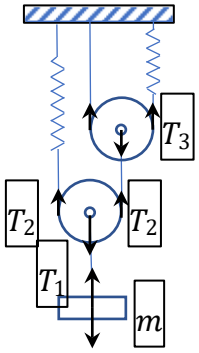
§3 Заключительный этап: индивидуальный тур

3.1 Задачи по физике (9 класс)

За каждую решенную задачу начислялось 10 баллов.

Задача 1. Система невесомых блоков с пружинами изображена на рисунке. Жёсткости пружин известны ($kk_1 = 10$ Н/м и $kk_2 = 50$ Н/м), трения в блоках нет. К нижнему блоку подвесили груз массы *m=400г* и аккуратно отпустили так, чтобы система пришла в равновесие. На сколько опустится нижний блок?



Решение.

Для груза массы m: $TT_1 = mg$ Для нижнего блока $2T_2 = T_1$· Для верхнего блока $2T_3 = T_2$· Тогда сила с которой растянута пружина 1: $T_2 = mg/2$· Сила с которой растянута пружина 2: $T_3 = mg/4$ По закону Гука найдём растяжение пружин:

$$\Delta x_1 = \frac{mg}{2k_1}, \quad \Delta x_2 = \frac{mg}{4k_2}.$$

Смещение верхнего блока $\Delta x_2/2$. Смещение нижнего блока

$$\frac{1}{2}\left(\frac{\Delta x_2}{2} + \Delta x_1\right) = \frac{1}{2}\left(\frac{mg}{8k_2} + \frac{mg}{2k_1}\right) = \frac{mg}{4}\left(\frac{1}{4k_2} + \frac{1}{k_1}\right) = \frac{0.4 \cdot 10}{4}\left(\frac{1}{4 \cdot 50} + \frac{1}{100}\right) = 0{,}015\text{м}.$$

Ответ: 1.5 см.

Критерии оценивания (до 10 баллов)

1. Правильно расставлены силы 1б.
2. Правильно найдены силы натяжения пружин 2б
3. Найдено Смещение верхнего блока 2б
4. Найдено Смещение нижнего блока 3б
5. Получен правильный ответ 2б

Задача 2. Плоский цилиндр нагревают до температуры 60℃. Если его после этого положить на горизонтальную поверхность льда имеющего температуру 0℃, то лёд будет плавиться и цилиндр будет погружаться в лёд. На сколько цилиндр погрузится в лёд в процентном отношении к своей толщине? Удельная теплоёмкость материала цилиндра C =

400 Дж/(кг · °C), плотность $\rho = 9$ г/см^3, удельная теплота плавления льда $\lambda = 3{,}4 \cdot 10^5$ Дж/кг, плотность льда $\rho_0 = 0{,}9$ г/см^3.

Решение:

Количество теплоты на охлаждение цилиндра равно количеству теплоты на плавление льда:

$$Cm_1\Delta T = \lambda m_2$$
$$C\rho\ V_1\Delta T = \lambda\rho_0 V_2$$
$$C\rho\ Sh\Delta T = \lambda\rho_0 Sx,$$

где x – глубина погружения цилиндра в лёд, h – высота цилиндра, S – площадь основания цилиндра.

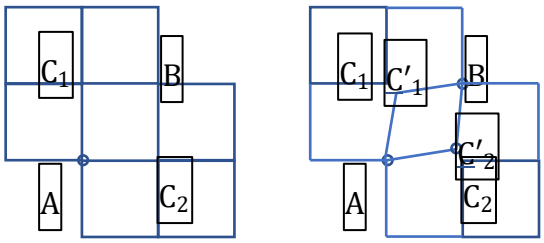
$$\frac{x}{h} = \frac{C\Delta T\rho}{\lambda\rho_0} = \frac{400 \cdot 60 \cdot 9}{340000 \cdot 0.9} = 0.71 = \mathbf{71\%}.$$

Ответ: **71%.**

Критерии оценивания (до 10 баллов)

1. Записано уравнение теплового баланса 2б.
2. Выражен объём через плотность для цилиндра и растаявшего льда 3б.
3. Произведены все преобразования и получен ответ 5б.

Задача 3. Электрическая цепь образует сетку, стороны квадрата которой имеют сопротивление 14 Ом. Чему равно сопротивление между точками A и B?



Решение.

Разъединим проводники в точках C_1 и C_2. В силу симметрии схемы, напряжение между точками C_1 и C'_1, C_2 и C'_2, равно нулю, то есть две схемы на рисунке эквивалентны, тогда можно подсчитать:

$$RR_{AAAA} = \frac{5}{7}RR = \mathbf{1100}\ \text{Ом}.$$

Критерии оценивания (до 10 баллов)

