

# ЭЛЕКТРОННАЯ ИНЖЕНЕРИЯ: УМНЫЙ ДОМ

Профиль «Электронная инженерия: Умный дом» посвящен решению задачи проектирования дома будущего. Участникам данного профиля предстоит оборудовать умный системами охранно-пожарной сигнализации на основе датчиков присутствия, дыма, протечек, а также обеспечить его контролем доступа, видеонаблюдением, интеллектуальным освещением, рассчитать тепло- и энергопотребление

Профиль включает в себя задачи по двум школьным предметам: **физика** и **информатика**.

## §1 Первый отборочный этап

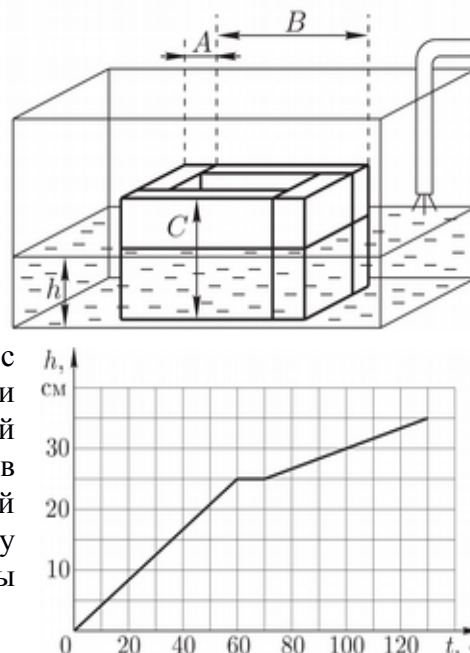
Первый отборочный тур проводится индивидуально в сети интернет, работы оцениваются автоматически средствами системы онлайн-тестирования. Для каждой из параллелей (9 класс или 10-11 класс) предлагается свой набор задач по физике, информатика общая для всех классов. На решение задач учащимся отводилось 2 суток. Первый этап состоял из двух независимых попыток с разными задачами, участник мог решать только одну попытку (в случае, если он решал обе, в зачет шла лучшая). Решение каждой задачи дает определенное количество баллов. Баллы зачисляются в полном объеме за правильное решение задачи. Участники получают оценку за решение задач в совокупности по всем предметам данного профиля (**математика** и **физика**).

### 1.1 Первая попытка Задачи по физике (8-9 класс)

#### Задача 1.1.1

*Условие:*

Семён соорудил колодец из 4 блоков массой 3,24 кг каждый. Колодец он поместил на дно стеклянного сосуда прямоугольной формы. Площадь основания сосуда  $540 \text{ см}^2$ . Блоки между собой к дну сосуда приклеены водоустойчивым клеем. Эта система нужна Семёну для автоматизации подачи воды в бассейн, для чего ему понадобилось провести предварительный эксперимент по снятию зависимости уровня воды в бассейне от времени. Семён подаёт воду в сосуд с помощью шланга, опущенного между стенкой сосуда и колодцем. Вода подаётся шлангом с постоянной скоростью. Измеренная зависимость уровня воды в сосуде от времени приведена на рисунке. В начальный момент времени сосуд пуст. Определите по этому графику плотность материала, из которого сделаны блоки колодца. Массой клея пренебечь.



*Решение:*

На графике, от нулевой до 60-й секунды – участок, соответствующий заполнению пространства между стенками сосуда и колодцем. Объем наливаемой воды определяется произведением высоты  $h$  на разность между площадью дна сосуда  $S_0$  и площадью внешнего

сечения колодца  $S_l=(B+A)^2$ . В течение последующих 10 секунд уровень воды в сосуде не меняется, что обусловлено заполнением водой внутренней полости колодца через верхний край блоков. Площадь сечения внутренней полости колодца  $S_l=(B-A)^2$ . Начиная с 70-й секунды уровень воды превышает высоту стен колодца. Из графика следует, что длина блока  $C=25$  см. С этого момента заполнение сосуда происходит медленнее, чем на первом участке, так как площадь сечения заполняемого объёма стала больше. Участок графика от 70-й до 130-й секунды позволяет определить расход поступающей воды в сосуд в литрах в секунду:

$$v = \frac{\Delta h S_0}{\Delta t_3} = \frac{10 \text{ см} \times 540 \text{ см}^2}{60 \text{ с}} = 90 \frac{\text{см}^3}{\text{с}} = 0,09 \frac{\text{л}^3}{\text{с}}$$

Теперь на основе первой части графика (до 60-й секунды) можно определить внешнюю площадь сечения колодца. Поскольку за 60 секунд уровень воды достиг значения 25 см, то объём поступившей за это время воды с одной стороны равен произведению этой высоты на разность площадей  $S_0$  и  $S_l$ , а с другой стороны этот объём равен произведению расхода воды на время её поступления:

$$h_0(S_0 - (A+B)^2) = v \Delta t_1$$

$$(B+A)^2 = 324 \text{ см}^2 \quad (1)$$

Заполнение внутренней полости колодца продолжалось 10 секунд, следовательно

$$h_0(B-A)^2 = v \Delta t_2$$

$$(B-A)^2 = \frac{v \Delta t_2}{h_0} = 36 \text{ см}^2 \quad (2)$$

Из (1) и (2) найдем  $B=12$  см и  $A=6$  см. Тогда объём блока  $V = ABC = 6 * 12 * 25 = 1800 \text{ см}^3 = 0,0018 \text{ м}^3$

Плотность

$$\rho = M/V = \frac{3,24}{0,0018} = 1800 \text{ кг/м}^3$$

### Задача 1.1.2

Условие:

Семён и Настя играют в машинки. Массы машинок 200 и 400 грамм и они связаны нитью, выдерживающей силу натяжения  $10 \text{ Н}$ . Настя и Семён, тянут горизонтально в разные стороны с переменными силами  $F_1=t$  и  $F_1=2t$ . В какой момент времени нить оборвется? Трения нет.

Решение:

Запишем закон Ньютона для обоих тел

$$\begin{aligned} -F_1 + T &= m_1 a; & -t + T &= m_1 a; \\ -T + F_2 &= m_2 a; & -T + 2t &= m_2 a; \end{aligned}$$

Выразим ускорения и приравняем

$$\frac{T-t}{m_1} = \frac{-T+2t}{m_2}; \quad T \frac{m_1+m_2}{m_1 m_2} = t \frac{2m_1+m_2}{m_1 m_2}; \quad t = T \frac{m_1+m_2}{2m_1+m_2} = 10 \frac{200+400}{400+400} = 10 \frac{6}{8} = 7,5 \text{ с}$$

### Задача 1.1.3

Условие:

Учитель физкультуры в Школе Семёна бежит навстречу колонне школьников со скоростью  $u=7$  км/ч. Школьники бегут в ряд по одному со скоростью  $v=14$  км/ч. Длина колонны детей равна  $l=60$  м. Когда учитель пробегает мимо школьника – школьник разворачивается и бежит вдоль колонны в обратную сторону с той же по модулю скоростью. Чему будет равна длина колонны, когда все школьники развернутся?

Решение:

Относительная скорость  $v+u$ . Время относительного движения  $t = \frac{l}{v+u}$ .  
За это время первый ребёнок пройдёт относительно учителя расстояние  $(v-u)t$ . Тогда длина колонны:

$$l' = (v-u)t = l \frac{v-u}{v+u} = 60 \frac{14-7}{14+7} = 60 \frac{7}{21} = 20 \text{ м.}$$

### Задача 1.1.4

Условие:

Настя и Семён собирают предохранитель для своей радиостанции. У них есть проволока из свинца двух диаметров. В справочнике они нашли, что уже для тока 1,8 А плавится свинцовая проволока диаметром  $d = 0,30$  мм, а проволока диаметром  $D = 0,60$  мм плавится при токе  $I = 5,0$  А. При каком токе разорвёт цепь предохранитель, составленный из двух свинцовых проволочек указанных диаметров, соединённых параллельно? Длины проволочек считать одинаковыми.

