23
24 int main(int argc, const char **argv) {
25     int w = 0;
26     int h = 0;
27     std::cin >> w >> h;
28     std::vector<std::string> map(h);
29     for (int i = 0; i < h; ++i) {
30         std::cin >> map[i];
31     }
32
33     std::vector<std::list<std::pair<int, int> > > areas(256);
34     for (int y = 1; y < h - 1; ++y) {
35         for (int x = 1; x < w - 1; ++x) {
36             areas[get_mask(map[y - 1].data() + x - 1,
37                           map[y + 0].data() + x - 1,
38                           map[y + 1].data() + x - 1)].push_back({ x, y });
39         }
40     }
41
42     int k = 0;
43     std::cin >> k;
44
45     for (int i = 0; i < k; ++i) {
46         std::string path;
47         std::cin >> path;
48
49         std::string top_line;
50         std::string mid_line;
51         std::string bot_line;
52         std::cin >> top_line >> mid_line >> bot_line;
53         const int mask = get_mask(top_line.data(), mid_line.data(), bot_line.data());
54
55         std::list<std::pair<int, int> > answers;
56         for (const auto& coord : areas[mask]) {
57             int x = coord.first;
58             int y = coord.second;

```

```
59     bool is_possible = true;
60
61     // move backwards
62     for (int j = path.size() - 1; j >= 0; --j) {
63         switch (path[j]) {
64             case 'n': y += 1; break;
65             case 's': y -= 1; break;
66             case 'e': x -= 1; break;
67             case 'w': x += 1; break;
68             default: break;
69         }
70         if (map[y][x] == '#') {
71             is_possible = false;
72             break;
73         }
74     }
75     if (is_possible) {
76         answers.push_back(coord);
77     }
78 }
79
80 const int answers_count = answers.size();
81 int cur_answer = 0;
82 for (const auto& coords : answers) {
83     ++cur_answer;
84     std::cout << "(" << coords.first << "," << coords.second << ")";
85     std::cout << std::string((answers_count != cur_answer) ? ", " : "\n");
86 }
87 }
88
89 return 0;
90 }
```