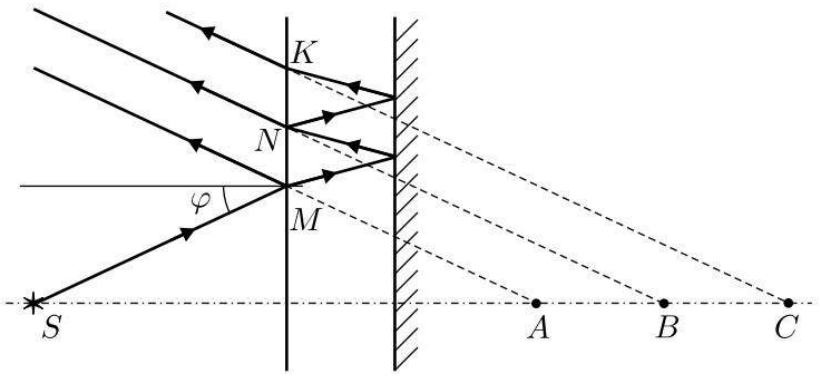
1. Получена эквивалентная схема 5 б

2. Произведён расчёт сопротивления схемы 5б.

Задача 4. Точечный источник света находится вблизи зеркала с посеребрённой задней поверхностью. Толщина зеркала d, показатель преломления n. При малых углах падения в отражении видно основное изображение и два дополнительных меньшей яркости. С помощью рисунка и вычислений покажите, как будут располагаться дополнительные изображения по отношению к основному.

Решение.

Дополнительные изображения формируются за счёт отражения от передней грани зеркала и повторного отражения внутри зеркала.



На рисунке показано, как будут располагаться дополнительные изображения A и C. Если угол $\varphi\varphi$ падения световых лучей на зеркало мал, то

$$MN \approx \frac{2\varphi d}{n} \approx NK, \quad AB = BC \approx \frac{MN}{\varphi} = \frac{\mathbf{2d}}{\boldsymbol{n}}$$

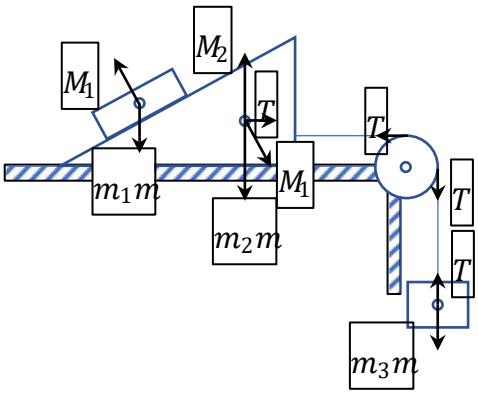
Итак, эти дополнительные изображения будут располагаться **впереди и позади** основного на расстояниях $\boldsymbol{2d/n}$ от него.

Критерии оценивания (до 10 баллов)

1. Приведён рисунок хода лучей 4б
2. Получены соотношения для расстояний между изображениями 6б.

3.2 Задачи по физике (10-11 класс)

За каждую решенную задачу начислялось 10 баллов.



Задача 1. Система, изображённая на рисунке, находится в покое. Трение отсутствует. Найдите ускорения всех тел после начала движения в системе при условии, что все массы одинаковы, а угол наклона плоскости равен $\pi\pi/4$.

Решение.

Расставим силы, действующие на каждое из тел (см. рисунок). Запишем второй закон Ньютона для каждого из тел в проекциях на вертикальную и горизонтальную оси:

$$\text{Тело 3: } m_3 g - T = m_3 a_3;$$

$$\text{Тело 2: } N_2 = m_2 g + N_1 \cos\alpha,$$

$$T + N_1 \sin\alpha = m_2 a_3.$$

Решая систему этих уравнений, получим:

$$T = \frac{m_3}{m_2 + m_3}(m_2 g - N_1 \sin\alpha), \quad a_3 = g\frac{m_3}{m_2 + m_3} + \frac{N_1 \sin\alpha}{m_2 + m_3} \qquad (2.1)$$

Для первого тело необходимо учесть, что его ускорение складывается из двух составляющих: ускорения a_3 наклонной плоскости по отношению к горизонтальной поверхности и ускорения a'_1 самого тела по отношению к наклонной плоскости.

$$\text{Тело 1: } -N_1 \sin\alpha = m_1(-a'_1 \cos\alpha + a_3),$$

$$N_1 \cos\alpha - m_1 g = -m_1 a'_1 \sin\alpha.$$

Выразим отсюда a'_1 и N_1, получим:

$$a'_1 = g\sin\alpha + a_3\cos\alpha, \quad N_1 = m_1 g\cos\alpha - a_3\sin\alpha \qquad (2.2)$$

Подставим (2.2) в (2.1):

$$a_3 = g\frac{m_1\cos\alpha\sin\alpha + m_3}{m_1\sin^2\alpha + m_2 + m_3}.$$

Учитывая что все массы одинаковы, а угол наклона плоскости равен $\pi/4$:

$$a'_1 = \frac{4\sqrt{2}}{5}g = \frac{11.3\text{м}}{\text{с}}, \quad a_3 = \frac{3}{5}g = 6\text{м/с}.$$

Ускорение тела 1 по отношения к горизонтальной плоскости найдём из теоремы косинусов:

$$a_1 = \sqrt{a'^2_1 + a_3^2 - 2a'_1 a_3\cos\alpha} = \frac{\sqrt{17}}{5}g = 8.2\text{м/с}.$$

Ответ: 8.2м/с, 6м/с, 6м/с.