Решение:

Для правильного решения задачи необходимо учитывать распределение токов между проволочками – в какой именно из них раньше будет достигнуто предельное значение тока.

$$\frac{I_1}{I_2} = \frac{R_2}{R_1} = \frac{S_1}{S_2} = \frac{d^2}{D^2}. \quad (1)$$

Из (1) следует, что при токе через  $R_2$  равном 5 А (предельном), ток через  $R_1$  равен 1,25 А. Поэтому в первом варианте сборки предохранителя первой расплавится толстая проволока ( $D = 0,6$  мм). В этот момент ток в цепи будет

$$I = 5 \text{ А} + 1,25 \text{ А} = 6,25 \text{ А}$$

– иными словами, после разрыва контакта в цепи  $R_2$  весь этот ток немедленно «сожжёт» и тонкую проволоку, т.е. предохранитель выполнит свою функцию и полностью разомкнёт цепь.

Ответ: 6,25 А.

## 1.2 Первая попытка

### Задачи по физике (10-11 класс)

#### Задача 1.2.1

Условие:

Рядом с домом Семёна течёт река. Ширина реки  $H=50\text{м}$ , а скорость течения  $v_1=3\text{км/ч}$ . Семён много раз за лето плавал через реку и засекал время. Таким образом он определил при каких условиях время минимально. Определите эти условия. Где окажется на другом берегу Семён в этом случае? Скорость Семёна относительно воды  $v_2=2\text{км/ч}$ .

Решение:

Если Семён плывёт под углом  $\alpha$  к берегу, тогда  $H=v_2\sin\alpha t$ , а смещение вдоль берега будет равно  $l=(v_1+v_2\cos\alpha)t$ . Время на пересечение реки  $t=H/(v_2\sin\alpha)$  минимально, при условии максимальности  $\sin\alpha$ , то есть при  $\alpha=\pi/2$ , тогда  $t_{\min}=H/v_2$ . Смещение вдоль берега будет равно

$$l=v_1 t=H \frac{v_1}{v_2}=\frac{50 \cdot 3}{2}=75 \text{ м.}$$

Путь Семёна

$$S=\sqrt{H^2+l^2}=\frac{H}{v_2}\sqrt{v_1^2+v_2^2}\approx 90 \text{ м.}$$

Ответ:

Условие минимальности времени – плыть нужно перпендикулярно берегу, или угол между скоростью Семёна относительно воды и берегом должен быть  $\alpha=\pi/2$ . На том берегу Семён окажется в 75 м ниже по течению.

#### Задача 1.2.2

Условие:

Дома у Семёна есть домашняя киностудия. Он решил снять небольшой фильм с погоней. В некоторый момент с моста автомобиль падает в реку. Масштаб макета моста и машины 1 : 40. Семёну нужно решить вопрос о количестве кадров эпизода с падением машины – с какой частотой нужно снимать, чтобы при просмотре со стандартной частотой 24 кадра в секунду картина выглядела правдоподобно?

Решение:

Если реальный автомобиль падает с высоты  $h$ , то игрушечный с высоты  $h/40$ . Обозначим времена падения реального автомобиля и игрушечного через  $t_p$  и  $t_u$ . Падение обоих происходит с одним и тем же ускорением – ускорением свободного падения. Тогда

$$\frac{t_p}{t_u}=\sqrt{\frac{h}{h/40}}.$$

Чтобы ситуация выглядела правдоподобно, за время падения оригинала и макета должно быть отснято одинаковое количество кадров. Отсюда:

$$N_p t_p = N_{и} t_{и}.$$

Тогда, используя отношение времён, получим:

$$N_p t_{и} \sqrt{40} = N_{и} t_{и},$$

$$N_{и} = N_p \sqrt{40} = 24 \sqrt{40} = 152$$

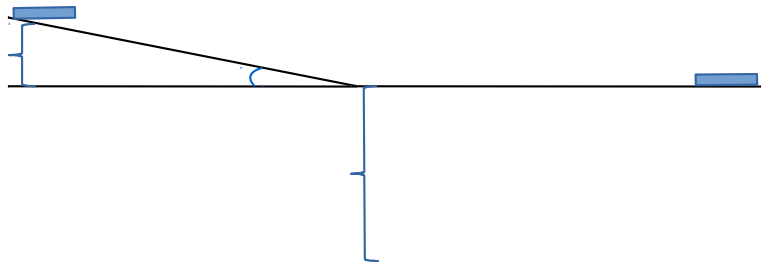
*Ответ:*

152 кадра в секунду (с округлением до целых).

### Задача 1.2.3

*Условие:*

Зимой во дворе Семён с папой сделали горку высотой  $H=2$  м и углом наклона  $\alpha=30^\circ$  к горизонту, переход с наклонной плоскости в горизонтальную получился резким, что отразилось на дальности скатывания. С нулевой начальной скоростью Семён скатывается с горки. Найдите путь по горизонтальному участку после горки до полной остановки если коэффициент трения везде одинаков и равен  $\mu=0,1$ .



*Решение:*

Второй закон Ньютона для санок в проекциях вдоль горки и перпендикулярно ей:

$$N = mg \cos \alpha,$$

$$-F_{тр} + mg \sin \alpha = ma.$$

Скорость у основания горки можно найти из закона изменения полной механической энергии:

$$\frac{mv_1^2}{2} = mgH - A_{тр}.$$

Работа, по определению, равна  $A_{тр} = l F_{тр} = l mg \cos \alpha = \frac{H}{\sin \alpha} mg \cos \alpha$ . Тогда квадрат скорости:  $v_1^2 = 2gH(1 - \mu \operatorname{ctg} \alpha)$ .

В основании горки санки испытывают удар и скорость изменяет своё направление на горизонтальное – вертикальный импульс гасится землёй, таким образом, начальной скоростью по горизонтальному участку пути будет не  $v_1$ , а  $v_1 \cos \alpha$ . Применим для горизонтального участка теорему о кинетической энергии:

$$0 - \frac{mv_1^2 \cos^2 \alpha}{2} = A_{тр} = -S F_{тр} = -S \mu mg.$$

Силу трения снова нашли из определения  $F_{тр} = \mu N'$ . При движении по горизонтальному участку сила реакции опоры равна  $N' = mg$ . Выразим искомый путь:

$$S = \frac{v_1^2 \cos^2 \alpha}{2\mu g} = \frac{H}{\mu} \cos^2 \alpha (1 - \mu \operatorname{ctg} \alpha).$$

$$S = \frac{2}{0,1} \cos^2 30^\circ (1 - 0,1 \operatorname{ctg} 30^\circ) = 12,4 \text{ м.}$$

Если не учитывать удар (что не есть правильно!) в основании горки, то путь получается больше (разница в  $\cos^2 \alpha$ ) – 16,5 м.

*Ответ:*

С округлением до десятых 12,4 м.

#### **Задача 1.2.4**

*Условие:*

На кухне у Семёна поддерживается постоянная температура. Семён программирует автомат, управляющий работой холодильника, для начала проделав эксперимент. Он начал с пустого холодильника без продуктов. Если температура в морозильной камере поднимается до минус одного градуса Цельсия, то включается компрессор и работает 15 минут. За это время температура падает до минус двух градусов по Цельсию. Тогда программа выключает компрессор. Через 25 минут всё повторяется когда температура в камере вновь поднимается до минус одного градуса Цельсия. После проведения измерений, Семён отключил холодильник от сети, записал время отключения. Найдите время, за которое растаял весь лёд со стенок морозильной камеры, если его там было четыре килограмма. Теплоёмкость льда 2,1 кДж/(кг К), а его удельная теплота плавления 330 кДж/кг. Теплоёмкостью камеры пренебречь.

*Решение:*

Поток тепла в камеру зависит от разности температур снаружи и внутри камеры. При таянии льда поток тепла в камеру практически такой же, как и в рабочем режиме. Мощность потока тепла можно определить по нагреванию камер при отключенном компрессоре:

$$P = mc_n \frac{\Delta t}{\tau}$$

Здесь  $\Delta t = 1\text{C}^\circ$  – разность температур,  $\tau = 25$  минут – время до включения компрессора. Время до полной разморозки можно найти из условия:

$$PT = m\lambda, T = \frac{\lambda}{c_n} \frac{\tau}{\Delta t} = \frac{330000}{2100} \frac{25}{60} = 65,5 \text{ ч} = 2,7 \text{ сут}$$

*Ответ:*

В часах с округлением до десятых – 65,5 часа.

#### **Задача 1.2.5**

*Условие:*

Общее сопротивление двух последовательно соединённых проводников 5 Ом, а параллельно соединённых – 1,2 Ом. Определите сопротивление каждого проводника.

