3. ЗАКЛЮЧИТЕЛЬНЫЙ ЭТАП

Предметный тур

6.1. Дополненная реальность. Математика. 9 класс

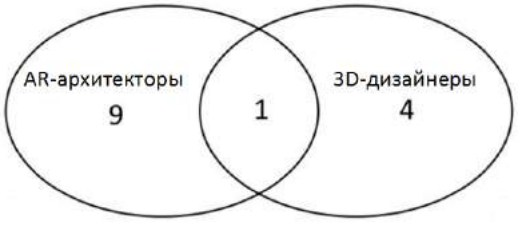
Задача 6.1.1. Кто работает в компании? (24 баллов)

В компании AltFuture работают 20 специалистов разного профиля: 3D-дизайнеры, AR-архитекторы, программисты (никаких других специалистов больше нет). Известно, что каждый десятый AR-архитектор является 3D-дизайнером, каждый пятый 3D-дизайнер - AR-архитектором, а среди программистов треть является – AR-архитекторами и треть – дизайнерами. Сколько всего программистов работает в компании?

Решение

Число AR-архитекторов очевидно кратно 10, поскольку «каждый десятый AR-архитектор является 3D-дизайнером». Значит, их либо 10, либо 20. 20 AR-архитекторов быть не может, поскольку такое их число будет противоречить условию, согласно которому только «каждый пятый 3D-дизайнер является AR-архитектором». Следовательно, AR-архитекторов – 10 человек.

Поскольку «каждый десятый AR-архитектор является 3D- дизайнером», число AR-архитекторов, которые являются 3D- дизайнерами $10 : 10 = 1$ человек. Так как «каждый пятый 3D-дизайнер является AR-архитектором», то в компании $1 \cdot 5 = 5$ 3D-дизайнеров



Тогда 9(чистых AR-архитекторов)+1(AR-архитектор и3D-дизайнер)+4 (чистых 3D-дизайнера) = 14 человек. Получается, что «чистых» программистов $20 - 14 = 6$.

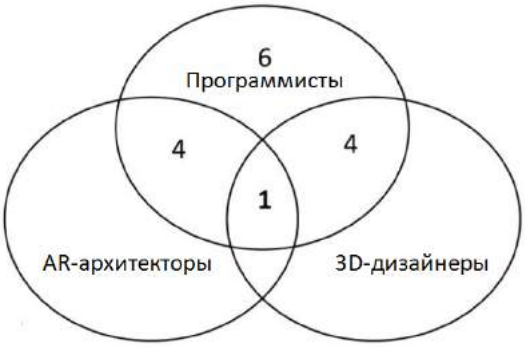
Известно, однако, что не все программисты «чистые»: среди них треть является AR-архитекторами и треть – 3D-дизайнерами. Следовательно, общее число программистов должно быть кратно 3 и их число > 6 (поскольку только чистых программистов уже 6). Значит, их может быть 9, 12, 15 или 18 – последовательно рассмотрим

все 4 варианта.

1. Допустим, общее число программистов – 9. Треть от 9 равна 3, а значит, 3 программиста должны оказаться AR-архитекторами и 3 – 3D-дизайнерами. При этом мы знаем, что 6 программистов – «чистые», т.е. «нечистых» остается всего 3. Получается, что 3 программиста-AR-архитектора и 3 программиста-3D-дизайнера – это одни и те же люди

2. Допустим, общее число программистов – 12. Треть от 12 равна 4, а значит, 4 программиста должны оказаться AR-архитекторами и 4 – 3D-дизайнерами. Учитывая, что «чистых» программистов 6, «нечистых» в сумме должно быть тоже 6. Это возможно только если найдется 2 человека, которые будут одновременно программистами, 3D-дизайнерами и AR-архитекторами.

3. Допустим, общее число программистов – 15. Треть от 15 равна 5, а значит, 5 программистов должны оказаться AR-архитекторами и 5 – 3D-дизайнерами. При этом «чистых» программистов 6, и «нечистых» должно быть 9. Значит, должен быть один человек, который является одновременно программистом, AR-архитектором и 3D-дизайнером.



Это решение удовлетворяет условию задачи.

4. Проверим последний случай, когда общее число программистов – 18. $18 : 3 = 6$. Значит, среди программистов 6 – 3D-дизайнеры. Однако известно, что 3D-дизайнеров всего 5 (противоречие).

Ответ: В компании ALT-Future 15 человек являются программистами.

Система оценки

1. Правильно определено количество «чистых» программистов (6 баллов).
2. Правильно определены варианты возможного общего числа всех программистов компании (8 баллов).
3. Получен правильный ответ (10 баллов).

Задача 6.1.2. Количество посетителей компании AltFuture (15 баллов)

Рядом с офисом компании AltFuture расположено несколько видеокамер, посредством которых сотрудники компании отслеживают количество своих посетителей. Записи с камеры №1 показали, что с понедельника до субботы включительно офис компании посетили 510 человек. Камера №2 зафиксировала 392 посетителя с понедельника по среду включительно, а по камере №3, работавшей во вторник и пятницу, насчитали 220 человек. Четвертая камера была включена в среду, четверг и субботу, ее показания составили 208 человек. Наконец, обработав показания камеры №5, удалось определить, что с четверга по субботу включительно за услугами компании обратилось 118 человек. Обработка данных со всех камер ведется автоматически и выдается в конце недели суммарным показателем за те дни, когда камеры работали. Определите, сколько посетителей побывало в офисе компании AltFuture в понедельник?

Решение

Способ 1 Визуализируем условия задачи в виде таблицы посещаемости офиса компании:

-	Пн	Вт	Ср	Чт	Пт	Сб	Всего
1 камера	+	+	+	+	+	+	510
2 камера	+	+	+	-	-	-	392
3 камера	-	+	-	-	+	-	220
4 камера	-	-	+	+	-	+	208
5 камера	-	-	-	+	+	+	118

За исключение понедельника каждый день упоминается 3 раза. Это приводит к двойному учету посетителей четырьмя последними камерами во все дни кроме понедельника. Таким образом, искомое количество посетителей в понедельник можно найти из следующего выражения:

$$2 \cdot 510 - (392 + 220 + 208 + 118) = 1020 - 938 = 82$$

Способ 2

Составим систему уравнений:

$$\begin{cases} \text{ПН} + \text{ВТ} + \text{СР} + \text{ЧТ} + \text{ПТ} + \text{СБ} = 510, \\ \text{ПН} + \text{ВТ} + \text{СР} = 392, \\ \text{ВТ} + \text{ПТ} = 220, \\ \text{СР} + \text{ЧТ} + \text{СБ} = 208. \\ \text{ЧТ} + \text{ПТ} + \text{СБ} = 118. \end{cases}$$

Решив систему уравнений, можно попытаться найти ответ. Однако, следует заметить, решение системы значительно упрощается, если обратить внимание на тот факт, что ответ можно получить, вычитая уравнения (3) и (4) из уравнения (1), т.е. $\text{ПН} = 82$.

Ответ: В офисе компании ALT-Future в понедельник побывало 82 человека.

Система оценки

1. Правильно составлена система уравнений или есть таблица учета наблюдений посещаемости офиса компании (5 баллов).
2. Получено правильное решение (10 баллов).

Задача 6.1.3. Работа для стажеров (15 баллов)

3D-дизайнер, работающий в компании ALT-Future, делает за 1 час целое число моделей для проекта, больше 5, а стажер – на 2 модели меньше. Один 3D-дизайнер выполняет техническое задание по проекту за целое число часов, а два стажера вместе – на 1 час быстрее. Какое количество 3D-моделей включено в техническое задание по проекту?

Решение

Пусть $x > 5$ моделей делает 3D-дизайнер за 1 час, тогда стажер за один час делает $x - 2$ трехмерные модели. Пусть также 3D-дизайнер выполняет заказ за t часов, где t – целое число. Составим уравнение:

$$xt = 2(x - 2)(t - 1)$$

$$t = \frac{2x-4}{x-4}$$

$$t = 2 + \frac{4}{x-4}$$

Дробь $4/(x-4)$ должна быть целым числом. При $x > 5$ это возможно, когда $x = 6$ или $x = 8$. В первом случае получаем, что $t = 4$, во втором – $t = 3$. В обоих случаях техническое задание проекта содержит заказ на изготовление $xt = 24$ трехмерные модели.

Ответ: Техническое задание, поступившее в компанию ALT-Future, содержало заказ на разработку 24 трехмерных моделей.

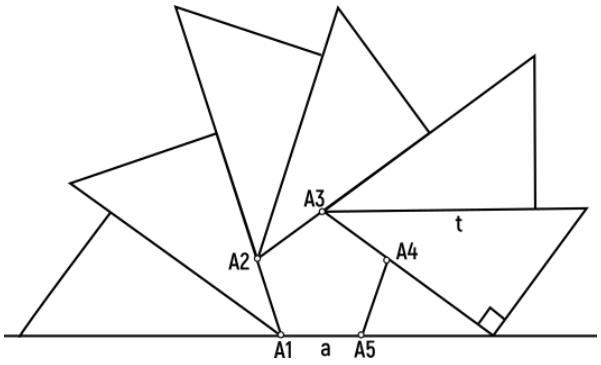
Система оценки

1. Правильно составлено уравнение к задаче (2 балла).
2. Определены все возможные значения времени выполнения заказа 3D-дизайнером (10 баллов).
3. Правильно определено количество 3D-моделей, включенное в техническое задание по проекту (3 балла)

Задача 6.1.4. Портал дополненной реальности (26 баллов)

Компания AltFuture взялась за заказ по созданию портала дополненной реальности для нового торгового комплекса.