Критерии оценивания (до 10 баллов)

1. Правильно расставлены силы для каждого из тел 3б
2. Правильно записан второй закон Ньютона в проекциях на оси (1+2+1 для первого, второго и третьего тел, соответственно, итого 4б)
3. Найдены ускорения тел 3б.

2. В современной энергетике важную роль играет двигатель Стирлинга. Идеальный цикл Стирлинга состоит из двух изотерм и двух изохор. Рассмотрим процесс изотермического сжатия. Пусть рабочим телом является влажный воздух. В начальном состоянии отношение парциального давления воздуха к парциальному давлению пара равно 1,2. При уменьшении объёма рабочего тела в 3 раза, его давление увеличилось в 2 раза. Определите сколько процентов от первоначального количества пара сконденсировалось?

Решение.

Пусть p – парциальное давление пара в исходном состоянии. Тогда для сухого воздуха начальное давление равно $1.2p$, начальный объём $3V$, конечное давление $p_2^{в}$, конечный объём V. Уравнение Менделеева-Клапейрона в начальном и конечном состоянии для сухого воздуха:

$$1.2\,p\,3V = \nu RT, \quad p_2^{в}V = \nu RT \quad \Rightarrow \quad p_2^{в} = 3.6\,p.$$

Для пара:

$$p\,3V = \nu_1 RT, \quad p_2^{п}V = \nu_2 RT \quad \Rightarrow \quad p_2^{п} = 3p\frac{\nu_2}{\nu_1} = 3p(1-k).$$

Здесь k – доля сконденсировавшегося пара. По условию, давление рабочего тела увеличилось в 2 раза:

$$p_2^{в} + p_2^{п} = 2(1.2p + p),$$
$$3.6\,p + 3p(1-k) = 4.4p.$$

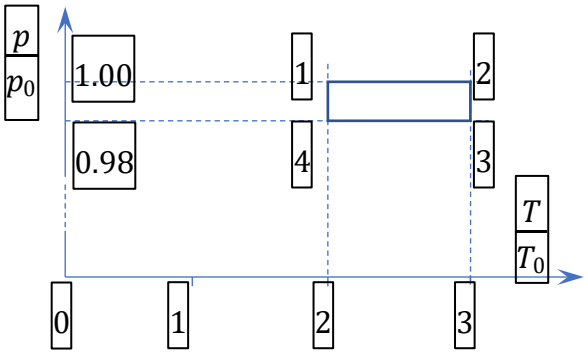
Решая уравнение, получим k=11/15=0.733.

Ответ: **73.3%**.

Критерии оценивания (до 10 баллов)

1. Записан закон идеального газа для начального и конечного состояний пара и сухого воздуха 4б.
2. Правильно используется закон Дальтона 2.
3. Решена система уравнений, получен ответ 4б

3. На pT-диаграмме изображён цикла тепловой машины, у которой рабочим телом является идеальный газ. Работа газа на участке 1-2 равна 100 Дж. Оцените с точностью 2% процента работу газа на остальных участках.



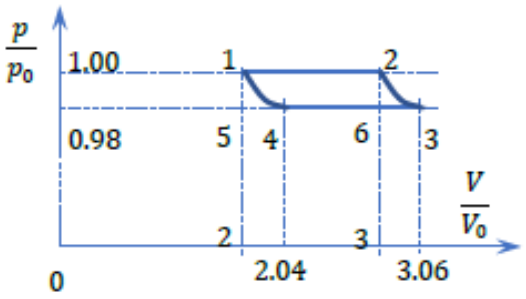
Решение.

Изобразим процесс на pV-диаграмме. Используя уравнение Менделеева-Клапейрона найдём объём во всех состояниях 1, 2, 3, 4: νRT$_0$2VV_0

$$V_1 = 2\frac{\nu RT_0}{p_0} = 2V_0, \quad V_4 = \frac{2V_0}{0.98} \approx 2.04V_0,$$

$$V_2 = \frac{\nu RT_0}{p_0} = 3V_0, \quad V_3 = \frac{3V_0}{0.98} \approx 3.06V_0,$$

где $V_0 = \nu RT_0/p_0$. Работы газа на участках 1-2 и 3-4 соответственно равны



$$A_{1-2} = p_0(V_2 - V_1) = p_0V_0 = 100 \text{ Дж},$$

$$A_{3-4} = 0.98p_0(V_4 - V_3) = -p_0V_0 = -100 \text{Дж}.$$

Участках 1-4 и 3-2 pV-диаграммы приближённо можно считать прямыми. Тогда:

$$A_{2-3} \approx \frac{p_2 + p_3}{2}(V_3 - V_2) = 0.99p_0\left(\frac{3V_0}{0.98} - 3V_0\right) \approx 6.06 \text{ Дж},$$

$$A_{4-1} \approx \frac{p_1 + p_4}{2}(V_1 - V_4) = 0.99p_0\left(2V_0 - \frac{2V_0}{0.98}\right) \approx -4.04 \text{ Дж}.$$