Решение:

$$\frac{1}{x} + \frac{1}{y} = \frac{1}{1.2}, x + y = 5$$

Решая систему уравнений найдем сопротивления проводников – 2 и 3 Ома (порядок цифр не важен, одно из сопротивлений 2 Ом, другое 3 Ом).

### 1.3 Первая попытка Задачи по информатике

#### Задача 1.3.1 (1 балл)

Условие:

Для управления умным домом с центрального сервера постоянно приходится отправлять исполнительным устройствам числовые команды. Например, осветительный прибор может поддерживать 21 градацию яркости (0..20). Если центральный сервер посылает команду 0, лампочка отключается и не горит, если посылается команда 20, лампочка горит с максимальной яркостью, при команде 10 используется только половина яркости.

Пульт управления осветительным прибором может содержать количество возможных команд, отличное от доступной градации яркости самого светильника. Например, пусть на пульте управления имеется только 6 возможных состояний (0..5). При управлении осветительным прибором с такого пульта необходимо обеспечить равномерное распределение изменения яркости (с одинаковым шагом) по всему интервалу доступных значений пульта таким образом, чтобы при нулевом значении на пульте управления на лампочку подавалась команда 0, при максимальном значении на пульте управления, подавалась максимальная доступная для светильника величина.

Таблица соответствия значений пульта и осветительного прибора для приведенного примера представлена в таблице:

Значение на пульте	Значение на осветительном приборе
0	0
1	4
2	8
3	12
4	16
5	20

В данном примере градации значений на пульте и на осветительном приборе оказались хорошо согласующимися друг с другом, т.е. удалось подобрать целочисленный шаг, для равномерного распределения освещения в зависимости от значения на пульте. В случае, когда шаг получается не целочисленным, значение, которое следует передать осветительному прибору, должно округляться вниз до ближайшего целого.

#### Формат входных данных

В единственной строке дано два целых числа  $L$  – количество возможных значений, принимаемых осветительным прибором, и  $C$  – количество доступных значений пульта управления ( $2 \leq L \leq 100, 2 \leq C \leq L$ ).

### Формат выходных данных

В единственной строке через пробел необходимо вывести  $C$  целых чисел – значения, передаваемые на осветительный прибор, соответствующие значению на пульте управления, по порядку, начиная с 0.

### Пример

Входные данные	Выходные данные
21 6	0 4 8 12 16 20

Входные данные	Выходные данные
21 4	0 6 13 20

*Решение:*

По условию первое значение всегда равно 0, а последнее значение равно  $L - 1$ .

Все промежуточные числа  $V_i$  можно рассчитать по формуле  $V_i = \frac{i \cdot (L - 1)}{C - 1}$ , где  $i = 0..(C - 1)$ ,  $V$  – последовательность, являющаяся ответом на задачу.

### Код решения (C++)

```
#include <iostream>
#include <cstdio>

using namespace std;

int main()
{
    int l, c;
    cin >> l >> c;
    --l;

    for (int i = 0; i < c; i++)
        cout << i * l / (c - 1) << ' ';

    return 0;
}
```

### Задача 1.3.2 (3 балла)

*Условие:*

При управлении умным домом возникают ситуации, когда необходимо запускать и прекращать некоторые события в правильном порядке. Например, чтобы воспроизвести песню, необходимо сначала запустить медиа сервер, и только потом включить выбранный звуковой файл.

По данной записи событий и установленному правилу порядка включения и выключения устройств вам необходимо определить, корректно ли запускались и отключались



все имеющиеся устройства. Под корректностью подразумевается правильный порядок запуска/отключения, соответствующий заданным правилам и отсутствие попыток повторного запуска или повторного отключения какого-либо устройства. Под повторным запуском (отключением) устройства подразумевается попытка запустить (отключить) уже запущенное (отключенное) устройство.

При решении задачи полагать, что все устройства изначально отключены, но по окончании выполнения всех команд некоторые из них могут оставаться включенными.

Примечание: Если поступает команда отключить устройство  $d_2$ , которое должно отключаться после устройства  $d_1$ , при этом устройство  $d_1$  изначально находится в отключенном состоянии и не было включено ни разу, такая команда является корректной (если  $d_2$  не отключается повторно).

### Формат входных данных

В первой строке даны три целых числа  $N$  – количество задействованных устройств,  $R$  – количество правил порядка запуска и отключения устройств, и  $M$  – количество произведенных команд включения и выключения ( $2 \leq N \leq 1000, 1 \leq R \leq N \cdot (N - 1), 1 \leq M \leq 10^5$ ).

Следующие  $R$  строк содержат три целых числа  $t$  – режим запуска ( $t=1$ ) или отключения ( $t=0$ ),  $d_1$  и  $d_2$  – номера устройств в том порядке, в котором они должны включаться или отключаться в зависимости от режима, заданного числом  $t$ .

В следующих  $M$  строках содержатся команды в виде двух целых чисел:  $t_c$  – режим включения/отключения (аналогично  $t$ ), и  $d$  – номер устройства, над которым производится действие  $t_c$ .

Устройства нумеруются целыми числами от 1 до  $N$ .

### Формат выходных данных

В единственной строке выведите «OK» в случае если последовательность команд корректна, и «WRONG» в противном случае. Кавычки выводить не нужно.

### Пример

Входные данные	Выходные данные
3 3 5 1 1 2 0 2 1 0 3 1 1 3 1 1 0 3 1 2 1 3	OK

Входные данные	Выходные данные
3 3 7 1 1 2 0 2 1 0 3 1	WRONG

1 3	
1 1	
0 3	
1 2	
1 3	
0 2	
0 1	

Входные данные	Выходные данные
3 3 6 1 1 2 0 2 1 0 3 1 1 3 1 1 0 3 1 2 0 2 0 1	OK

*Решение:*

Для решения данной задачи для каждого запроса на корректность выполнения операции необходимо выполнять проверку состояния всех устройств, которые должны быть включены или выключены, в зависимости от того, требуется включить или отключить определенное устройство в текущей операции. В случае, если хотя бы одно из устройств не удовлетворяет заданным условиям, необходимо вывести «WRONG», дальнейшие проверки можно не выполнять. Если все условия соблюдены, необходимо зафиксировать новое состояние текущего устройства и продолжить обрабатывать запросы. Если все запросы обработаны, вывести «OK».

**Код решения (C++)**

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <cstdio>
#include <vector>

using namespace std;

bool check(vector<vector<bool>> &req, vector<bool> &status, int
d, int req_val)
{
    if (status[d] == req_val)
    {
        return false;
    }
}
```

```

    for (int i = 0; i < req.size(); i++)
    {
        if (req[i][d] && status[i] != req_val)
        {
            return false;
        }
    }

    return true;
}

int main()
{
    int n, r, m;

    scanf("%d%d%d\n", &n, &r, &m);

    vector<vector<bool>> on(n, vector<bool>(n, false));
    vector<vector<bool>> off(n, vector<bool>(n, false));

    for (int i = 0; i < r; i++)
    {
        int mode, d1, d2;
        scanf("%d%d%d\n", &mode, &d1, &d2);
        d1--;
        d2--;
        if (mode)
        {
            on[d1][d2] = 1;
        }
        else
        {
            off[d1][d2] = 1;
        }
    }

    vector<bool> status(n, false);
    bool flag = true;
    for (int i = 0; i < m; i++)
    {
        int mode, d;
        scanf("%d%d\n", &mode, &d);
        --d;
        if (!check(mode ? on : off, status, d, mode))
        {
            flag = false;
            break;
        }
        status[d] = mode;
    }

    printf(flag ? "OK" : "WRONG");