В техническом задании, которое принес заказчик, приведен макет портала и рисунок ожидаемого результата:



а) макет

б) ожидаемый результат

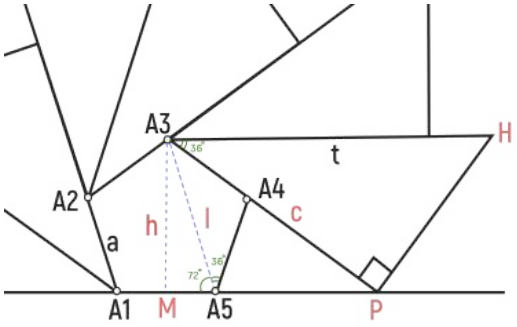
Ворота портала должны быть выполнены ввиде правильного пятиугольника ($A1$, $A2$, $A3$, $A4$, $A1$), из сторон которого выходят “листья” ввиде конгруэнтных прямоугольных треугольников.

Важно, чтобы результат точно отображал идею заказчика, поэтому листья не должны заходить за пределы уровня земли.

Составьте выражения для определения размера гипотенузы треугольника - “листа”,образующего вход в портал.

Решение

Обозначим отрезок $MA3$ за h. Отрезок $A3A5$ за l и отрезок $A3P$ за . Отрезок $A3H$ за t, где t - искомый катет.



Из треугольника $A3A4A5$:

$$l = 2a \cdot cos36^\circ$$

Из треугольника $A3MA5$:

$$h = l \cdot sin72^\circ \Rightarrow h = 2a \cdot cos36^\circ \cdot sin72^\circ$$

Из треугольника $A3MP$:

$$cos54\circ = \frac{a}{c} \Rightarrow c = \frac{h}{cos54^\circ} \Rightarrow c = \frac{2 \cdot a \cdot cos36^\circ \cdot sin72^\circ}{cos54^\circ}$$

Из треугольника $A3PH$:

$$cos36^\circ = \frac{c}{t}$$

$$\Rightarrow t = \frac{c}{cos36^\circ}$$

$$\Rightarrow t = \frac{2 \cdot a \cdot cos36^\circ \cdot sin72^\circ}{cos36^\circ \cdot cos54^\circ}$$

$$\Rightarrow t = \frac{4 \cdot a \cdot cos36^\circ \cdot sin36^\circ}{cos54^\circ}$$

$$\Rightarrow t = \frac{4 \cdot a \cdot cos36^\circ \cdot cos54^\circ}{cos54^\circ}$$

$$\Rightarrow t = 4 \cdot a \cdot cos36^\circ$$

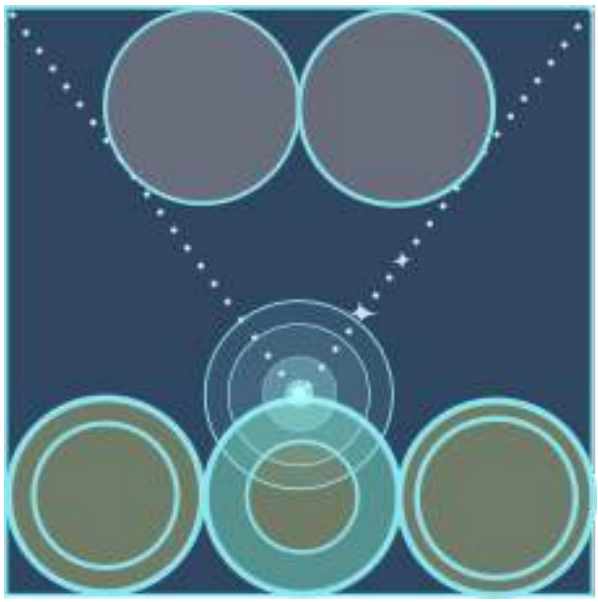
Ответ: Размер катета "листа"должен быть равен $4 \cdot a \cdot cos36^\circ$

Система оценки

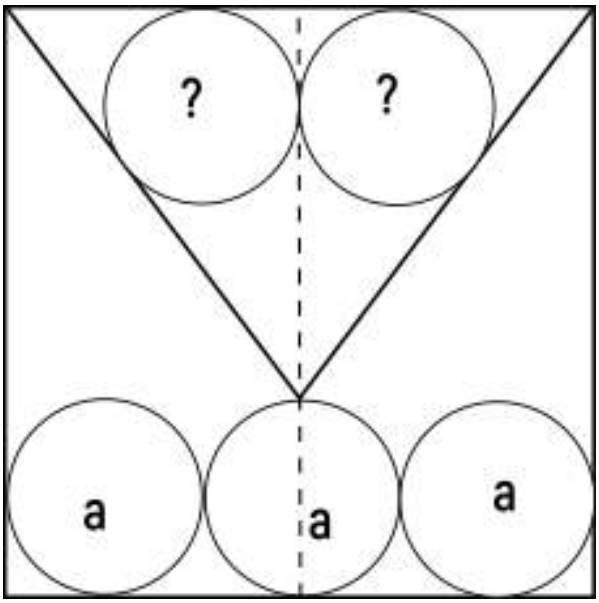
1. Правильно выполнено построение к задаче (6 баллов).
2. Составлено верное выражение для определения катета C (образующего вход в портал) одного из треугольников - «листьев» портала (10 баллов).
3. Правильно составлено выражение для определения размера гипотенузытреугольника-"листа»,образующегопортал(10 баллов).

Задача 6.1.5. Музыкальный AR-интерфейс (20 баллов)

Команда AltFuture разрабатывает вместе с коллегами из Японии музыкальный интерфейс в дополненной реальности для разного вида шоу и выставок. На Андрея, ведущего 3D-дизайнера компании, возложили проектировку трехмерных моделей музыкальных инструментов для AR-интерфейса. В ходе работы над проектом, Андрей заметил, что на чертеже одного из музыкальных инструментов, коллегами из Японии указаны размры не всех объектов. На рисунке показан эскиз модели, присланный иностранными специалистами.



а) Ожидаемый результат

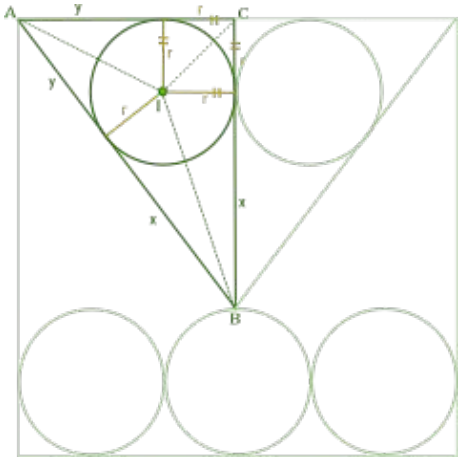


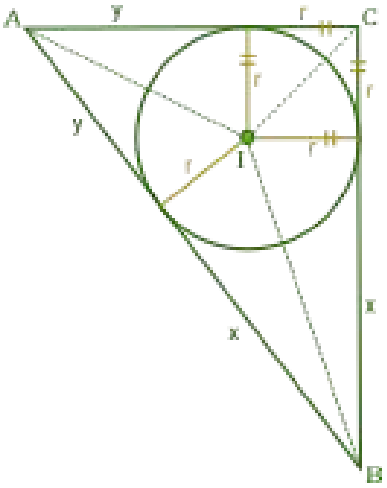
б) Макет

На чертеже изображен квадрат, на нижней стороне которого располагаются 3 круга одного размера. Треугольником обозначено, как будет идти волна распространения звука в игре. Видно, что одно из соударений волны проходит через центр квадрата, задевая среднюю окружность. А вот размер двух окружностей, который оказались вписанными в треугольник остались неизвестны. Андрей предположил, что размеры всех 5 окружностей, равны и решил делать модель с расчетом на это. Докожите, что предположение Андрея верно.

Решение

Рассмотрим треугольник ABC.





Если считать, что все стороны квадрата раны 6, то треугольник будет иметь стороны 3, 4, 5. Радиус трех нижних окружностей равен 1, тогда нужно доказать, что радиус вписанной в ABC окружности так же равен 1: Рассмотрим треугольник IBC. У него основание BC и высота r, $\Rightarrow Sibc = \frac{BC \cdot r}{2}$ Точно так же в треугольниках ICA И IAB площади будут равны:

$$\frac{CA \cdot r}{2}$$

$$\frac{AB \cdot r}{2};$$

Учитывая, что площадь треугольника

$$ABC = \frac{AC \cdot CB}{2},$$

получаем следующее равенство:

$$\frac{AC \cdot AB}{2} = \frac{BC \cdot r}{2} + \frac{CA \cdot r}{2} + \frac{AB \cdot r}{2}; \Rightarrow$$

$$AC \cdot AB = (BC + CA + AB) \cdot r;$$

Подставляя $BC = 3$, $CA = 4$, $AB = 5$ получаем $r = 1$.

Что и требовалось доказать.

Система оценки

1. Правильно выполнено построение к задаче (8 баллов).
2. Приведено верное доказательство(12 баллов).