Погрешности этих величин не превосходят площадей треугольников 236 и 145 соответственно. При этом относительные погрешности составляют

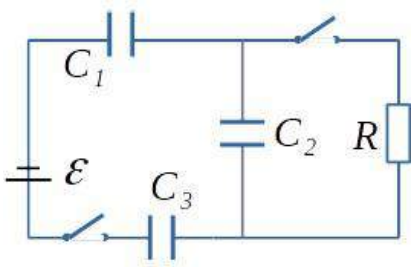
$$\frac{\Delta A_{2-3}}{|A_{2-3}|} < \frac{S_{236}}{|A_{2-3}|} = \frac{p_2 - p_3}{p_2 + p_3} < 0.02,$$

$$\frac{\Delta A_{1-4}}{|A_{1-4}|} < \frac{S_{145}}{|A_{1-4}|} = \frac{p_1 - p_4}{p_1 + p_4} < 0.02.$$

Критерии оценивания (до 10 баллов)

1. Цикл перерисован в диаграмме pV 2б.
2. Найдены объёмы в состояниях 1-4 2б.
3. Найдена работа A_{3-4} 1б
4. Найдены оценки работ AA_{2-3}, AA_{4-1} 3б
5. Получена оценка погрешности 2б

4. Какой заряд протечёт через резистор R после одновременного замыкания ключей. До замыкания конденсаторы C_1, C_3 не заряжены, а конденсатор C_2 заряжен до разности потенциалов U_0. Величины C_1, C_2, C_3, U_0, E считать известными.



После замыкания ключей конденсатор C_2 разряжается через R, а C_1 и C_3 заряжаются зарядом q_1:

$$\frac{q_1}{C_1}+\frac{q_1}{C_2}=\varepsilon, \qquad \rightarrow \quad q_1=\varepsilon\frac{C_1C_2}{C_1+C_2}$$

Через R пройдёт заряд q_1 и $q_0 = C_2U_0$:

$$\Delta q = q_1 + q_2 = \varepsilon\frac{\boldsymbol{C_1C_2}}{\boldsymbol{C_1+C_2}}+\boldsymbol{C_2U_0}$$

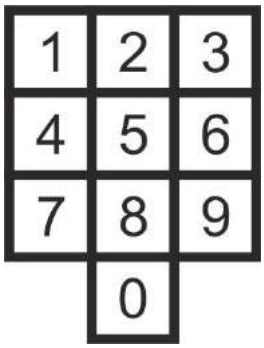
Критерии оценивания (до 10 баллов)

1. Найден заряд qq_1, 4б
2. Найден заряд qq_0 2б
3. Получен ответ 4б

3.3 Задачи информатике

Задача 1. 10 баллов

Виктор Петрович работает аналитиком в компании по установке охранных сигнализаций "Умная безопасность". Устройства, устанавливаемые в квартиры и в жилые дома, поддерживают 4-значные цифровые пароли. Панель для ввода паролей выглядит следующим образом:



В процессе анализа статистики взломанных паролей Виктор Петрович заметил, что клиентов, чьи пароли содержат подстроки из 3 цифр, идущих подряд на одной горизонтальной, вертикальной или диагональной прямой на панели для ввода паролей (например, 1-5-9, 1-2-3, 1-4-7, 5-8-0 и др.), взламывают гораздо чаще. В связи с этим, было принято решение запретить пароли таких конфигураций. Определите количество различных разрешенных вариаций паролей. В поле ответа введите единственное целое число - ответ на задачу.

Примеры запрещенных паролей: 1-5-9-3 (содержит подстроку 1-5-9); 5-9-8-7 (содержит подстроку 9-8-7).

Примеры разрешенных паролей: 2-8-5-0; 1-5-3-9; 2-7-5-9.

Ответ: 9642

Задача 2. 15 баллов

В здании корпорации Гулькин Софт N этажей и L лифтов. На каждом этаже установлено по одной кнопке для вызова лифта. При нажатии на неё активируется Интеллектуальная система управления лифтами, которая была спроектирована таким образом, чтобы от момента нажатия кнопки до момента прибытия сотрудникам корпорации приходилось ждать минимальное время. На самом деле, система просто выбирает самый ближайший свободный лифт и отправляет его на соответствующий этаж. Одним солнечным пятничным утром что-то пошло не так. Казалось, что при нажатии кнопки система назначает совершенно случайный лифт, без какой-либо логики. Доходило до абсурда: если какой-либо лифт уже находится на нужном этаже, система всё равно могла назначить лифт, находящийся где-нибудь ещё. В этот день президент корпорации Гулькин Софт проводил собеседование Петра Игнатьевича на должность ведущего программиста одного из ключевых проектов. В качестве тестового задания Петру Игнатьевичу было поручено разобраться с Интеллектуальной системой управления лифтами.