    return 0;
}

```

### Задача 1.3.3 (1 балл)

Условие:

Для сокращения передаваемой информации для управления многозадачными устройствами умного дома набор команд для включения и выключения сразу нескольких функций одного устройства кодируется в одно единственное целое число следующим образом.

Пусть устройство имеет  $K$  различных функций. Функции нумеруются по порядку от  $0$  до  $(K-1)$ . Тогда кодируемое число  $C$  равняется  $t_0 \cdot 2^0 + t_1 \cdot 2^1 + t_2 \cdot 2^2 + \dots + t_{K-1} \cdot 2^{K-1}$ , где  $t_i$  равняется  $0$ , если функция должна быть отключена, и  $1$  если функцию необходимо задействовать.

Для каждой заданной команды вам необходимо определить, сколько функций, в результате её выполнения, изменят своё состояние (задействована/отключена).

При решении задачи полагать, что все функции изначально отключены, но по окончании выполнения всех команд некоторые из них могут оставаться задействованными.

#### Формат входных данных

В первой строке даны два целых числа  $N$  – количество поступающих команд, и  $K$  – количество различных функций устройства ( $1 \leq N \leq 10^3, 1 \leq K \leq 30$ ).

Следующие  $N$  строк содержат по одному целому числу  $C$  – закодированная команда ( $0 \leq C < 2^{31}$ ).

#### Формат выходных данных

Для каждой команды в отдельной строке выведите сколько функций в результате её выполнения изменят своё состояние с задействована на отключена или наоборот с отключена на задействована.

#### Пример

Входные данные	Выходные данные
5 3	1
2	3
5	1
7	2
1	3
6	

Решение:

Для простоты решения необходимо уметь работать с бинарными операциями используемого языка программирования. Заметим, что результат  $D$  побитовой операции исключающего ИЛИ чисел  $S_1$  и  $S_2$  в двоичном представлении будет содержать единицы только в тех разрядах, которые соответствуют функциям, состояние которых было изменено при переходе от состояния  $S_1$  к состоянию  $S_2$ . Таким образом, ответом на задачу будет количество единиц в числе  $D$ , полученном для каждой пары последовательных состояний.

## Код решения (C++)

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <cstdio>

using namespace std;

int calc(int val, int new_val)
{
    int res = 0;

    val ^= new_val;

    while (val)
    {
        res += (val & 1);
        val >>= 1;
    }

    return res;
}

int main()
{
    int n, k;
    int val = 0;
    scanf("%d%d\n", &n, &k);

    for (int i = 0; i < n; i++)
    {
        int new_val;
        scanf("%d\n", &new_val);
        printf("%d\n", calc(val, new_val));
        val = new_val;
    }

    return 0;
}
```

### Задача 1.3.4 (2 балла)

*Условие:*

При проектировании системы управления умным домом необходимо обеспечивать связь различных устройств между собой. Например, температурные датчики должны передавать получаемые показатели устройству управления отоплением и кондиционирования. В свою очередь, это устройство, при необходимости, должно передать команду запуска или остановки соответствующей обстановке системе. На ранних стадиях разработки умного дома нет необходимости объединять все имеющиеся устройства в одну сеть, поэтому устройства объединяются только в том случае, если они действительно зависят друг от друга. Однако позже необходимость объединения в одну большую сеть может возникнуть. То есть если изначально медиа-система, объединяющая в себе колонки, медиа-сервер и дисплей никоим образом не были связаны с системой отопления, то в случае возникновения желания вывести текущую температуру на экран две эти системы придётся объединить в одну сеть.

Так как наш дом всё-таки «умный», каждое устройство может пропустить через себя сообщение для любого другого, при наличии необходимой связи. Таким образом, если в описанной системе проложить связь между кондиционером и колонками, учитывая, что кондиционер через устройство управления отоплением связан с температурным датчиком, а колонки через медиа-сервер связаны с дисплеем, то задача вывода на экран текущей температуры становится вполне решаемой.

В данной задаче вам не нужно рассуждать о рациональности такого подхода передачи данных. От вас требуется определить, каким наименьшим количеством связей необходимо дополнить существующую сеть, чтобы каждое устройство могло передать данные любому другому, и от какого наибольшего количества связей после этого можно избавиться, чтобы связь между всеми устройствами по-прежнему сохранялась.

### Формат входных данных

В первой строке даны два целых числа  $N$  – количество различных устройств, и  $K$  – количество имеющихся связей между ними ( $2 \leq N \leq 10^3, 1 \leq K \leq \frac{N \cdot (N - 1)}{2}$ ).

Следующие  $K$  строк содержат по 2 целых числа – номера устройств, между которыми имеется соединение. Устройства нумеруются по порядку от 1 до  $N$ .

### Формат выходных данных

В единственной строке выведите два числа:  $A$  – наименьшее количество связей, которые необходимо добавить, чтобы любые два устройства могли передать данные друг другу;  $D$  – наибольшее количество связей, которые можно удалить после добавления  $A$  связей, чтобы между любыми двумя устройствами сохранялся канал передачи данных.

### Пример

Входные данные	Выходные данные
7 5 1 2 1 3 2 3 4 5 5 7	2 1

### Решение:

Используя любой обход графа (в глубину или в ширину) посчитаем количество  $Q$  компонент связности. Количеством связей  $A$ , которые необходимо добавить в сеть, равно  $A = (Q - 1)$ ; объединяя каждый раз по две отдельные компоненты связности в одну, пока весь граф не станет связным, понадобится ровно такое количество рёбер.

Чтобы сеть была полностью единой, но не имела лишних связей, в ней должно быть ровно  $(N - 1)$  связей, то есть граф должен являться деревом. Тогда, количество связей  $D$ , которые необходимо удалить, равняется  $D = K + A - N + 1 = K + Q - N$ .

## Код решения (C++)

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <cstdio>
#include <vector>

using namespace std;

void dfs(vector<vector<bool>> &g, vector<int> &c, int v, int idx)
{
    c[v] = idx;

    for (int i = 0; i < g.size(); i++)
    {
        if (g[v][i] && !c[i])
        {
            dfs(g, c, i, idx);
        }
    }
}

int main()
{
    int n, k;
    scanf("%d%d\n", &n, &k);

    vector<vector<bool>> g(n, vector<bool>(n, false));
    vector<int> c(n, 0);

    for (int i = 0; i < k; i++)
    {
        int v1, v2;
        scanf("%d%d", &v1, &v2);
        --v1, --v2;
        g[v1][v2] = true;
        g[v2][v1] = true;
    }

    int comp_cnt = 0;
    for (int i = 0; i < n; i++)
    {
        if (!c[i])
            dfs(g, c, i, ++comp_cnt);
    }

    printf("%d %d", comp_cnt - 1, k + comp_cnt - n);

    return 0;
}
```

### Задача 1.3.5 (3 балла)

Условие:

В одной из ранних версий системы умного дома имелось  $D$  устройств, каждое из которых имело свой уникальный порядковый номер от 1 до  $D$ . Сервер, управляющий умным домом, опрашивал все устройства на необходимость выполнения какой-либо команды. Сначала он опрашивал устройство под номером 1, потом устройство под номером 2 и так далее до  $D$ . После этого он опрашивал их в обратном порядке:  $D-1, D-2, \dots, 1$ . Крайние устройства (под номерами 1 и  $D$ ) при смене направления обхода повторно не опрашивались. Если какому-либо из устройств было необходимо выполнить команду, оно ожидало пока до него дойдёт очередь опроса сервера. Сервер, опрашивая устройство и получая сообщение о необходимости выполнения какой-либо команды, ставил эту команду в очередь и переходил к опросу следующего устройства. Параллельно с этим опросом, если в очереди имелась какая-либо команда от устройства, эта команда начинала выполняться, но только в том случае, если выполнение предыдущей команд завершено. Каждая команда имеет своё время выполнения и пока она не выполнена, сервер не пытается запустить следующую команду из очереди. Сама очередь выполнения команд устроена по принципу «первый пришёл – первый обработан», т.е. команда, которая попала в очередь раньше, выполняется раньше.

Устройства начинают опрашиваться в момент времени  $T_0=0$ , начиная с устройства под номером 1. На опрос каждого устройства сервер тратит одну единицу времени. Если устройство передаёт сообщение о необходимости выполнения команды, оно добавляется в очередь в момент времени равный  $t_0+1$ , где  $t_0$  – момент начала выполнения опроса этого устройства. Возможны случаи, когда устройство, ждущее опроса чтобы отправить один запрос на выполнение, генерирует ещё один и более запросов. В таком случае все запросы одного устройства при опросе сервером так же добавляются в очередь в момент времени  $t_0+1$ , но выполняются в том порядке, в котором они возникали на самом устройстве.

Ваша задача по списку команд, которые запрашивают устройства в определенные моменты времени, определить время окончания их выполнения.

#### Формат входных данных

В первой строке даны два целых числа  $D$  – количество различных устройств, и  $K$  – количество команд, запрошенных устройствами ( $1 \leq D \leq 10^3, 1 \leq K \leq 10^5$ ).

В следующих  $K$  строках содержится по три числа:  $t$  – момент времени, в который устройству потребовалось выполнить команду,  $d$  – номер устройства, запрашивающего команду и  $l$  – продолжительность выполнения команды ( $1 \leq t \leq 10^5, 1 \leq d \leq D, 0 \leq l \leq 10^3$ ). Все команды отсортированы в порядке неубывания времени  $t$ .

#### Формат выходных данных

В отдельных строках выведите  $K$  чисел – моменты времени завершения выполнения команд в том порядке, в котором они даны во входных данных.

#### Пример

Входные данные	Выходные данные
5 4	7
0 3 4	10
2 2 2	11
2 1 1	20
12 1 3	



Входные данные	Выходные данные
6 6	7
3 4 3	30
4 2 20	31
4 1 1	34
6 1 3	62
53 1 1	65
55 1 3	

*Решение:*

Для решения данной задачи достаточно промоделировать процесс выполнения задач, поступающих в очередь. Однако моделировать каждый временной отсчёт нерационально. Можно рассматривать только временные метки, соответствующие каким-нибудь «значимым» событиям, например, время добавления команды в очередь. Для быстрого определения этого времени, на основании времени появления запроса на устройстве и *id* этого устройства, можно использовать формулы, представленные в функции `get_time_added` в прилагаемом коде решения.

**Код решения (C++)**