6.2. Дополненная реальность. Математика. 10-11 класс

Задача 6.2.1. (35 баллов)

Мячик радиуса 9.5 см вылетает из некой точки пространства (x_0, y_0, z_0), $z_0 > 0$, со скоростью $(0, 0, -4.85)$ см/с. В точке $(0, 0, 0)$ находится камера и снимает движения мячика, луч зрения камеры совпадает с положительным направлением оси Z. В момент времени $t_1 = 5$ с угловой размер (угловой диаметр) мячика относительно камеры равен $T_1 = 11°50'14.309''$, в момент $t_2 = 11.25$ с $-T_2 = 17°14'4.454''$. Угловой размер можно найти из треугольника составленного из расстояния до мячика и его радиуса. Камера находится в корпусе диаметром $D = 25.5$ см.

На какое расстояние перпендикулярно оси Z нужно подвинуть мячик, чтобы он только коснулся корпуса камеры.

Смещение к оси Z считать отрицательным, от оси Z положительным. Ответ округлите до сотых долей сантиметра.

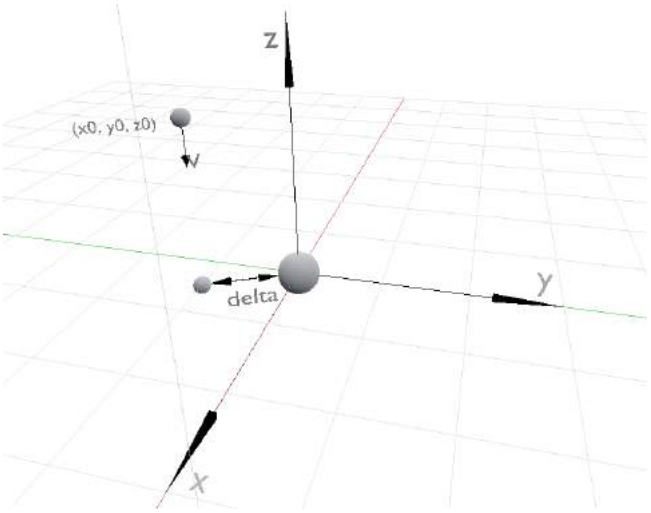
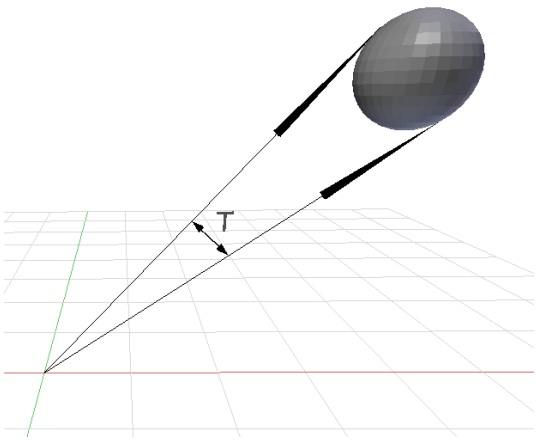


Иллюстрация углового размера:



Решение

Движение мячика происходит только в направлении Z, поэтому точкой пересечения будет $(x_0, y_0, 0)$. Условием касания мячика и камеры будет:

$$R + D/2 == \sqrt{x_0^2 + y_0^2}$$

Таким образом задача сводится к двум неизвестным. Искомая величина смещения выражается как:

$$\delta = (R + D/2) - \sqrt{x_0^2 + y_0^2}$$

Угловой размер (угловой диаметр) равен $\theta = 2 \cdot arctg(R/L)$, L это расстояние от камеры до мячика, R - радиус мячика. Выразим L можно через угловой размер:

$$L = \frac{R}{\operatorname{tg} \frac{\theta}{2}}$$

С другой стороны L:

$$L^2 = x_0^2 + y_0^2 + z^2$$
$$L^2 = x_0^2 + y_0^2 + (z_0 + v_z \cdot t)^2$$

В момент времен t_1 и t_2:

$$L_1^2 = x_0^2 + y_0^2 + (z_0 + v_z \cdot t_1)^2$$
$$L_2^2 = x_0^2 + y_0^2 + (z_0 + v_z \cdot t_2)^2$$

Вычтем одно из другого и выразим z_0:

$$z_0 = \frac{L_1^2 - L_2^2 - v_z^2 \cdot (t_1^2 - t_2^2)}{2 \cdot v_z \cdot (t_1 - t_2)}$$

Из любого уравнения выше можем выразить сумму квадратов x_0 и y_0:

$$x_0^2 + y_0^2 = L_1^2 - (z_0 + v_z \cdot t_1)^2$$

Значения L можно вычислить из выражений выше.

Подставив выражение выше найдем что:

$$z_0 = 113.1$$
$$\sqrt{x_0^2 + y_0^2} = 22.4294109...$$

Найдем смещение необходимое для касания.

$$\delta = (9.5 + 12.75) - 22.4294109... = -0.17941... = -0.18$$

Ответ: -0.18.

Система оценки

1. Определено условия касания (5 баллов)
2. Определена связь координат мячика и углового размера (10 баллов)
3. Величина смещения определена правильно (20 баллов)

Задача 6.2.2. (25 баллов)

Существуют три кольцевых маршрута автобусов №1, №2, №3. Первый пересекается со вторым на одной остановке Q, второй пересекается с третим на одной остановке S. Первый и третий маршруты не пересекаются. Автобусы курсируют по часовой стрелке. Время полного круга каждого из маршрутов: $L_1 = 59$, $L_2 = 37$ и $L_3 = 89$ минут соответственно. Автобусы №1, №2 начинают движение одновременно в момент времени $t = 0$ на остановке Q.

Маршрут №3 начинает движение с остановки Z, когда маршрут №2 проходит остановку S.

Пассажиру нужно доехать с остановки A маршрута №1, от которой до остановки Q ехать $\delta_{AQ} = 23$ минуты. От S до Z $\delta_{SZ} = 41$ минуты. Время движения от Q до S равно $\delta_{QS} = 19$ минуты.

Найти первое возможное время посадки на останоке A так, что время в пути до Z минимально. Время считать в минутах от начала движения маршрутов №1 и №2, автобусы курсируют бесконечно. Пересадки считать моментальными, пересадка возможна если автобусы оказываются на одной остановке в одно и то же время с точностью до минуты и во все последующие моменты времени.

Решение

Минимальное возможное время без учета ожидания следующего маршрута равно:

$$T = \delta_{AQ} + \delta_{QS} + \delta_{SZ}$$

Такое время поездки достигается если маршруты №1 и №2 встретяться на остановке Q и на этом же круге №2 встретится с №3 на S. Обозначим время посадки на A как t, к моменту времени $t + 23$ маршруты №1 и №2 должны совершить полный круг каждый. Поэтому остаток от деление $t + 23$ на 59 и 37 равно нулю:

$$(t + 23)\%59 = 0$$
$$(t + 23)\%37 = 0$$

Так как 59 и 37 простые числа искомое t должно удовлетворять следующему выражению:

$$t = 59 \cdot 37 \cdot n - 23$$

где n - целое положительное число.

Так же известно что в момент времени $t + 23 + 19 + 41$ пассажир должен быть на остановке Z. Так как автобус №3 начинает курсировать с Z то это время так же должно соответствовать целому числу кругов:

$$(t + 23 + 19 + 41)\%89 = 0$$

$$t = 89 \cdot m - 23 - 19 - 41$$
$$59 \cdot 37 \cdot n - 23 = 89 \cdot m - 23 - 19 - 41$$

$$59 \cdot 37 \cdot n = 89 \cdot m - 60$$

$$m = (59 \cdot 37 \cdot n + 60)/89$$

Подбираем $n = 65$ что соответствует моменту времени $t = 141872$.

Ответ: 141872.

Система оценки

1. Определено минимальное время поездки (5 баллов)
2. Определено время для пересадки без ожидания с маршрута 1 на 2 (5 баллов)
3. Определено соотношение (без количественного значения) для пересадки без ожидания с маршрута 1 на 2 (5 баллов)
4. Время выхода на остановку определено верно (10 баллов)

Задача 6.2.3. (10 баллов)

Есть бесконечная решетка с квадратными ячейками, ширина прутьев 10 см, размер ячеек таков что куб со стороной 20 см проходит ячейку без зазоров. Решетка расположена горизонтально. Сверху вертикально вниз падают мячики, диаметр которых 15 см. количество мячиков 100000. Координаты мячиков выбираются случайным образом. Мячики, которые задевают решетку исчезают. Мячики выпускаются таким образом, что соударений между ними нет. Найти количество целых мячиков. Ответ округлить до целых.