Так уж вышло, что Пётр Игнатьевич думал, что проходит собеседование на должность младшего аналитика рынка ценных бумаг, ведь в резюме он чётко указал, что имеет опыт в этой области и вообще очень любит теорию вероятностей и математическую статистику. В связи с этим, Пётр Игнатьевич воспринял задание "разобраться с Интеллектуальной системой управления лифтами" следующим образом: Рассчитать математическое ожидание* времени от момента нажатия кнопки вызова лифта до момента его прибытия, если известны время перемещения каждого лифта на один этаж и текущее расположение всех лифтов. Считать, что система может назначить любой из L лифтов с равной вероятностью.

Пётр Игнатьевич смог справиться с этой задачей, а сможете ли вы?

* *Математическое ожидание* - среднее значение случайной величины при стремлении количества выборок или количества её измерений к бесконечности.

Входные данные

В первой строке даны три целых числа: количество этажей N, количество лифтов L и время T, за которое каждый лифт перемещается на один этаж ($2 \le N \le 10^3$; $1 \le L \le 10^3$; $1 \le T \le 10^5$).

Следующие L строк содержат по одному целому положительному числу - номер этажа, на котором находится каждый из лифтов.

Выходные данные

Выведите N строк в каждой из которых математическое ожидание времени от момента нажатия кнопки вызова лифта на соответствующем этаже до момента его прибытия в виде несократимой дроби (числитель и знаменатель разделены символом "/" без каких-либо дополнительных символов). В первой строке математическое ожидание для первого этажа, во второй для второго этажа и т.д. Если математическое ожидание равно нулю, вывести единственный ноль без символа "/" и без знаменателя.

Пример выходных данных:	Пример выходных данных:
1/1 2/3 1/1	1/1 2/3 1/1

Решение

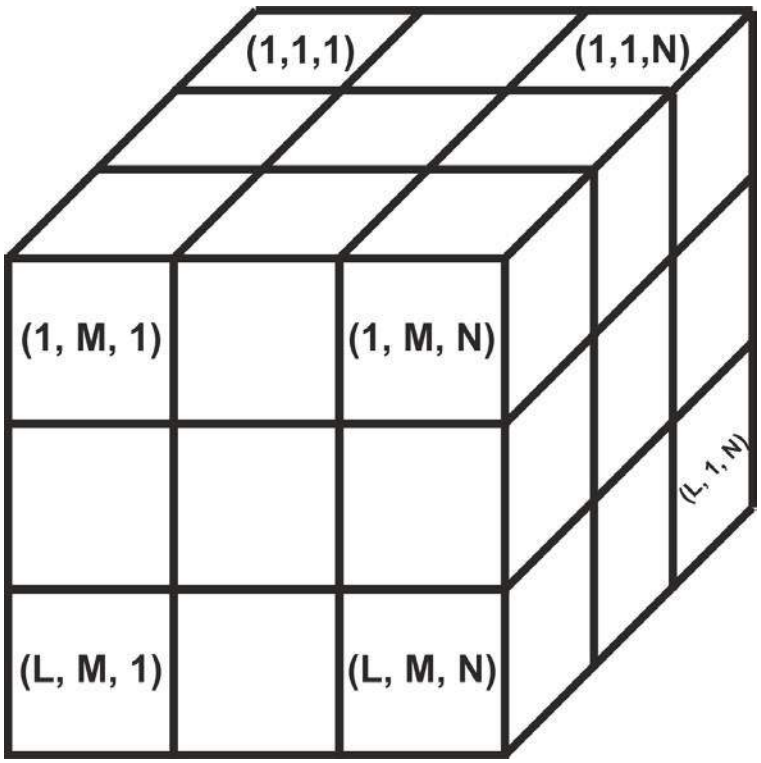
```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <algorithm>
#include <cmath>
```

```
using namespace std;

long long gcd_(long long a, long long b) {
return b ? gcd_(b, a % b) : a;
}  int
main() {
	int n, l, t;
	int ar[10000];

	scanf("%d%d%d", &n, &l, &t);

	for (int i = 0; i < l; i++)
scanf("%d", ar + i);
	for (int i = 0; i < n; i++) {
long long sum = 0;		for (int j = 0;
j < l; j++) {
			sum += abs(i - (ar[j] - 1));
		}

		if (sum == 0)
	printf("0\n");		else
{
			sum *= t;
			long long g = gcd_(l, sum);
printf("%lld/%lld\n", sum / g, l / g);
		}

	}

	return 0;
    }
```

Задача 3. 20 баллов

Для получения доступа к охранной системе умного дома Васи необходимо ввести многоуровневый графический пароль. На самом деле, многоуровневый графический пароль представляет собой трёхмерный лабиринт, состоящий из L слоёв размера $M \times N$. Перемещаться по этому лабиринту можно только в свободные соседние ячейки. Соседними считаются ячейки, одна из координат которых отличается на 1, а две другие совпадают.

Вася очень заботится о безопасности своего жилища, поэтому каждые две недели меняет структуру лабиринта многоуровневого графического пароля. Важно, чтобы новый пароль был корректным, иначе охранная система станет неуправляемой. Несмотря на то, что задуманная Васей структура лабиринта является корректной, в охранную систему он вводит её вручную, а значит, есть вероятность задать в качестве многоуровневого графического пароля некорректную структуру лабиринта. Вася хочет обезопасить себя от подобных ситуаций, поэтому ему нужна программа, которая будет сообщать о том, является ли введённая структура корректной или нет. Вася просит вас написать такую программу.