```
#define _CRT_SECURE_NO_WARNINGS
#pragma comment(linker, "/STACK:64000000")

#include <iostream>
#include <cstdio>
#include <string>
#include <algorithm>

using namespace std;

typedef struct Command {
    int id;
    int time;
    int dev;
    int time_cost;
    Command(int _id, int _time, int _dev, int _time_cost) :
        id(_id),
        time(_time),
        dev(_dev),
        time_cost(_time_cost)
    {}

    Command() :
        id(0),
        time(0),
        dev(0),
        time_cost(0)
    {}
};
```

```

    {}
} Command;

int get_time_added(int dev_time, int dev_id, int n)
{
    int m = n - 1;
    int t0 = dev_time / (m * 2) * m * 2;

    if (dev_id == 0)
    {
        return (t0 == dev_time ? t0 : t0 + 2 * m) + 1;
    }
    if (dev_id == n - 1)
    {
        return (t0 + m >= dev_time ? t0 + m : t0 + 3 * m) + 1;
    }

    int time_0 = t0 + dev_id;
    int time_1 = t0 + dev_id + (m - dev_id) * 2;

    return (dev_time <= time_0 ? time_0 : time_1) + 1;
}

int cmp(const Command &a, const Command &b)
{
    return a.time < b.time;
}

int cmp_id(const Command &a, const Command &b)
{
    return a.id < b.id;
}

int main()
{
    Command queue[100000];
    int head = 0;
    int tail = 0;

    int d, k;

    scanf("%d%d\n", &d, &k);

    for (int i = 0; i < k; i++)
    {
        scanf("%d%d%d\n", &queue[tail].time, &queue[tail].dev,
&queue[tail].time_cost);
        queue[tail].id = i;
        --queue[tail].dev;
        queue[tail].time = get_time_added(queue[tail].time,
queue[tail].dev, d);
    }
}

```

```

        tail++;
    }

    stable_sort(queue, queue + tail, cmp);

    int time_cur = 0;
    while (head < tail)
    {
        Command &command = queue[head++];
        if (time_cur < command.time)
        {
            time_cur = command.time + command.time_cost;
        }
        else
        {
            time_cur += command.time_cost;
        }
        command.time = time_cur;
    }

    sort(queue, queue + tail, cmp_id);

    for (int i = 0; i < tail; i++)
    {
        printf("%d\n", queue[i].time);
    }