Решение

Шарик пройдет решетку только в случае если его координаты его центра находятся в диапазоне $(\pm 2.5, \pm 2.5)$ см от центра ячейки. Т.е. **окно безопасности** равно $5 \cdot 5 = 25$ кв. см. Так как у нас есть N ячеек, есть N **окон безопасности**. Вероятность шарика попасть в любое окно пропорционально отношению площадей **окна безопасности** и площади ячейки с прутьями. Так как ширина прутьев 10 см, можно считать, что к каждой ячейке добавляются по 5 см с каждой стороны. Поэтому площадь ячейки с прутьями $30 \cdot 30 = 900$ см^2.

$$p = N \cdot 25/N \cdot 900 = 25/900 = 0.027778$$

$$N = p \cdot 100000 = 2777.8 = 2778$$

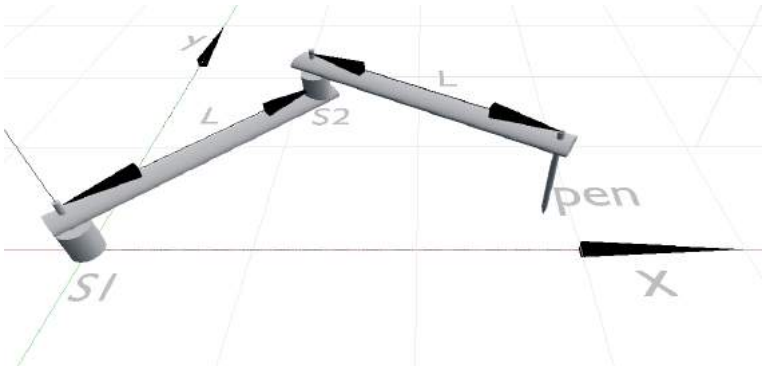
Ответ: 2778.

Система оценки

1. Дано соотношение вероятности мячика попасть (или миновать) в решетку (5 баллов)
2. Дан верный ответ (5 баллов)

Задача 6.2.4. (30 баллов)

Манипулятор состоит из двух плечей эффективной длины L. Основание первого плеча установлено на валу сервопровода S1, закрепленного в начале координат. Сервопривод S2 закреплен на конце первого плеча, а к его валу прикреплено основание второго плеча. К концу второго плеча прикреплен карандаш (см. рисунок). Сервоприводы могут поворачиваться в диапазоне ± 90 градусов. При положение сервоприводов (0°, 0°) оба плеча расположены вдоль положительного направления оси X. Если смотреть с положительного направления Z - увеличение угла поворота сервопривода, соответствует движению плеча, в основании которого установлен сервопривод, по часовой стрелке. Уменьшение - к движению против часовой стрелки.



Найти выражение параметрического задания углов поворота сервоприводов $s_1 = f(t), s_2 = g(t)$ для рисовки части оси X, которую позволяет отрисовать конструктивные ограничения манипулятора. Так же дайте диапазон изменения параметра t с точностью до градуса. Линия должна быть орисована в положительном направлении оси X.

Решение

Из-за конструктивных особенностей данный манипулятор может отрисовать только часть оси X от $(\sqrt{2}l, 0)$ до $(2l, 0)$. Плечи манипулятора с осью X формируют равнобедренный треугольник. Если угол между первым манипулятором и осью X равен $s_1 = \alpha$, то угол между вторым манипулятором и осью X так же должен быть α. Угол лежаший напротив основания равен $180 - 2\alpha$. Следовательно модуль угла поворота второго сервопривода равен $s_2 = 180 - (180 - 2\alpha) = 2\alpha$, направление противоположно повороту S1, поэтому $s_2 = -2\alpha$. Таким образом углы задаются как:

$$s1 = t; s2 = -2t$$

Когда карандаш находится в положении $(\sqrt{2}l, 0)$ это соответствует равнобедренному прямоугольному треугольнику $\alpha = 45°$. Так как относительно оси X плечо должно отклониться против часовой стрелки то угол поворота $S1$ равен -45°. При положении $(2l, 0)$, как следует из условия, оба сервопривод повернуты на 0°.

Ответ: $s1 = \pm t; s2 = \mp 2t;\ t = -45° \ldots 0°$.

Система оценки

1. Определено, что сервоприводы должны вращаться в противоположные стороны (10 баллов)
2. Правильно определены координаты начала и конца линии (5 баллов)
3. Дан вывод соотношений углов поворота сервопривода (5 баллов)
4. Дан верный ответ (10 баллов)

6.3. Дополненная реальность. Информатика

Задача 6.3.1. Задача 1 (15 баллов)

Космический корабль путешествует от одной планеты к другой планете. Необходимо определить за какое наименьшее количество прыжков корабль достигнет точки назначения. Корабль может прыгать только к ближайшей планете и не может посещать уже ранее посещенные планеты.

Формат входных данных

В первой строке задано количество звезд $n(2 \leqslant\ n \leqslant 100)$. Вторая строка содержит номера стартовой и конечной планет разделенные пробелом, нумерация начинается с 0. Следующие строки содержат целочисленные координаты планет в трехмерном пространстве $x, y,\ z(-100 \leqslant x, y, z \leqslant 100)$ разделенные пробелом.

Формат выходных данных

В отдельной строке единственное целое число — ответ на задачу.

Пример №1

Стандартный ввод
111 0 5 0 0 0 2 3 0 4 1 2 -2 3 0 7 1 3 6 5 4 3 2 7 1 1 1 6 8 -8 1 -1 -1 2 -5 2

Стандартный вывод
5

Решение

Считаем координаты планет в трёхмерном пространстве в массив. Нам также потребуется вспомогательная функция, рассчитывающая расстояние между точками в пространстве (по теореме Пифагора) и массив для учёта уже посещённых планет. Начинаем со стартовой планеты и двигаемся к ближайшей (определяя расстояния до всех не посещённых ранее планет), считая число перемещений до тех пор, пока не достигнем точки назначения. Важно учесть, что стартовая и конечная точка могут совпадать.

Пример программы-решения

Ниже представлено решение на языке Python3

```
def distance(planet1, planet2):
    d = [(i - j) ** 2 for i, j in zip(planet1, planet2)]
    return sum(d) ** 0.5


def find_nearest(planet, planets, visited):
    min_distance = 10 ** 6
    nearest = None
    for i, p in enumerate(planets):
        if visited[i]:
            continue
        d = distance(planet, p)
        if d < min_distance:
            min_distance = d
            nearest = i
    visited[nearest] = True
    return nearest


n = int(input())
s, e = list(map(int, input().split()))

planets = []
for _ in range(n):
    planets.append(list(map(int, input().split())))

visited = [False] * n
visited[s] = True

steps = 0
while s != e:
    s = find_nearest(planets[s], planets, visited)
    steps += 1

print(steps)
```

Задача 6.3.2. Задача 2 (20 баллов)

Дана матрица состоящая из 0 и 1. Значением 1 в матрице нарисованы прямоугольники. Необходимо определить площадь каждого отдельного прямоугольника и вывести в порядке возрастания.

Формат входных данных

В первой строке через пробел заданы $h, w (2 \leqslant h, w \leqslant 50)$ — высота и ширина матрицы соответственно. В следующих строках заданы значения матрицы по строкам и столбцам. Прямоугольники отделены друг от друга и от краев матрицы как минимум через один ноль, включая диагонали. Минимальный возможный размер прямоугольника 2 на 2 значения.

Формат выходных данных

В каждой строке целочисленное значение площади прямоугольника. Значения площадей упорядочены по неубыванию.

Пример №1

Стандартный ввод
10 20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Стандартный вывод
8 9 25

Решение

Одно из простых в реализации решений состоит из двух проходов. Сначала, проходом всего поля маркируем ячейки каждого прямоугольника своим номером (первый прямоугольник - числом 2, второй — числом 3 и т.д.). Обход можно совершать как циклом (быстрее), так и рекурсией (короче). Вторым проходом считаем число промаркированных ячеек каждого типа.

Пример программы-решения

Ниже представлено решение на языке Python3

```
# put your python code here

def neighbours(i, j):
    return [i-1, j], [i+1, j], [i, j-1], [i, j+1]


```

```
def label_area(m, i, j, label):
    m[i][j] = label
    for ni, nj in neighbours(i, j):
        if m[ni][nj] == 1:
            label_area(m, ni, nj, label)

h, w = map(int, input().split())
m = []

for i in range(h):
    line = map(int, input().split())
    m.append(list(line))

label = 2
for i in range(h):
    for j in range(w):
        if m[i][j] == 1:
            label_area(m, i, j, label)
            label += 1

areas = {}

for i in range(h):
    for j in range(w):
        value = m[i][j]
        if value != 0:
            if value not in areas:
                areas[value] = 1
            else:
                areas[value] += 1

for area in sorted(areas.values()):
    print(area)
```

Задача 6.3.3. Задача 3 (30 баллов)

В заданной матрице определить максимально длинную последовательность из значений, которые отличаются друг от друга на 1, вывести полученную последовательность в порядке возрастания. Последовательность должна состоять из чисел, которые прилегают друг к другу, т.е. между числами нет значения 0. Минимальное число от которого может начинаться последовательность $\geqslant$. Значение 0 не участвует в составлении последовательности. У каждого значения в последовательности может быть не больше двух соседних значений включая диагонали. Если найдено несколько последовательностей одинаковой длины, то вывести последовательность с максимальной суммой.

Формат входных данных

В первой строке через пробел заданы h, w $(10 \leqslant h, w \leqslant 50)$ — высота и ширина матрицы соответственно. В следующих строках заданы значения матрицы по строкам и столбцам.

Формат выходных данных

Последовтельность целых чисел, каждое число в новой строке — ответ на вопрос задачи.