Пароль считается корректным, если существует путь из ячейки с координатами $(1,1,1)$ в ячейку (L,M,N) (индексация ячеек начинается с 1, см. рисунок выше) причём такой, что при прохождении лабиринта не нужно возвращаться на один из ранее посещённых слоёв (т.е. на пути от начала к концу первая координата не убывает). ,

Входные данные

В первой строке даны три целых числа L,M,N - размеры лабиринта ($1 \le L,M,N \le 100$).

В следующих L раз по M строк содержится N символов описывающих лабиринт. Символ "#" означает, что ячейка является стеной, символ "." - свободная ячейка. Описание всего лабиринта ведётся в порядке возрастания координат. То есть самый первый символ первой строки описания описывает ячейку $(1,1,1)$, последний символ этой же строки описывает ячейку $(1,1,N)$, а ячейка (L,M,N) описывается самым последним символом последней строки.

Выходные данные

В единственной строке выведите "YES", если структура корректна и "NO" в противном случае. Кавычки выводить не нужно.

Пример входных данных 1:

3 3 3
.#.
.#.

##.
#.# ...
...
.## ...

Пример выходных данных 1:
YES

Пример входных данных 2:
3 3 3
.#.
.#.
##.
#..
..# ##.
...
...

Пример выходных данных 2:
NO

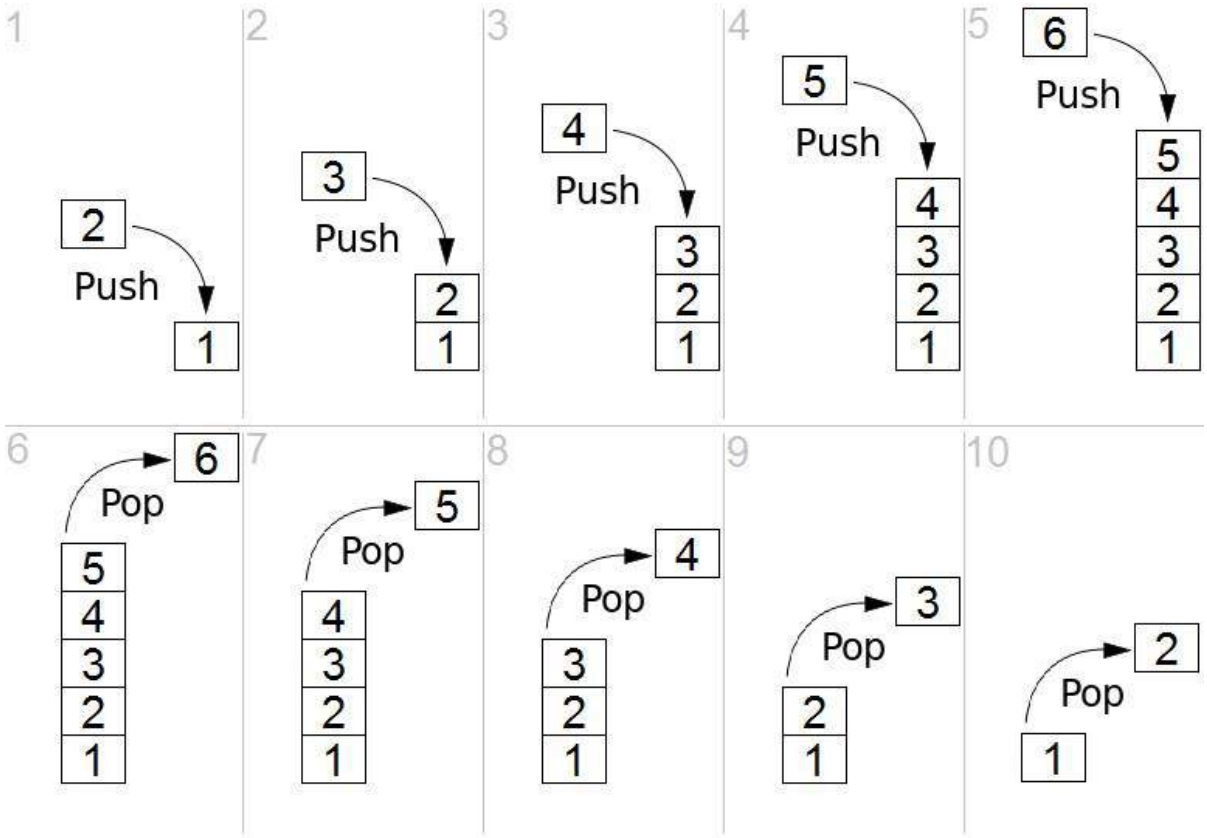
Решение

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <algorithm>
#include <cmath>

using namespace std;

long long gcd_(long long a, long long b)
{     return b ? gcd_(b, a % b) :
a;
}  struct
point_3d {
int x, y, z;

    point_3d operator+(const point_3d p)
    {
        return { x + p.x, y + p.y, z + p.z };
    }
};

point_3d directions[] = {
    { 1, 0, 0 },
{ 0, 1, 0 },
```

```
    { 0, -1, 0 },
{ 0, 0, 1 },
    { 0, 0, -1 },
};  int l, m, n; char
map[100][100][101]; bool
visited[100][100][100] = {};