    return 0;
}

```

## 1.4 Вторая попытка Задачи по физике (8-9 класс)

### Задача 1.4.1

*Условие:*

Семён первую треть всего времени движения шёл по грунтовой дороге со скоростью 3 км/ч. Следующую треть времени он перемещался по шоссе со скоростью 6 км/ч. Последний участок, длинной в треть всего пути, турист шёл со скоростью  $v_3$ . Найдите эту скорость. Определите, при какой скорости  $u$  он прошёл бы тот же путь за то же время, двигаясь равномерно.

*Решение:*

Поскольку третий участок составляет треть пути, и он был пройден за треть всего времени, то  $v_3 = u$ . Тогда

*Ответ:* обе скорости равны **4,5 км/ч**.

### Задача 1.4.2

*Условие:*

Семён решил сделать весы точнее, чем кухонные у него дома. Такие, чтобы точность была до десятых долей грамма. Он взял мерный стакан с водой и пробирку, проделал

следующий опыт. Пробирку с небольшим грузиком на дне опустил в мерный стакан. Отметил уровень воды в стакане. Досыпал в пробирку соли. Уровень воды в стакане изменился на 28 мм. С помощью нитки Семён измерил внешний периметр сечения перпендикулярного оси пробирки, получив 45 мм. Сколько граммов соли в пробирке? Плотность воды  $1000 \text{ кг/м}^3$ .

*Решение:*

Плотность воды  $1000 \text{ кг/м}^3$ .

В первом случае условие равновесия пробирки

Во втором случае условие равновесия пробирки

Выразим из этих двух уравнений массу соли:

Площадь сечения пробирки: Здесь  $P$  — периметр окружности пробирки.

В итоге, получаем:

*Ответ:* **4,5 г.**

### **Задача 1.4.3**

*Условие:*

Семен собрал прибор для измерения количества теплоты при обмене энергией между телом и средой. Прибор прост в изготовлении – стальная кружка с водой (примерно половина), теплоизоляция для кружки и воды в ней от окружающей среды. Крышка также из теплоизолирующего материала, с отверстием для градусника. Диапазон измерений градусника от  $+10$ , до  $+80$  градусов Цельсия. Цена деления термометра  $2 \text{ }^\circ\text{C}$ . Масса стаканчика 70 г. К термометру Семён приделал подвижную «энергетическую» шкалу с ценой деления 1 кДж. Шкалу откалибровал так, чтобы её ноль совпал с начальной температурой воды в стакане. Затем опустил в стакан испытуемое тело и после установления теплового равновесия определил по энергетической шкале, сколько джоулей получило или отдало тело в результате теплообмена с прибором. Вопрос:

1. Сколько воды было в приборе, если одному делению шкалы термометра соответствует одно деление энергетической шкалы?
2. Определите рабочий диапазон прибора в джоулях, если начальная температура была  $+22 \text{ }^\circ\text{C}$ .

Удельная теплоёмкость стали  $500 \text{ Дж/(кг }^\circ\text{C)}$ , воды  $4200 \text{ Дж/(кг }^\circ\text{C)}$ .

*Решение:*

Обозначим масштаб энергетической шкалы  $\mu$ . Выразим полученное прибором количество теплоты через изменение температуры

- 1) Пусть масса воды равна  $m_0$ . Полученное системой тепло идёт на нагрев воды и стаканчика:  
и
- 2) Поскольку количество полученной теплоты пропорционально изменению температуры, то пределы для количества теплоты находятся из пределов температуры:

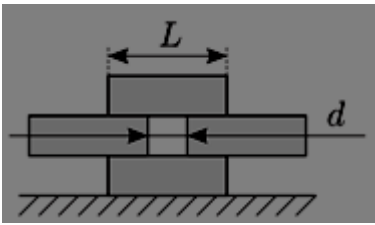
*Ответ:*

Масса воды в приборе **230 г.**, диапазон измеряемых энергий от **-12 кДж** до **58 кДж**.

### **Задача 1.4.4**

*Условие:*

Четыре кирпича длиной 27 см сложены как на рисунке. Найти максимальное расстояние  $d$  между средними кирпичами, если все кирпичи расположены горизонтально. Трением пренебrecь, сила тяжести приложена к центру соответствующего бруска.



Решение:

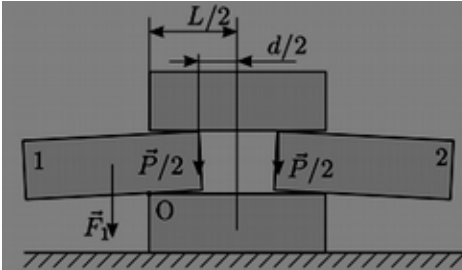


Рис. 10

Система, состоящая из четырех брусков, будет находиться в равновесии, при условии, что сумма моментов внешних сил, действующих на бруски (1) и (2), равна нулю (рис. 10). Запишем правило моментов сил, действующих на брусок (1), относительно точки О. Чтобы яснее представлять место приложения сил, изобразим средние бруски слегка наклонёнными (это положение они займут, если их раздвинуть на расстояние чуть большее, чем  $d$ ). Сила тяжести  $F_1 = mg$  приложена к центру бруска. Поскольку он сдвинут влево на расстояние  $d/2$ , то и плечо силы тяжести равно  $d/2$ . Вес  $P = mg$  верхнего бруска приложен к верхним рёбрам брусков (1) и (2) и, следовательно, распределён между ними поровну (к каждому ребру приложена сила  $P/2$ ). Плечо этой силы относительно точки О равно  $(\frac{L}{2} - \frac{d}{2})$ . Согласно правилу моментов:

$$mg \cdot \frac{d}{2} = \frac{mg}{2} \cdot \left( \frac{L}{2} - \frac{d}{2} \right).$$

Отсюда выражаем  $d$ :

$$d = \frac{L}{3}.$$

Ответ:  $d = 27/3 = 9$  см.

## 1.5 Вторая попытка Задачи по физике (10-11 класс)

### Задача 1.5.1

Условие:

С какой минимальной скоростью нужно бросить камень, чтобы он перелетел квадратный сарай со стороной 3м.

Решение:

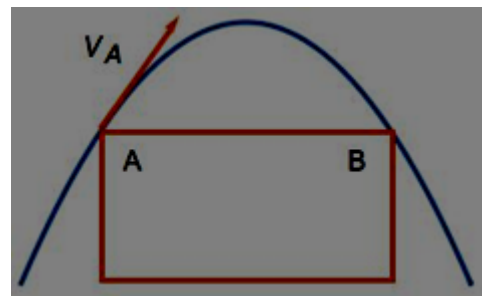
Скорости в исходной точке и точке А связаны законом сохранения, тогда из минимальности начальной скорости следует минимальность скорости в точке А:

где  $a$  – длина стороны квадрата,

В точке А минимальность определяется из классической задачи – бросаем на одном уровне из А в В:

минимальна если синус максимален, то есть угол равен 45 градусов:

Тогда:



Ответ: 9 м/с.

### Задача 1.5.2

Условие:

Частица массой  $m_1=40г$ , движущаяся со скоростью  $u = 1.73$  м/с налетает на покоящуюся частицу массой  $m_2=20г$  и после абсолютно упругого соударения отскакивает от нее под углом  $\alpha = 30^\circ$  к направлению своего первоначального движения. С какой скоростью начнет двигаться вторая частица? Ответ укажите с точностью до единиц м/с.

Решение:

Частица массой  $m_1=40г$ , движущаяся со скоростью  $u = 1.73$  м/с налетает на покоящуюся частицу массой  $m_2=20г$  и после абсолютно упругого соударения отскакивает от нее под углом  $\alpha = 30^\circ$  к направлению своего первоначального движения. С какой скоростью начнет двигаться вторая частица?

По закону сохранения импульса в проекции на оси

$$mv = mv_1 \cos \alpha + \frac{m}{2} v_2 \cos \beta \quad (1)$$

$$0 = mv_1 \sin \alpha + \frac{m}{2} v_2 \sin \beta \quad (2)$$

По закону сохранения энергии

$$\frac{mv^2}{2} = \frac{mv_1^2}{2} + \frac{mv_2^2}{4} \quad (3)$$

Решая совместно, найдем

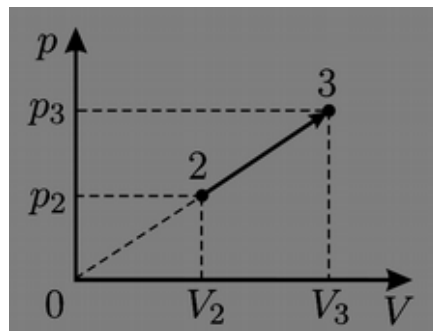
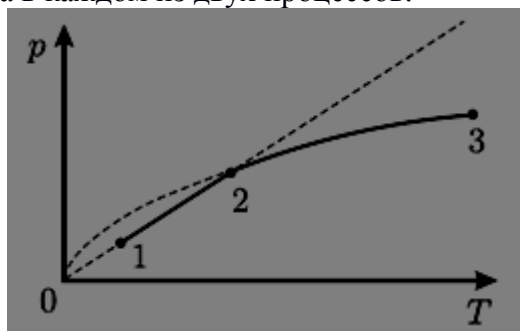
$$v_2 = \frac{2v}{\sqrt{3}} = 1,17v = 2 \text{ м/с}$$

Ответ: 2 м/с.

### Задача 1.5.3

Условие:

Один моль идеального одноатомного газа участвует в процессах: 1–2–3. В первом из них, во втором, где и – некоторые положительные константы. Определите теплоёмкость газа в каждом из двух процессов.



Решение:

Идеальный одноатомный газ ( $\nu = 1$  моль) участвует в процессах: 1–2–3. В первом из них  $p = \alpha T$ , во втором  $p = \beta \sqrt{T}$ , где  $\alpha$  и  $\beta$  – некоторые положительные константы. Определите теплоёмкость газа в каждом из двух процессов.

Процесс 1-2 изохорный

$$C_{12} = \frac{i}{2} \nu R = \frac{3}{2} R = 12,5 \frac{\text{Дж}}{\text{К}}$$

Процесс 2-3:  $T = \frac{p^2}{\beta^2}$ , . Запишем уравнение Менделеева-Клапейрона

$pV = R \frac{p^2}{\beta^2}$ ,  $p = \frac{\beta^2}{R} V$ , процесс линейный (см. рисунок). Первое начало термодинамики для 2-3:

$$Q_{23} = \Delta U_{23} + A_{23} = \frac{3}{2} R \Delta T_{23} + \frac{1}{2} p_3 V_3 - \frac{1}{2} p_2 V_2 = \frac{3}{2} R \Delta T_{23} + \frac{1}{2} R \Delta T_{23} = 2 R \Delta T_{23}$$

По определению теплоёмкости:

$$C_{23} = \frac{Q_{23}}{\Delta T_{23}} = 2 R = 16,6 \frac{\text{Дж}}{\text{К}}$$

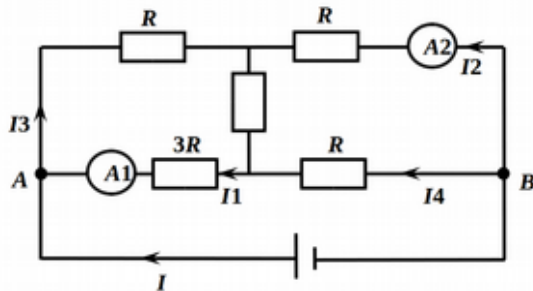
Ответ:

$$C_{12} = \frac{3}{2} R = 12,5 \frac{\text{Дж}}{\text{К}}, \quad C_{23} = 2 R = 16,6 \frac{\text{Дж}}{\text{К}}.$$

#### Задача 1.5.4

Условие:

В схеме, показанной на рисунке, амперметр  $A_1$  показывает силу тока  $I_1 = 1$  А. Какую силу тока показывает амперметр  $A_2$ ? Все приборы идеальные. Указанные на рисунке сопротивления считать известными.



Решение:

Сумма токов в узле, по закону Кирхгофа, равна нулю, тогда для узлов А и В:

$$I = I_1 + I_3 = I_2 + I_4 \quad (1)$$

Разность потенциалов  $U_{AB} = I_3 R + I_2 R = R(I_2 + I_3)$

В то же время  $U_{AB} = I_1 3R + I_4 R$

Приравняем:  $I_3 R + I_2 R = I_1 3R + I_4 R$

Сократим на R:  $I_3 + I_2 = 3I_1 + I_4 \quad (2)$

Вычтем из уравнения (2) уравнение (1):

$$I_2 - I_1 = 3I_1 - I_2$$

$$4I_1 = 2I_2$$

$$I_2 = 2I_1$$

Ответ: 2 А.

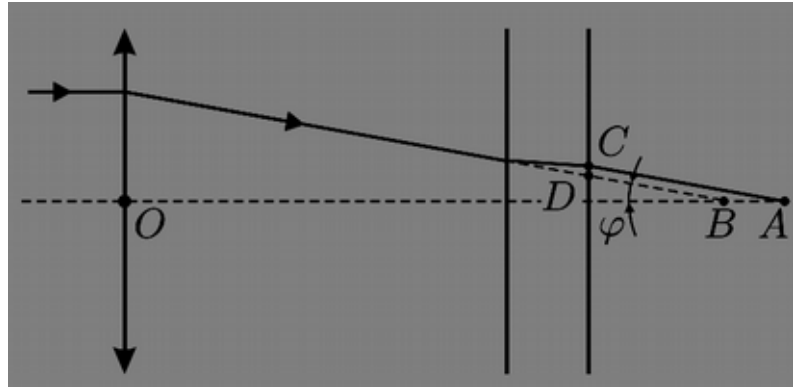
### Задача 1.5.5

Условие:

Между объективом фотоаппарата с фокусным расстоянием  $F = 16$  мм и плёнкой установлен жёлтый светофильтр из стекла толщиной  $d = 1$  мм с показателем преломления  $n = 1,5$ . Фотоаппарат фокусируют на бесконечность, производят съёмку, после чего светофильтр, не меняя положения объектива, убирают. На какое расстояние будет теперь сфокусирован аппарат? Ответ дайте в сантиметрах с точностью до десятых.

Решение:

Изобразим на рисунке оптическую схему. По условию фотоаппарат перед съёмкой со светофильтром был сфокусирован на бесконечность. Это означает, что приходящий из бесконечности луч света, преломляясь в линзе объектива и на пластинке светофильтра, попадает на плёнку (точка А). Обозначим местоположение фокуса линзы объектива (в отсутствие светофильтра) буквой В, точку выхода светового луча из светофильтра буквой С, угол падения луча на светофильтр  $\varphi$ . Будем считать этот угол малым.



Известно, что плоскопараллельная пластинка смещает падающий на неё луч на расстояние  $CD = d(1 - (1/n))\varphi$ , не изменяя направления луча. Тогда из чертежа следует, что  $AB = CD/\varphi = d(1 - (1/n))$ , и расстояние от объектива до плёнки:

$$OA = F + AB = F + d\left(1 - \frac{1}{n}\right)$$

Когда уберём светофильтр на плёнку будут попадать лучи, приходящие от объектов, расположенных на некотором расстоянии  $l$  от объектива. Применяя формулу тонкой линзы, получим:

$$l = \left(\frac{1}{F} - \frac{1}{OA}\right)^{-1} = F + \frac{F^2 n}{d(n-1)} = 78.4 \text{ см}$$

Ответ:  $l = 78.4$  см.

## 1.6 Вторая попытка Задачи по информатике

### Задача 1.6.1 (2 балла)

Условие:

В разные моменты времени в систему управления умным домом поступают различные команды на выполнение той или иной операции. Для оптимального распределения нагрузки на систему каждой операции, которую должна выполнить система, присваивается приоритет выполнения в виде целого числа  $P$ . При поступлении каждой команды на выполнение того или иного действия система добавляет её в очередь. Выполнение операций не начинается до тех пор, пока в очереди не наберётся ровно  $K$  команд. Все эти команды выполняются в порядке не возрастания приоритетов  $P$ . В случае



если две команды имеют одинаковый приоритет, они выполняются в том порядке, в котором поступали на вход системе. На время выполнения всех  $K$  команд, накопившихся в очереди, система перестаёт принимать новые запросы. Когда все запросы из очереди выполнены, система ждёт следующие  $K$  команд для выполнения.

Ваша задача по заданной последовательности поступления запросов с известными приоритетами определить последовательность, в которой будут выполнены запрошенные операции.

Обратите внимание, что последние команды могут никогда не выполняться, если количество запросов не кратно  $K$ .

### Формат входных данных

В первой строке заданы два целых числа  $N$  – количество обрабатываемых запросов, и  $K$  – количество команд в очереди, необходимое для запуска выполнения запрошенных операций ( $1 \leq N \leq 10^5, 1 \leq K \leq N$ ).

В следующих  $N$  строках содержатся по два целых числа  $i$  – номер запроса, и  $P_i$  – приоритет  $i$ -ой операции ( $1 \leq P_i \leq 10^5$ ). Номера запросов  $i$  эквивалентны их порядковому номеру и даны во входных данных в порядке возрастания от 1 до  $N$ .

### Формат выходных данных

Через пробел выведите номера запросов  $i$  в том порядке, в котором они будут выполнены. Номера запросов, которые выполнены не будут, выводить не надо.

### Пример

Входные данные	Выходные данные
10 3 1 8 2 4 3 7 4 2 5 3 6 4 7 8 8 7 9 8 10 1	1 3 2 6 5 4 7 9 8

### Решение:

Для решения данной задачи необходимо реализовать быстрый алгоритм сортировки с вычислительной сложностью  $O(N \cdot \log(N))$ . С помощью этого алгоритма последовательности из  $K$  входных пар  $\langle id_i, P_i \rangle$  необходимо сортировать по убыванию величины  $P_i$  и выводить в ответ идентификаторы  $id_i$  в полученном порядке.

### Код решения (C++)