Пример №1

Стандартный ввод
10 20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 0 0 0 0 9 8 7 6 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0 0 0 0 3 0 0 0 0 5 0 0 0 0 11 0 0 0 0 1 0 0 0 0 4 0 0 0 0 6 0 0 0 0 12 0 0 0 0 2 0 0 0 0 5 0 0 0 0 7 0 0 0 0 13 14 15 0 0 3 0 0 0 0 6 0 0 0 0 8 0 0 0 0 0 0 0 0 0 4 0 0 0 0 7 0 0 0 0 9 10 11 12 13 14 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Стандартный вывод
6 7 8 9 10 11 12 13 14 15

Решение

Первым шагом в решении после считывания матрицы может быть выборка последовательностей из неё. Для этого подойдёт рекурсивный обход или вложенный цикл. Для избежания дублирования выборки можно отмечать уже обработанные ячейки в отдельном массиве. Далее отбираем для каждой последовательности подпоследовательность — числа, отличающиеся только на единицу друг от друга, например, 3, 4, 5, 6. Отобранные последовательности ранжируем (сортируем) по сумме элементов.

Пример программы-решения

Ниже представлено решение на языке Python3

```
# put your python code here


import numpy as np

def neighbours(i, j):
    return [i-1, j], [i+1, j], [i, j-1], [i, j+1]

def read_sequence(m, i, j, sequence):
    sequence.append(m[i][j])
    m[i][j] = 0
    for ni, nj in neighbours(i, j):
        if m[ni][nj] != 0:
            read_sequence(m, ni, nj, sequence)
```

```
15
16  h, w = map(int, input().split())
17  m = []
18
19  for i in range(h):
20      line = map(int, input().split())
21      m.append(list(line))
22
23  sequences = []
24  for i in range(h):
25      for j in range(w):
26          if m[i][j] != 0:
27              sequence = []
28              read_sequence(m, i, j, sequence)
29              sequences.append(sorted(sequence))
30
31  max_len = 0
32  index = 0
33  for i, sequence in enumerate(sequences):
34      ln = len(sequence)
35      if ln == max_len:
36          if sum(sequence) > sum(sequences[index]):
37              max_len = 0
38      if ln > max_len:
39          max_len = ln
40          index = i
41
42  for value in sequences[index]:
43      print(value)
```

Задача 6.3.4. Задача 4 (35 баллов)

С расстояния 100 метров в стену с отверстиями разного диаметра последовательно бросают мячи, которые имеют разный радиус. Мячи всегда бросают из одной позиции. Необходимо определить количество шаров пролетевших сквозь стену. При этом движение мяча рассматривается как линейное, т.е. без учета гравитации. Для каждого шара задан угловой размер и координаты на расстоянии в 50 метров после броска. Пролет мяча считается только при чистом пролете мяча через отверстие, т.е. если мяч не задел край.

Формат входных данных

В первой строке задается целое число $n(1 \leqslant n \leqslant 100)$ количество брошеных шаров. Во второй строке задано целое число $h(1 \leqslant h \leqslant 100)$ — количество отверстий в стене. В третьей строке через пробел заданы координаты точки, из которой бросают шары (x, y, z). В следующих h строках заданы координаты отверстий в стене и диаметр отверстий через пробелы (x, y, z, d). За ними следуют координаты мячей и угловой размер на 50 метрах в минутах $(x_{50}, y_{50}, z_{50}, s)$. Координаты, размеры мячей и размеры отверстий представлены целыми числами. Угловой размер задается в минутах.

Формат выходных данных

Единственное целое число — количество шаров, пролетевших сквозь стену.

Пример №1

Стандартный ввод
14 6 100 10 10 0 10 10 4 0 17 10 1 0 4 4 2 0 2 12 2 0 20 14 3 0 24 5 2 50 10 10 120 50 10 10 350 50 9 10 100 50 9 9 150 50 24 4 1 50 13 7 100 50 8 7 200 50 7 7 10 50 6 7 100 50 12 3 200 50 11 5 100 50 20 14 4 50 2 3 120 50 23 3 1
Стандартный вывод
3

Решение

Линейный размер мяча определяется $size = 2 \cdot L \cdot \sin(D/2)$, где L расстояние до объекта, D — угловой размер. Координаты попадания мяча в стену определяются исходя из того, что нормаль из начальной точки и прямая проходящая через начальную точку и точку на 50 метрах образуют подобные треугольники по двум углам, при заданных условиях коэффициент равен 2. Таким образом координаты ударения мяча в стену можно определить по удвоенному смещению начальной точки от точки на 50 метрах прибавленному к координатам начальной точки. Определение пролета мяча сквозь стену определяется следующим образом: $r_1 > (d + r_2)$

Пример программы-решения

Ниже представлено решение на языке Python3

```
import math

def distance(p1, p2):
    d = [(i - j) ** 2 for i, j in zip(p1, p2)]
    return sum(d) ** 0.5

def hit_point(p1, p2):
    hit_x = 0
```

```
    hit_y = p1[1] + (p2[1] - p1[1]) * 2
    hit_z = p1[2] + (p2[2] - p1[2]) * 2
    return hit_x, hit_y, hit_z

def linear_size(angular_size, distance):
    angular_size = math.radians(angular_size)
    return 2 * distance * math.sin(angular_size / 2)


n = int(input())
h = int(input())

start_point = list(map(int, input().split()))

holes = []

for _ in range(h):
    holes.append(list(map(int, input().split())))

hits = []
for _ in range(n):
    ball = list(map(int, input().split()))
    size = linear_size(ball[-1] / 60, 50)
    hits.append([hit_point(ball[:3], start_point), size])


passed = 0
for hit in hits:
    for hole in holes:
        d = distance(hole[:3], hit[0])
        if hole[-1] > (d + hit[1]):
            passed += 1

print(passed)
```

6.4. Виртуальная реальность. Математика. 9 класс

Задача 6.4.1. (15 баллов)

Найдите общий корень уравнений

$$x^2 - x - 42 = 0,$$

$$3x^3 - 16x^2 - 33x - 14 = 0.$$

Решение

Решив первое уравнение каким-либо известным способом находим его корни

$$x_1 = 7, x_2 = -6.$$

Аналогично находим корни второго уравнения:

$$x_1 = -1, x_2 = 7, x_3 = -\frac{2}{3}.$$

Общим корнем будет $x = 7$.

Ответ: 7.

Критерии оценки

+4 баллов за нахождение корней первого уравнения;

+9 баллов за нахождение корней второго уравнения;

+2 балла за выбор общего корня.

Замечания

1. За нахождения общего корня перебором баллы снижаются только в случае, когда не была проведена проверка того, что найденное число является корнем обоих уравнений.
2. Если найденный общий корень не совпадает с ответом, то за задачу ставится 0 баллов.
3. Если при вычислении корней обоих уравнений были совершены арифметические ошибки, то баллы за каждое из этих уравнений ставятся пропорционально числу правильно найденных корней.

Задача 6.4.2. (15 баллов)

Найдите все простые числа p, для которых $2p^4 + 46$ — квадрат целого числа.

Решение

Найдем для начала хотя бы одно решение. Для этого переберем малые простые числа и убедимся, что $p = 5$ подходит.

Покажем, что других решений у уравнения нет. Для этого заметим, что если $p \neq 5$, то оно взаимно просто с пятью, а значит p^4 дает остаток 1 при делении на 5.

Тогда выражение $2p^4 + 46$ дает остаток 3 при делении на 5. Легко убедиться, что квадраты целых чисел могут давать только остатки $0, 1, 4$. Следовательно, больше нет решений в целых числах.

Ответ: $p = 5$.

Критерии оценки

+5 баллов за подобранный пример;

+10 баллов за доказательство единственности.

Задача 6.4.3. (20 баллов)

Дана трапеция с основаниями $AB = 18\sqrt{3}$ и $CD = 12\sqrt{3}$. На AB и CD взяты точки M и N соответственно так, что $MN = 6$.

а) *(5 баллов)* Найдите площадь трапеции, если дополнительно известно, что угол между MN и AB равен $60°$.

б) *(15 баллов)* Найдите наибольшее возможное значение наименьшего из углов трапеции, если угол A — прямой.

Решение

а) Заметим, что высота в такой трапеции будет равна $h = \sin 60° \cdot 6 = 3\sqrt{3}$. Тогда по формуле площади трапеции $S = \frac{1}{2}h(a+b) = 135$.

б) Так как $AB > CD$, то острый угол B — наименьший из углов трапеции. Опустим из вершины C перпендикуляр CE на AB. Его длина не больше длины $MN = 6$. Кроме того, из условия следует, что

$$BE = 18\sqrt{3} - 12\sqrt{3} = 6\sqrt{3}.$$

Поэтому $\operatorname{tg}\angle B \leqslant \dfrac{1}{\sqrt{3}}$, откуда $\angle B \leqslant 30°$.

Для того чтобы получить максимальное значение угла B, достаточно предположить, что MN — высота трапеции.

Ответ: а) 135.

Критерии оценки

а) *(5 баллов)*

+5 баллов за правильно найденную площадь.

б) *(15 баллов)*

+10 баллов за доказанную оценку;

+5 баллов за правильно подобранный пример с углами $30°$ и $150°$.

Задача 6.4.4. (25 баллов)

На аукционе Илья соревнуется с Петей за ценную картину. Изначальная ее цена — 1000 рублей. За раз стоимость лота можно поднять на 2000 или 5000 рублей. Парни так сильно жаждут получить картину, что не уйдут, пока она не достанется одному из них. Аукционист сообщил, что если цена за картину поднимется выше 28200 рублей, то последний поднявший цену получит картину. Первым цену поднял Илья... (на сколько — вы не знаете)

а) *(10 баллов)* Правильно ли он поступил? Сможет ли он при любом поведении его соперника, получить заветный лот?

б) *(15 баллов)* Изменится ли ответ, если вместо 2000 или 5000 рублей ребята могут поднимать цену на 20 или 50?