#define IN_RANGE(val,range) ((val)>=0&&(val)<(range))
#define VALID_POS(p) (IN_RANGE(p.x,l)&&IN_RANGE(p.y,m)&&IN_RANGE(p.z,n))
#define AVAILABLE(p) (map[p.x][p.y][p.z]=='.') #define
VISITED(p) (visited[p.x][p.y][p.z])
 void dfs(point_3d
p)
{     visited[p.x][p.y][p.z] =
true;

    for (int i = 0; i < 5; i++) {
point_3d new_pos = p + directions[i];
        if (!VALID_POS(new_pos) || !AVAILABLE(new_pos) || VISITED(new_pos))
continue;         dfs(new_pos);
    }
}  int
main() {
    scanf("%d%d%d", &l, &m, &n);

    for (int x = 0; x < l; x++)
for (int y = 0; y < m; y++)
scanf("%s", map[x][y]);

    dfs({ 0,0,0 });
     printf("%s\n", visited[l - 1][m - 1][n - 1] ? "YES" :
"NO");

    return 0;
 }
```

Задача 4. 25 баллов

Стек - структура данных, представляющая собой список элементов, организованных по принципу "последний пришёл - первый ушёл". Вершиной стека будем называть последний пришедший элемент. При добавлении очередного элемента в стек, он становится новой вершиной. При удалении (извлечении) элемента из стека извлекается вершина, а новой вершиной становится элемент, который был добавлен непосредственно перед удалённым. Если в стеке не осталось данных, будем говорить, что стек пуст.



Для организации управления умным домом используется стек задач. В произвольный момент времени в стек добавляются различные задачи. Например: закрыть шторы, включить свет, настроить систему отопления и т.д.

Если стек не пустой, сервер извлекает из него очередную задачу и посылает сигнал соответствующему устройству. Устройство выполняет задачу и отправляет серверу ответ, что задача выполнена. Получив этот ответ, сервер достаёт очередную задачу и процедура повторяется.

Если изначально стек пуст, и в момент времени T приходит первая задача, в этот же момент времени сервер извлекает эту задачу и начинает её выполнение. Две задачи не могут быть добавлены в стек в один и тот же момент времени.

На выполнение каждой задачи системе требуется затратить некоторое время. В связи с чем порядок выполнения задач становится не совсем очевидным.

Рассмотрим следующую ситуацию.

В момент времени 1 в стек добавляется задача "закрыть шторы", которая затрачивает 3 единицы времени. В момент времени 3 в стек добавляется задача "включить свет", которая затрачивает 1 единицу времени. В момент времени 4 в стек добавляется задача "настроить систему отопления", которая затрачивает 4 единицы времени. Что в это время делает сервер. В момент времени 1, как только пришла задача "закрыть шторы", сервер тут же извлекает её

из стека и начинает обработку. Выполнение этой задачи закончится в момент времени 4. В этот же момент в стеке уже лежат две задачи: "включить свет" и "настроить систему отопления" (обратите внимание, добавление задачи в стек всегда происходит раньше, чем попытка сервером извлечь задачу в этот же момент). Сервер извлечёт задачу "настроить систему отопления" и начнёт её выполнение. Задача будет обработана в момент времени 8. Серверу останется выполнить задачу "включить свет". В момент времени 9 стек полностью пуст и выполнение всех задач завершено.

В рамках данной задачи вам необходимо вывести последовательность выполнения задач сервером.

Входные данные В первой строке даны 2 целых числа N и M ($2 \le N \le 10^5$; $2 \le M \le 10^5$) - количество различных задач и количество добавлений задач в стек соответственно.

В следующих N строках содержится по два целых числа ID_n и T_n и одно слово S_n. ID_n - уникальный номер задачи ($0 \le ID_n \le 10^8$); T_n - время выполнения задачи ($1 \le T_n \le 10^8$); S_n - словесное описание задачи, состоящее из строчных букв латинского алфавита, длиной не более 50 символов.

В следующих M строках содержатся пары чисел ID_m и T_m - номер задачи и момент времени, в который она добавляется в стек. Гарантируется, что номера добавляемых в стек задач ID_m содержатся в описании задач выше. Времена T_m неотрицательные, не превышают 10^8 и даны в порядке возрастания.

Выходные данные Выведите M строк, каждая из которых содержит время начала выполнения задачи и её строковое описание. Времена должны идти в порядке возрастания.

Пример входных данных:	**Пример выходных данных:**
3 3 598 3 curtainsclose 1 4 heatingon 11 1 lighton 598 1 11 3 1 4	1 curtainsclose 4 heatingon 8 lighton

Решение

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <map>

using namespace std;
```

```
 struct task_t
{     int t;
string s;
};  int
main() {
int n, m;
    scanf("%d%d", &n, &m);

    char buf[51];
int id, tmp;

    map<int, task_t> dic;

    for (int i = 0; i < n; i++) {
task_t task;
        scanf("%d%d%s", &id, &task.t, buf);
task.s = buf;         dic[id] = task;
    }
    long long cur_time = 0;
int stack[100000]; // only ids
int top = 0;
    for (int i = 0; i < m; i++) {
scanf("%d%d", &id, &tmp);         while (top
> 0 && cur_time < tmp) {            task_t
&task = dic[stack[--top]];             long
long length = task.t;
            printf("%lld %s\n", cur_time, task.s.c_str());
cur_time += length;
        }
        if (cur_time > tmp)
stack[top++] = id;         else if
(cur_time == tmp || !top) {
task_t &task = dic[id];
            printf("%lld %s\n", tmp, task.s.c_str());
            cur_time = tmp + task.t;
        }
    }
    while (top > 0) {
        task_t &task = dic[stack[--top]];
long long length = task.t;
        printf("%lld %s\n", cur_time, task.s.c_str());
cur_time += length;
    }
    return 0;
}
```