```
#define _CRT_SECURE_NO_WARNINGS
#pragma comment(linker, "/STACK:4000000")
```

```

#include <iostream>
#include <cstdio>
#include <string>
#include <vector>
#include <utility>
#include <algorithm>
#include <ctime>

using namespace std;

int main()
{
    int n, k;
    scanf("%d%d\n", &n, &k);
    vector <pair<int, int>> ar(k);

    int cnt = 0;
    for (int i = 0; i < n; i++)
    {
        scanf("%d%d\n", &ar[cnt].second, &ar[cnt].first);
        ar[cnt].first -= 2 * ar[cnt].first;
        cnt++;
        if (cnt == k)
        {
            cnt = 0;
            sort(ar.begin(), ar.end());
            for (int j = 0; j < k; j++)
            {
                printf("%d ", ar[j].second);
            }
        }
        printf("\n");

        return 0;
    }
}

```

### Задача 1.6.2 (2 балла)

*Условие:*

Для принятия решения о необходимости включения отопления система управления умным домом использует следующий подход. В каждой из  $R$  комнат установлены  $S$  датчиков температуры. Каждый из датчиков может находиться в одном из двух состояний: 0 – в комнате холодно, 1 – в комнате тепло. Если все датчики в одной из комнат находятся в одном и том же состоянии, решение о необходимости включения или выключения отопления может быть принято однозначно. Если же показания датчиков в комнате расходятся, то для принятия решения о включении или выключении отопления системе приходится действовать в соответствии с определенным алгоритмом.

Одним из критериев для принятия такого решения является наименьшее количество датчиков  $C$ , которые должны сменить своё состояние (с 0 на 1 или наоборот) для того, чтобы показания всех датчиков в комнате совпадали.

Ваша задача – подсчитать значение  $C$  для каждой из  $R$  комнат.

### Формат входных данных

В первой строке заданы два целых числа  $R$  – количество комнат, и  $S$  – количество датчиков в каждой комнате ( $1 \leq R, S \leq 10^3$ ).

В следующих  $R$  строках содержится по  $S$  чисел  $t_{j,i}$  равных 0 или 1, характеризующих состояние  $i$ -го датчика в  $j$ -ой комнате.

### Формат выходных данных

Через пробел выведите  $R$  чисел  $C_j$  – наименьшее количество датчиков, которые должны сменить своё состояние для того, чтобы показания всех датчиков в  $j$ -ой комнате совпадали.

### Пример

Входные данные	Выходные данные
2 4 0 1 1 0 1 0 1 1	2 1

### Решение:

Для решения данной задачи будем подсчитывать количество нулей и единиц в каждой строке. Ответом  $C_j$  для каждой комнаты будет являться минимальное число из количества нулей или единиц в соответствующей строке.

### Код решения (C++)

```
#define _CRT_SECURE_NO_WARNINGS
#pragma comment(linker, "/STACK:4000000")