Решение

а) Покажем, что Петя может всегда выиграть, как бы Илья не ходил. Для этого Петя может всегда ходить противоположнвм образом, то есть если Илья поднял цену на 2000 рублей, то Петя поднимет на 5000 рублей и наоборот. Тогда

после 4 таких ходов стоимость станет равной 29000 рублей и Петя заберет картину.

б) Рассмотрим игру с конца с позиции выигрышных и проигрышных позиций. Так как максимальная цена — 28200 рублей, то 28200 — выигрышная, 28190 — выигрышная, ..., 28160 — выигрышная, 28150 — проигрышная, 28140 — проигрышная, 28130 — выигрышная, 28120 — выигрышная, 28110 — проигрышная, 28100 — выигрышная, 28090 — выигрышная, 28080 — проигрышная, 28070 — проигрышная, ...

Можно заметить, что в данном случае будут повторяться постоянно 7 позиций:

...П, П, В, В, П, В, В...

Итак, получим, что 1050 и 1020 — это проигрышные позиции, поэтому как бы первый раз не сходил Илья он может потом продолжить играть так, чтобы выиграть.

Ответ: а) Нет, у Пети есть выигрышная стратегия; б) Да, у Ильи есть выигрышная стратегия.

Критерии оценки

а) *(10 баллов)*

+5 баллов за нахождение стратегии для Пети;

+5 баллов за рассуждения о том, что Илья проиграет, независимо от своей игры.

б) *(15 баллов)*

+10 баллов за доказательство того факта, что позиции повторяются;

+5 баллов за рассуждения о том, что Илья выиграет, независимо от игры оппонента.

Замечания

1. Если без каких-либо объяснений сказано, что позиции повторяются, то ставится 5 вместо 10 баллов.
2. Если правильно найден вид периода позиций, но нет рассуждений, почему, если рассматривать с конца, будут повторяться именно эти 7 позиций, то ставится аналогично 5 вместо 10 баллов.
3. Если все рассуждения о периодичности сделаны верно, но для позиций 1050 и 1020 сказано, что хотя бы одна выигрышная, то снимается все 5 баллов.

Задача 6.4.5. (25 баллов)

Вася случайным образом поставил на доску

а) *(10 баллов)* двух королей;

б) *(15 баллов)* короля, ладью и ферзя;

С какой вероятностью они будут бить друг друга (во втором пункте подразумевается, что каждые две фигуры будут бить друг друга)?

Замечание: король бьет 8 клеток вокруг себя; ладья бьет те фигуры, которые стоят с ней в одном ряду или столбце (ладья не может перепрыгивать через фигуры); ферзь бьет те фигуры, которые находят с ним в одном ряду или столбце или на одной диагонали (аналогично ферзь не может перепрыгивать через фигуры).

Решение

а) Всего вариантов расставить двух различных королей — $64 \cdot 63 = 4032$. Теперь рассмотрим варианты, когда они бьют друг друга. Таких вариантов $4 \cdot 3 + 24 \cdot 5 + 36 \cdot 8 = 420$. Искомая вероятность будет равна $\frac{420}{4032} = \frac{5}{48}$

б) Эти три фигуры могут попарно друг друга бить только в том случае, если они стоят «уголком», где король и ферзь стоят по диагонали, а ладья соседствует с ними обоими. Тогда заметим, что всего подобных уголков в каждом 2×2 квадратике можно сделать 8 штук, а значит, так как в 8×8 есть 49 различных (они накладываются друг на друга) квадратов 2×2, то всего вариантов, когда все они друг друга бьют — $49 \cdot 8 = 392$.

Следовательно, так как всего $64 \cdot 63 \cdot 62 = 249984$ вариантов, то искомая вероятность равна $\frac{392}{249984} = \frac{7}{4464}$.

Ответ: а) $\frac{5}{48}$; б) $\frac{7}{4464}$.

Критерии оценки

а) *(10 баллов)*

+10 баллов за рассуждения, объясняющие найденную вероятность.

б) *(15 баллов)*

+5 баллов за рассуждения о возможных позициях фигур;

+10 баллов за рассуждения, объясняющие найденную вероятность.

Замечания

1. Если в процессе вычислений вероятностей допущены арифметические ошибки, но все рассуждения правильны, то балл не снижается.

6.5. Виртуальная реальность. Математика. 10-11 класс

Задача 6.5.1. (10 баллов)

Дано 8 чисел, одно из которых целое. Известно, что сумма любых семи из них целая. Могут ли среди чисел быть нецелые?

Решение

Рассмотрим сумму без и с целым числом и вычтем одно из другого. Тогда получим, что $x - a = b$, где a и b — это какие-то целые числа. Тогда x тоже целое число. В силу произвольности x следует, что все восемь чисел целые.

Ответ: Нет, не могут.

Критерии оценки

+10 баллов за рассуждения о том, почему все числа целые.

Задача 6.5.2. (20 баллов)

В двух различных узлах бесконечной сетки сидят заяц и волк. Они оба могут перемещаться только по сторонам клеток. Первым двигается заяц, потом волк, после него заяц... За один свой ход волк преодолевает v сторон клеток, а заяц — z (двигаются они строго по сторонам клеток).

а) *(10 баллов)* Пусть скорости у них равны. Докажите, что заяц может так действовать, чтобы волк его никогда не догнал.

б) *(10 баллов)* Пусть скорость волка $v = z + 1$. Докажите, что при любом расположении персонажей волк сможет догнать зайца.

Решение

а) Пусть заяц всегда будет двигаться от волка. Заметим, что он всегда может двигаться так, чтобы кратчайшее расстояние между ним и волком увеличилось ровно на z. Тогда, как бы волк ни двигался, но после его хода расстояние между ними не сократится более, чем на $v = z$. Поэтому за два хода (зайца и волка) расстояние между ними не уменьшится, а следовательно, если оно изначально было положительным, то и после любого хода будет оставаться таким же.

б) В этом пункте нам надо смотреть на погоню со стороны волка. Он может повторить все движения зайца и при этом еще пройти на одну клетку больше. Тогда пусть он будет последним передвижением сокращать расстояние между ними. При таком поведении расстояние между волком и зайцем будет сокращаться всегда хотя бы на единицу после их двух ходов. Так как в начале игры оно было конечным, то наступит момент, когда волк догонит зайца.

Критерии оценки

а) *(10 баллов)*

+10 баллов за рассуждения о том, почему после двух ходов заяц сможет не уменьшить расстояние между ним и волком.

б) *(10 баллов)*

+10 баллов за рассуждения о том, почему волк может сокращать расстояния между ним и зайцем.

Замечания

1. Может быть приведено более оптимальное решение, где волк не повторяет движения зайца, а просто сокращает расстояния между ними, двигаясь по кратчайшему пути. В этом случае, если не приведено обоснований, почему

после двух ходов расстояние между ними сократится, то ставится половина баллов.

2. За решения без обоснований и пояснений, почему волк догонит или не догонит зайца, оцениваются нулем баллов.

Задача 6.5.3. (20 баллов)

На доске написано k различных ненулевых действительных чисел. Известно, что если выбрать m $(1 \leqslant m \leqslant k)$ произвольных различных чисел с доски и перемножить, то получим число с доски.

а) *(15 баллов)* Найдите максимальное возможное значение k.

б) *(5 баллов)* Дополнительно известно, что все числа по модулю не превосходят 2019. Чему равняется максимально возможная сумма чисел с доски? Приведите пример, когда это значение суммы достигается

Решение

а) Во-первых, покажем, что у нас есть только одно число, модуль которого больше 1. Если бы таких чисел было бы несколько, то можно было бы выбрать два числа a, b с наибольшими модулями и тогда их произведение было бы больше любого из чисел на доске, что противоречит условию. Аналогичные рассуждения можно провести для чисел, меньших единицы по модулю.

Во-вторых, если у нас есть число не равное ± 1, то -1 не может входить в набор. Иначе бы мы получили пару чисел больших единицы по модулю или меньших и вернулись бы к предыдущим двум рассуждениям. Следовательно, всего чисел не больше трех. Пример: $\frac{1}{2}, 1, 2$.

б) Заметим, что всевозможные наборы чисел, которые могут быть выписаны на доске, выглядят так: $\frac{1}{n}, 1, n$ (или их какое-то подмножество). Следовательно, максимально возможная сумма чисел будет достигаться при $n = 2019$ (можно показать, что при увеличении n сумма всех трех чисел будет увеличиваться: например, взяв производную от суммы, как функции, зависящей от n).

Ответ: б) $2020\frac{1}{2019}$.

Критерии оценки

а) *(15 баллов)*

+5 баллов за рассуждения о том, почему нет больше 1 числа, большего единицы.

+5 баллов за рассуждения о том, почему нет больше 1 числа, меньшего единицы.

+5 баллов за рассуждения о том, почему нет -1 в наборе.

б) *(5 баллов)*

+3 балла за рассуждения о том, почему подходящий набор будет составлен из трех чисел: единицы, числа n и его обратного.

+1 балл за рассуждения, почему при увеличении n сумма увеличивается.

+1 балл за пример.

Задача 6.5.4. (25 баллов)

Перед Васей и Петей стоят три корзины (желтая, белая и красная) и лежат 17 пронумерованных шаров. Сначала Вася берет по шару и с равной вероятностью кладет в одну из корзин. Затем Петя достает шар с номером 1 из той корзины, в которой он лежит.