Задача 5. 30 баллов

Братья Антон и Владик любят делать всё сами. Установить систему наблюдения? Сами! Установить автоматические шторы? Сами! Подключить медиа-систему к центральному пульту управления? Сами! Вот и деревянный стол сделать тоже решили сами. Стол готов, а лишних досок и гвоздей осталось ещё много. Чтобы ни доски, ни гвозди не пропадали даром, ребята придумали игру. Каждый взял себе по доске и шариковой ручке. Закрыв глаза в

случайном месте доски ручкой ставится точка. Потом ещё одна точка. И ещё одна. И так N раз. В результате, у каждого на доске появляется по N произвольных точек. В каждую из точек ребята вбивают по 1 гвоздю, таким образом, чтобы половина гвоздя торчала из доски. Дальше, поверх всех гвоздей вбитых в доску натягивается одна замкнутая резинка таким образом, чтобы все гвозди оказались внутри получившегося многоугольника. Было решено, что побеждает тот из ребят, у кого "кучность" попадания ручкой в слепую по доске лучше. Другими словами тот, чья площадь получившегося при натягивании резинки многоугольника меньше. Но вот незадача, визуально площади сравнить оказалось очень непросто. Ваша задача по координатам всех гвоздей, вбитых в доску, определить площадь получившегося описанным образом многоугольника.

Входные данные

В первой строке записано целое число N - количество гвоздей вбитых в доску ($3 \le N \le 10^5$). Далее в N строках по два целых числа X и Y - координаты вбитых гвоздей ($|X|,|Y| \le 10^9$).

Выходные данные

В единственной строке выведите площадь многоугольника. Ответ всегда выводить ровно с 11 знаком после запятой.

Пример входных данных:

```
5
0 0
3 1
2 1
3 5
1 3
```

Пример выходных данных:

```
8.0
```

Решение

```
#pragma comment(linker,"/STACK:128000000")
```

```
#define _CRT_SECURE_NO_WARNINGS #define
_USE_MATH_DEFINES

#include <iostream>
#include <cmath>
#include <algorithm>
 using namespace
std;

typedef struct Point {
long long x, y;
} Point;

Point first;

long long cross_prod(const Point &p1, const Point &p2, const Point &p3) {
    Point v1 = p2;
v1.x -= p1.x;
v1.y -= p1.y;

    Point v2 = p3;
v2.x -= p1.x;
v2.y -= p1.y;

    return v1.x * v2.y - v1.y * v2.x;
}
long long normsqr(const Point &p1, const Point &p2)
{
    long long dx = p2.x - p1.x;
long long dy = p2.y - p1.y;

    return dx * dx + dy * dy;
}  bool pred(const Point &p1, const Point
&p2)
{
    long long cp = cross_prod(first, p2, p1);
    bool res = (cp < 0LL) || (cp == 0 && normsqr(first, p1) < normsqr(first, p2));

    return res;

    //return cross_prod(first, p2, p1) < 0LL;
}  int
main()
{
    Point ar[100000];
Point ar2[100000];
int n;
    scanf("%d\n", &n);
     int first_idx = 0;     for
(int i = 0; i < n; i++)
    {
        scanf("%lld%lld\n", &ar[i].x, &ar[i].y);
if (ar[i].x < ar[first_idx].x)
first_idx = i;
    }
    swap(ar[0], ar[first_idx]);
first = ar[0];

    sort(ar + 1, ar + n, pred);
```

```
    int cnt = 0;
ar2[cnt++] = ar[0];
ar2[cnt++] = ar[1];

    for (int i = 2; i < n; i++)
    {
        while (cnt > 1 && cross_prod(ar2[cnt - 2], ar2[cnt - 1], ar[i]) <= 0)
        {
cnt--;          }
        ar2[cnt++] = ar[i];
    }
    unsigned long long res = 0;

    for (int i = 2; i < cnt; i++)
        res += cross_prod(ar2[0], ar2[i - 1], ar2[i]);

    printf("%lld.%d\n", (res >> 1), (int)(res & 1) * 5);

    return 0;
}
```