#include <iostream>
#include <cstdio>
#include <string>
#include <vector>
#include <utility>
#include <algorithm>
#include <ctime>

using namespace std;

int main()
{
    int r, s;

    scanf("%d%d\n", &r, &s);

    for (int i = 0; i < r; i++)
    {
        int zeros = 0, ones = 0;
```

```

    for (int j = 0; j < s; j++)
    {
        int symbol;
        scanf("%d", &symbol);
        symbol ? ones++ : zeros++;
    }
    printf("%d ", min(ones, zeros));
}

printf("\n");

return 0;
}

```

### Задача 1.6.3 (3 балла)

*Условие:*

Устройства, входящие в систему умного дома, могут находиться в определенной зависимости друг от друга. Например, медиа-система не может полноценно функционировать без колонок и медиа-сервера. Медиа-сервер, в свою очередь, может нуждаться в хранилище данных, содержащем медиа-контент. Для обновления библиотеки музыки этому хранилищу требуется интернет, который обеспечивается маршрутизатором. Но маршрутизатор обслуживает не только хранилище данных, но и очень много других устройств.

При выходе из строя одного из устройств, в нерабочее состояние по цепочке переходят и все зависимые от него приборы.

В описанной ситуации при поломке хранилища данных перестанет работать медиа-сервер, а за ним и вся медиа-система. Но работа колонок не имеет обратной зависимости от медиа-системы, поэтому они из строя не выйдут. Так же как и маршрутизатор, обеспечивающий доступ в интернет. Таким образом, выход из строя хранилища данных слабо сказался бы на работе всей системы в целом. Другое дело – поломка маршрутизатора. От него зависит большое количество устройств, работающих под управлением умного дома, так что такая поломка была бы намного критичнее.

Ваша задача по заданным зависимостям определить какое количество устройств перестанет работать при выходе из строя одного из них.

#### Формат входных данных

В первой строке дано одно число  $N$  – количество различных устройств ( $2 \leq N \leq 100$ ).

В следующих  $N$  строках первое число  $K_i$  задаёт количество зависимостей  $i$ -го устройства. Затем дано  $K_i$  чисел  $d_{i,j}$  – номера устройств, от которых зависит  $i$ -е устройство ( $d_{i,j} \neq i$ ).

Все устройства нумеруются целыми числами от 1 до  $N$ . Гарантируется, что цепочки зависимостей не замкнуты, т.е. ни одно устройство не зависит само от себя через какие-либо другие устройства.

#### Формат выходных данных

Через пробел выведите  $N$  чисел  $T_i$  – количество устройств, которые перестанут работать при выходе из строя  $i$ -го устройства.

## Пример

Входные данные	Выходные данные
8 1 4 0 2 2 1 1 7 0 1 7 0 1 7	1 1 0 2 0 0 5 0

### *Решение:*

Для заданных ограничений задачу можно решить, реализовав любой обход графа (в ширину или в глубину) от каждой из вершин, подсчитывая при этом количество достижимых вершин, которое и будет являться ответом.

### Код решения (C++)

```
#define _CRT_SECURE_NO_WARNINGS
#pragma comment(linker, "/STACK:4000000")

#include <iostream>
#include <cstdio>
#include <string>
#include <vector>
#include <utility>
#include <algorithm>
#include <ctime>

using namespace std;

int main()
{
    int n;
    scanf("%d\n", &n);

    const int inf = 100000000;

    vector<vector<int>> g(n, vector<int>(n, inf));

    for (int i = 0; i < n; i++)
    {
        int v1 = i;
        int ki;
        scanf("%d", &ki);
```

```

    for (int k = 0; k < ki; ++k)
    {
        int v2;
        scanf("%d", &v2);
        --v2;
        g[v2][v1] = 1;
    }
}

for (int k = 0; k < n; k++)
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            g[i][j] = min(g[i][j], g[i][k] + g[k][j]);

for (int i = 0; i < n; i++)
{
    int cnt = 0;
    for (int j = 0; j < n; j++)
    {
        g[i][j] != inf ? cnt++ : 0;
    }
    printf("%d ", cnt);
}

printf("\n");

return 0;
}

```

### Задача 1.6.4 (3 балла)

*Условие:*

В современном мире часто приходится сталкиваться с вопросами конфиденциальности. Если какие-либо данные передаются по беспроводному каналу в незашифрованном виде, злоумышленник легко может перехватить сигнал и завладеть вашей информацией. В системах умного дома устройства постоянно передают различные данные. С целью обеспечения сохранности важной информации, все эти данные необходимо шифровать.

В рамках данной задачи о методах шифрования, используемых в упомянутой системе умного дома, вам надо знать одно – для работы им необходимы простые числа. Ваша задача написать программу, которая будет вычислять  $n$ -е по счету простое число.

#### Формат входных данных

В первой строке дано целое число  $k$  – количество простых чисел, которые необходимо найти ( $1 \leq k \leq 10000$ ).

В следующих  $k$  строках содержится по одному целому числу  $n_i$  – номер простого числа, которое необходимо вычислить ( $1 \leq n_i \leq 10000$ ).

#### Формат выходных данных

Выведите  $k$  строк, содержащих по одному целому числу  $p_i$  –  $n_i$ -ое по счету простое число.

## Пример

Входные данные	Выходные данные
3	5
3	11
5	13
6	

### Решение:

Для решения данной задачи при заданных ограничениях необходимо реализовать алгоритм быстрого поиска  $n$ -го простого числа. Таким алгоритмом является Решето Эратосфена. Каждое найденное простое число будем складывать по порядку в массив  $A$  с 1-индексацией. Найдя первые 10000 простых чисел, ответ на каждый запрос получается простым обращением к соответствующему элементу массива  $A[n_i]$ .

### Код решения (C++)

```
#define _CRT_SECURE_NO_WARNINGS
#pragma comment(linker, "/STACK:4000000")

#include <iostream>
#include <cstdio>
#include <string>
#include <vector>
#include <utility>
#include <algorithm>
#include <ctime>

using namespace std;

void prime(vector <int> &primes)
{
    primes.resize(0);
    primes.reserve(100000);
    bool f[1000001] = { 1, 1 };
    for (int i = 2; i < 500001; i++)
    {
        if (f[i])
            continue;

        primes.push_back(i);
        for (int j = i + i; j < 1000001; j += i)
            f[j] = 1;
    }
}

int main()
```

```

{
    vector<int> primes;
    prime(primes);

    int k;
    scanf("%d\n", &k);

    for (int i = 0; i < k; i++)
    {
        int n;
        scanf("%d\n", &n);
        printf("%d\n", primes[n - 1]);
    }

    return 0;
}

```

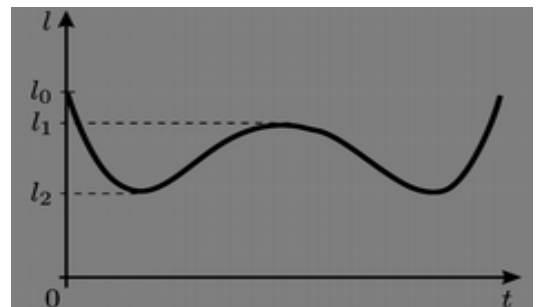
## §2 Второй отборочный этап

### 2.1 Задачи по физике

#### Задача 2.1.1 (5 баллов)

*Условие:*

С помощью лазерного дальномера была записана информация о расстоянии до тела, брошенного вертикально вверх с поверхности земли. Зависимость расстояния  $l(t)$  между этим телом и неподвижным наблюдателем приведена на графике. На какой высоте над землёй находится наблюдатель? На каком расстоянии от линии, по которой движется тело, находится наблюдатель? Чему равна начальная скорость тела? Величины  $l_0=10\text{м}$ ,  $l_1=8\text{м}$  и  $l_2=4\text{м}$ , ускорение свободного падения  $g=10\text{м/с}^2$ . Ответ округлите до целых.



*Решение:*

Пусть наблюдатель находится на высоте  $h$  и на расстоянии  $a$  от линии, по которой движется тело. При бросании тела возможны два случая:

1. тело не долетает до высоты, на которой находится наблюдатель – в этом случае расстояние  $l$  от тела до наблюдателя сначала уменьшается, а затем увеличивается;
2. тело поднимается выше наблюдателя – в этом случае расстояние  $l$  сначала уменьшается от  $\sqrt{a^2+h^2}$  до  $a$ , затем увеличивается до  $\sqrt{a^2+(H-h)^2}$ , где  $H$  – высота подъёма тела, а потом опять уменьшается до  $a$  и увеличивается до  $\sqrt{a^2+h^2}$ .

Как видно из приведённого в условии рисунка, реализуется именно второй случай. При этом

$$l_0 = \sqrt{a^2 + h^2}, l_1 = \sqrt{a^2 + (H - h)^2}.$$

Отсюда находим:  $a = 12$ ,  

$$h = \sqrt{l_0^2 - a^2} = \sqrt{l_0^2 - l_2^2} = 9 \text{ м}$$

и  

$$H = h + \sqrt{l_1^2 - a^2} = \sqrt{l_0^2 - l_2^2} + \sqrt{l_1^2 - l_2^2} = 16 \text{ м}.$$