а) *(5 баллов)* Какова вероятность того, что после этого среди корзин будет хотя бы одна пустая?

б) *(20 баллов)* После этого Петя с вероятностью $\frac{1}{2}$ достает из корзин шар с номером 2, или с вероятностью $\frac{1}{4}$ он достает из корзин два шара: с номером 2 и 3, или с вероятностью $\frac{1}{8}$ он достает из корзин три шара: с номером 2, 3 и 4 . . . Все 16 оставшихся шаров он достает с вероятностью $\frac{1}{2^{15}}$.

Какова вероятность того, что после этого среди корзин будет хотя бы одна пустая? (в этом пункте ответ может быть получен в виде суммы некоторого выражения; при его получении дальнейшие сокращения можно не проводить)

Решение

а) Допустим, что всего у нас n шаров ($n > 2$) в 3-х корзинах. Посчитаем вероятность того, что все корзины не пустые, с помощью формулы включений-исключений

$$p = 3 \cdot \left(\frac{2}{3}\right)^n - 3 \cdot \left(\frac{1}{3}\right)^n.$$

Заметим, что в наших корзинах остается 16 определенных шаров всегда, поэтому в этом пункте ответ будет получаться подстановкой $n = 16$ в формулу выше.

б) Здесь у нас повторяется тоже, что и в предыдущем пункте, но n может принимать разные значения с разной вероятностью. Тогда с вероятностью $\frac{1}{2}$ останется $n = 15$ шаров, с вероятностью $\frac{1}{4}$ останется $n = 14$ шаров, . . . Тогда итоговой вероятностью будет в каждом случае произведение вероятности того, что все три корзины не пустые при определенном числе шаров и вероятности того, что после выбора Пети останется это число шаров:

$$P = \frac{1}{2^{15}} + \frac{1}{2^{15}} + \frac{1}{2^{14}} + \sum_{n=3}^{15} \left(3 \cdot \left(\frac{2}{3}\right)^n - 3 \cdot \left(\frac{1}{3}\right)^n\right) \cdot \frac{1}{2^{16-n}}.$$

В этой сумме первые три слагаемых — это случаи, когда останется 0, 1 и 2 шара. Это сумму можно было не сокращать, хотя имеет место следующие выкладки.

Сократим немного выражение:

$$P = \frac{1}{2^{13}} + 3 \cdot \frac{1}{2^{13}} \cdot \sum_{t=0}^{12} \left(\cdot \left(\frac{2}{3}\right)^{t+3} - 3 \cdot \left(\frac{1}{3}\right)^{t+3} \right) \cdot 2^t.$$

По формуле суммы геометрической прогрессии имеем

$$P = \frac{1}{2^{13}} + \frac{3}{2^{13}} \cdot \left\{ \left(\frac{2}{3}\right)^3 \cdot \frac{1 - \left(\frac{2}{3}\right)^{13}}{1 - \frac{2}{3}} + \left(\frac{1}{3}\right)^3 \cdot \frac{1 - \left(\frac{1}{3}\right)^{13}}{\frac{1}{3} - 1} \right\}$$

Ответ: а), б) см. решение.

Критерии оценки

а) *(5 баллов)*

+2 балла за рассуждения, почему можно рассматривать 16 шаров вместо 17.

+3 балла за использование формулы включений-исключений.

б) *(20 баллов)*

+5 баллов за рассуждения о том, почему, как и в первом пункте, в каждом случае у нас вероятность зависит только от числа оставшихся шаров.

+5 баллов за рассуждения о том, почему надо перемножать вероятности.

+10 баллов за правильно выписанную формулу и объяснения, почему она так выглядит (включая вынесенные первые слагаемые).

Замечания

1. Если во втором пункте не разобран случай с $n = 0, 1, 2$, т. е. сумма итоговая неправильна и нет указаний нигде о том, что в этих случаях будет, то балл за формулу снижается на 5 баллов.

Задача 6.5.5. (25 баллов)

В прямоугольном треугольнике ABC (с прямым углом B) отметили точки D и E на сторонах AB и BC так, что DE параллельно AC. Из угла B опустили высоту BF на AC и на ней отметили точку L, из которой отрезок AD виден под прямым углом. Докажите, что лучи AL и LD отсекают на прямой BC отрезок, равный EC.

Решение

Обозначим $\angle FBC = \alpha$. Тогда можно показать, что $AD \cdot \operatorname{tg}\alpha = EC$ (можно, например, опустить высоты из E и D на AC и приравнять их, воспользовавшись тем, что $\angle CAB = \alpha$).

Пусть точки M и N получаются пересечением лучей AL и LD с прямой BC. Соединим точки A и N. Тогда D будет ортоцентром треугольника AMN.

Так как AB и NL — высоты в треугольнике AMN, то $\angle LBM = \angle MAN = \alpha$. Кроме того, равны углы NMA и ADL. Поэтому треугольники ADL и NLM подобны, откуда

$$\frac{AD}{MN} = \frac{AL}{LN} = \operatorname{ctg} \angle MAN = \operatorname{ctg}\alpha \Rightarrow AD \operatorname{tg}\alpha = MN.$$

Из двух равенств выше имеем искомое $MN = EC$.

Критерии оценки

+10 баллов за вывод формулы $AD \cdot \operatorname{tg}\alpha = EC$.

+5 баллов за замечание, что D ортоцентр треугольника AMN.

+10 баллов за вывод формулы $AD \operatorname{tg}\alpha = MN$.

Замечания

1. За полностью правильное решение, которое отличается от авторского баллы не снижать (в случае наличия ошибок, снижать баллы пропорционально их тяжести и влияния на решение задачи).

6.6. Виртуальная реальность. Информатика

Задача 6.6.1. Мишень (100 баллов)

Вы реализуете игру "дартс" в виртуальной реальности. Игра устроена следующим образом. Дана мишень в форме круга радиуса R, расположенная в плоскости XY с центром в координатах $(0, 0)$, разбитая на n равных секторов. Сектора пронумерованы от 1 до n по часовой стрелке. В момент броска мишень находится в таком положении, что сегмент с номером 1 расположен в верхней полуплоскости справа от оси Y. Мишень закручивается вокруг оси Z по часовой стрелке и приобретает постоянную угловую скорость w градусов в секунду.

Игрок, стоя лицом к мишени на расстоянии z, бросает дротик из координаты (x, y) в плоскости XY перпендикулярно мишени в её сторону. Дротик летит прямолинейно с постоянной скоростью v, не изменяя высоту полёта.

Требуется написать программу, которая определяет номер сектора, в который попадёт игрок.

Формат входных данных

Входные данные содержат целые числа R, w, x, y, z, v, n. Гарантируется, что участник не попал в границу между секторами.

Ограничения:

$1 \leqslant r, w, v, z \leqslant 100$ $-100 \leqslant x, y \leqslant 100$ $2 \leqslant n \leqslant 100$

Формат выходных данных

Выходные данные должны содержать единственное целое число – номер сектора, в который попал участник, или 0, если участник не попал в мишень.

Пример №1

Стандартный ввод
`10 20 10 0 100 100 100`

Стандартный вывод
`20`

Решение

Для того что проверить попадёт ли игрок в мишень достаточно проверить следующее условие: $R^2 \leqslant x^2 + y^2$.

Для вычисления времени полёта дротика до мишени используем формулу $t = z/v$. Используем формулу $Rotation = w \cdot t$ для того, чтобы узнать, на сколько градусов повернётся мишень во время полёта дротика.

Далее вычислим угол $StartAngle$ к оси координат OX радиус-вектора (x, y). Для удобства повернём $StartAngle$ на $-\pi/2$. $StartAngle = atan2(y, x) - \pi/2$.

Посчитаем конечный угол в радианах:

$$ResultAngle = StartAngle + Rotation/360 \cdot 2 \cdot \pi.$$

Преобразуем угол $ResultAngle$ так, чтобы он лежал в диапазоне от 0 до $2 \cdot \pi$.

Размер одного сектора $OneSector = 2 \cdot \pi/n$.

Тогда искомый индекс $n - \lfloor ResultAngel/OneSector \rfloor$.

Пример программы-решения

Ниже представлено решение на языке C++

```
#define _CRT_SECURE_NO_WARNINGS
#define _USE_MATH_DEFINES

#pragma comment(linker, "/STACK:66777216")

#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <vector>
#include <algorithm>
#include <cmath>
#include <stack>
#include <functional>
#include <set>
#include <queue>
#include <string>
#include <map>
#include <iomanip>
#include <sstream>
#include <cassert>

#define sqr(x) ((x)*(x))

using namespace std;

int main()
{
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);

        int R, r, w, x, y, z, v, n;
        cin >> R >> w >> x >> y >> z >> v >> n;
```

```
33
34          if (x * x + y * y > R * R)
35          {
36                  cout << 0;
37                  return 0;
38          }
39
40          double t = double(z) / double(v);
41          double start_angle = atan2(y, x) + 2 * M_PI - M_PI / 2;
42          double total_rotation = (t * w) / 360 * 2 * M_PI + start_angle;
43          double cur_sin = sin(total_rotation);
44          double cur_cos = cos(total_rotation);
45          double cur_angle = atan2(cur_sin, cur_cos);
46          if (cur_angle < 0)
47                  cur_angle += 2 * M_PI;
48          double one_sector = 2 * M_PI / n;
49          int ind = cur_angle / one_sector;
50
51          cout << n - ind;
52
53  }
```

Задача 6.6.2. Распознавание метки (100 баллов)

Приложение дополненной реальности пытается найти на изображении, полученном с камеры телефона, метку, представляющую собой ярко раскрашенный прямоугольник. Благодаря контрастному цвету прямоугольника на этапе предварительной обработки изображение удалось преобразовать в массив из W на H элементов, каждый из которых равен либо '#' (ASCII 35), если он принадлежит прямоугольнику, либо '.' (ASCII 46) в противном случае.

Алгоритм предварительной обработки неидеален, и мог совершить от 0 до N ошибок — выдать '#' вместо '.' или наоборот.

В первом примере теста ошибочным может быть левый верхний элемент элемент непосредственно справа от него, элемент непосредственно снизу от него. Все остальные варианты из нуля или одной ошибки не дают прямоугольника в результате исправления.

Напишите программу, которая по данному изображению подсчитывает количество различных возможных положений метки.

Формат входных данных

Первая строка входного файла содержит целые числа WHN. Следующие H строк содержат по W символов . и # каждая – изображение после предварительной обработки.

Ограничения:

$1 \leqslant W, H \leqslant 100$, $0 \leqslant N \leqslant W \times H$

Формат выходных данных

Выходной файл должен содержать единственное целое число — количество возможных расположений метки.

Система оценки

Решения работающие для $W, H \leqslant 20$ оцениваются из 50 баллов. Баллы выставляются за каждый успешно пройденный тест.

Пример №1

Стандартный ввод
3 3 1 .#. ##. ...
Стандартный вывод
3

Решение

Прямоугольник, подходящий для того, чтобы быть меткой, должен обладать следующим свойством: количество точек внутри этого прямоугольника + количество решёток вне этого прямоугольника не должно превосходить N.

Так как $W, H \leqslant 100$, можно перебрать все возможные прямоугольники и с помощью префиксных сумм ($. = 1$, $\# = 0$) *(100 баллов)* или двойного цикла *(50 баллов)*, посчитать количество точек внутри прямоугольника. Это количество обозначим P. Пусть рассматриваемый прямоугольник имеет координаты $x1, y1$ — левый верхний угол, $x2, y2$ — правый нижний угол. Тогда общее количество клеток, входящих в рассматриваемый прямоугольник, $S = (x2 - x1 + 1) \cdot (y2 - y1 + 1)$.

Общее количество решёток на изображении обозначим как C. Тогда количество решёток вне выбранного прямоугольника равно $K = C - (S - P)$. Следовательно, количество действий для исправления изображения равно $K + P$. Асимптотика решения с использованием префикс сумм $O(W^4)$, с использованием двойного цикла – $O(W^6)$.

Пример программы-решения

Ниже представлено решение на языке C++

```
#define _CRT_SECURE_NO_WARNINGS
#define _USE_MATH_DEFINES

#pragma comment(linker, "/STACK:66777216")

#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <vector>
#include <algorithm>
#include <cmath>
#include <stack>
#include <functional>
#include <set>
```

```cpp
#include <queue>
#include <string>
#include <map>
#include <iomanip>
#include <sstream>
#include <cassert>

#define sqr(x) ((x)*(x))

using namespace std;

int mp[101][101];
int pref_sum[101][101];

int calc_sum(int y1, int x1, int y2, int x2)
{
        int res = pref_sum[y2][x2];
        res += pref_sum[y1 - 1][x1 - 1];
        res -= pref_sum[y1 - 1][x2];
        res -= pref_sum[y2][x1 - 1];
        return res;
}

int main()
{
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
        int W, H, N;
        cin >> W >> H >> N;

        for (int i = 1; i <= H; i++)
        {
                int row_sum = 0;
                for (int j = 1; j <= W; j++)
                {
                        char val;
                        cin >> val;
                        if (val == '#')
                                mp[i][j] = 1;
                        row_sum += mp[i][j];
                        pref_sum[i][j] = row_sum;
                        pref_sum[i][j] += pref_sum[i - 1][j];
                }
        }
        int ans = 0;

        for (int i = 1; i <= H; i++)
                for (int j = 1; j <= W; j++)
                        for (int ii = i; ii <= H; ii++)
                                for (int jj = j; jj <= W; jj++)
                                {
                                        int sum_center = calc_sum(i, j, ii, jj);
                                        int res = (ii - i + 1) * (jj - j + 1) -
                                                sum_center;
                                        int res_outside = calc_sum(1, 1, H, W) -
                                                sum_center;
                                        res += res_outside;
                                        if (res <= N)
                                                ans++;
                                }
```

```
75
76          cout << ans;
77  }
```

Задача 6.6.3. Полоса препятствий (100 баллов)

Одно из заданий на соревнованиях по подводной робототехнике — полоса препятствий. Полоса состоит из n подряд идущих стен, высота i-й стены h_i.

Цель робота — преодолеть все стены, достать флаг, который находится в конце полосы за всеми стенами, и вернуться обратно. Стену можно преодолеть двумя способами: подняться, проплыть над ней и опуститься обратно или сделать подкоп. Если стена была преодолена с помощью подкопа, то в при проходе в обратную сторону можно воспользоваться уже готовым подкопом. Всего разрешается сделать не более m подкопов.

Робот имеет ограниченный заряд батареи. Изначально он способен подняться над стеной высотой k или меньше, далее после каждого подъёма максимальная высота стены, которую может преодолеть робот, снижается на 1.

Робот поддерживает две команды: D – идти через подкоп (если подкопа ещё нет, то робот выкапывает его), U – проплыть над препятствием.

Формат входных данных

Первая строка входного файла содержит целые числа nmk. Следующая строка содержат n целых чисел h_i - высоты стен.

Ограничения: $1 \leqslant n, m \leqslant 10^5$, $1 \leqslant k, h_i \leqslant 10^9$

Формат выходных данных

Выходной файл должен содержать строку NO, если задание выполнить невозможно. В противном случае – две строки: строку YES и строку из $2n$ символов U и D – последовательность команд робота, которая позволит ему преодолеть полосу препятствий и вернуться обратно. Первые n символов содержат описание прохождение от стены с номером 1 до стены с номером n, следующие n символов содержат описание прохождения от стены с номером n до стены с номером 1.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	40	$1 \leqslant n, m \leqslant 10^2$, $1 \leqslant k, h_i \leqslant 10^9$	-	полная
2	60	$1 \leqslant n, m \leqslant 10^5$, $1 \leqslant k, h_i \leqslant 10^9$	1	полная

Пример №1

Стандартный ввод
10 5 10 10 1 4 2 4 3 4 4 8 9

Стандартный вывод
YES DUDUDUUUDDDDUUUDUDUD

Решение

1. Можно потратить все подкопы, если $n \leqslant m$, иначе можно сделать n подкопов и пройти маршрут только с помощью команд D. Далее всегда считаем, что $n \leqslant m$.
2. Если стена преодолевается с помощью подкопа, то подкоп лучше сделать при первом проходе стены. Это позволит сэкономить один дополнительный прыжок.
3. Исходя из пункта 2, каждую стену, через которую следует преодолеть с помощью подъёма, мы переплываем дважды.
4. Теперь мы знаем, что всего следует сделать $2 \cdot n - 2 \cdot m$ подъёмов и высота последнего подъёма $H = k - (2 \cdot n - 2 \cdot m) + 1$.
5. Не важно, сможет ли робот переплыть препятствие при первом проходе, так как при обратном проходе максимальная высота подъёма робота будет меньше.
6. Используя пункты 1-5. Для решения задачи нам достаточно пробежать по всем стенам слева направо и действовать следующим образом: если высота стены меньше или равна H, то эту стену мы переплываем и увеличиваем максимальную возможную высоту подъёма H на 1. В противном случае эту стену следует преодолеть подкопом. Если уже робот уже смог преодолеть $n - m$ стен, то все оставшиеся стены проходим подкопом. Далее можно легко восстановить ответ, зная, какие стены как преодолеваются подкопом. Асимптотика решения $O(n)$.

Пример программы-решения

Ниже представлено решение на языке C++

```
#define _CRT_SECURE_NO_WARNINGS
#define _USE_MATH_DEFINES

#pragma comment(linker, "/STACK:66777216")

#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <vector>
#include <algorithm>
#include <cmath>
#include <stack>
#include <functional>
#include <set>
#include <queue>
```

```cpp
#include <string>
#include <map>
#include <iomanip>
#include <sstream>
#include <cassert>

#define sqr(x) ((x)*(x))

using namespace std;

int arr[1000000];
int mark[1000000];

int main()
{
	freopen("input.txt", "r", stdin);
	freopen("output.txt", "w", stdout);
	int n, m, k;
	cin >> n >> m >> k;
	int goal = k - n + m + 1;
	int val = k - 2 * n + 2 * m + 1;

	for (int i = 0; i < n; i++)
	{
		cin >> arr[i];
		if (arr[i] <= val && val <= k && val < goal)
		{
			val++;
		}
		else
		{
			mark[i] = 1;
		}
	}
	if (val >= goal)
	{
		cout << "YES" << endl;

		for (int i = 0; i < n; i++)
			if (mark[i])
				cout << "D";
			else
				cout << "U";

		for (int i = n - 1; i >= 0; i--)
			if (mark[i])
				cout << "D";
			else
				cout << "U";
	}
	else
		cout << "NO";
}
```