Решение. Все классы

Построим алгоритм решения задачи для произвольного количества заряженных тел.

0. Будем говорить далее о головах студентов, приобретших электрический заряд, как о материальных точках, а ряд, по которому они разлетелись, представим в виде горизонтального отрезка длины L. Координатную ось направим вправо вдоль отрезка, а за начало координат примем его левый конец. Занумеруем точки (заряды) слева направо и обозначим через x_k их координаты, $k = 1, 2, \ldots, N$, где N – общее количество точек.

1. Начнем с того, что итоговое положение двух точек очевидно. Поскольку все заряды имеют один и тот же знак, то все силы, возникающие между ними, являются силами отталкивания. Следовательно, крайние (слева и справа) точки должны в итоге оказаться на краях отрезка. Поэтому можно считать, что всегда $x_1 = 0$, $x_N = L$.

Поскольку, согласно условию, начальное положение не влияет на итог процесса, расположим остальные точки на равном расстоянии друг от друга. Тогда их координаты будут

$$x_k = \frac{(k-1)\,L}{N-1}.$$

2. Отвлечемся от конкретных числовых значений и рассмотрим более общую задачу. Пусть известны координаты всех точек x_k ($k = \overline{1, N}$) в некоторый момент времени. Нужно найти координаты этих же точек y_k ($k = \overline{1, N}$) через интервал времени $t_0 = 1$ с (величину t_0 нужно прописывать в формулах, чтобы избежать путаницы с размерностью). В соответствии с условием скорость каждой точки равна нулю. Их новое положение определяется уравнением движения

$$y_k = x_k + \frac{a_k\, t_0^2}{2}, \quad k = \overline{2, N-1}. \tag{1}$$

Здесь учтено, что $y_1 = x_1 = 0$ и $y_N = x_N = L$.

Ускорение точек найдем из второго закона Ньютона

$$m\, a_k = \sum_{\substack{j=1,\\ j\neq k}}^{N} F_{k,j},$$

который будем записывать в проекции на введенную ось. Через $F_{k,j}$ обозначена проекция силы, действующей на точку k (на заряд q_k) со стороны точки j (заряда q_j).

Модуль силы равен

$$|F_{k,j}| = \frac{1}{4\pi\varepsilon_0} \cdot \frac{q_k\, q_j}{|x_k - x_j|^2}.$$

Для определения направления силы (иными словами, знака ее проекции) воспользуемся величиной

$$\frac{x_k - x_j}{|x_k - x_j|}.$$

Она равна -1, если точка j находится правее точки k, и $+1$ в противоположном случае. Собирая все вместе, получаем для ускорения выражение

$$a_k = \frac{F_k}{m}, \qquad \text{где} \qquad F_k = \frac{1}{4\pi\varepsilon_0} \cdot \sum_{\substack{m=1,\\ m\neq k}}^{N} \frac{q_k\, q_j (x_k - x_j)}{|x_k - x_j|^3}, \quad k = \overline{2, N-1}. \tag{2}$$

После того, как по формулам (1) и (2) будут найдены новые положения всех точек через одну секунду, можно повторить те же действия и найти их положение по прошествии еще одной секунды и так далее.

3. Запишем первый вариант построенного алгоритма вычислений.

Алгоритм «Улетный» (пре альфа-версия)
Начало алгоритма
Задать L, m, t_0, N; q_k $(k=\overline{1,N})$
ДЛЯ k ОТ 1 ДО N
$$x_k = \frac{(k-1)\,L}{N-1}$$
$S=0$
ПОКА S меньше заданного количества шагов
Для каждой k-ой точки выполнить:
ДЛЯ k ОТ 2 ДО N-1
Найти равнодействующую:
$F=0$
ДЛЯ j ОТ 1 ДО N
ЕСЛИ $j \neq k$
$$F = F + \frac{q_k\,q_j(x_k-x_j)}{|x_k-x_j|^3}$$
$$F = F\cdot\frac{1}{4\pi\varepsilon_0}$$
Найти ускорение:
$a=F/m$
Найти новое положение:
$$y_k = x_k + \frac{a\,t_0^2}{2}$$
Пересохранить новое положение:
$x_k = y_k$
$S=S+1$
Вывести x_k
Конец алгоритма

В алгоритме отсутствует индексация сил и ускорений, т.к. эти величины имеют вспомогательный характер и запоминать их для каждой точки нет необходимости.

4. Написанный алгоритм выполняет указанное количество шагов, каждый из которых соответствует очередной секунде процесса.

Выполнив этот алгоритм для одного и для двух шагов, получим ответы на 2 вопрос.

Однако количество шагов, необходимое для нахождения конечных положений точек, неизвестно заранее. Поэтому требуется другое условие выхода из внешнего цикла. Его можно обнаружить в условии задачи, согласно которому можно пренебрегать перемещениями, не превосходящими величины $d = 0.05$ м. Как только будет обнаружено, что все точки сместились на столь малую величину, расчет следует прекратить.

Для этого в алгоритм нужно ввести логическую переменную-ключ, которая будет истиной, если все перемещения $y_k - x_k$ по модулю не превышают d, и ложью, если хотя бы одно превышает. Назовем эту пременную Key и дополним алгоритм описанными действиями.

Еще один момент, не учтенный в алгоритме, состоит в том, что после очередного шага какие-то две точки могут оказаться в одном месте. Это приведет к делению на ноль в формуле (2) и аварийному завершению работы программы. Чтобы избежать этого, воспользуемся тем, что расчет носит приближенный характер и не претендует на точное описание движения точек (но, вместе с тем, выдает верно их окончательное положение). Поэтому будем попросту игнорировать вклад точки j в равнодействующую, если ее положение совпадает с положением точки k.

Запишем окончательный вариант алгоритма, добавив в него два дополнения и убрав комментарии.

Алгоритм «Улетный» (релиз)
Начало алгоритма
Задать L, m, t_0, N; x_k, q_k $(k=\overline{1,N})$
$Key =$ ЛОЖЬ
ПОКА $Key =$ ЛОЖЬ
$Key =$ ИСТИНА
ДЛЯ k ОТ 2 ДО N-1
$F=0$
ДЛЯ j ОТ 1 ДО N
ЕСЛИ $(j \neq k)$ и $(x_j \neq x_k)$

$$F = F + \frac{q_k\, q_j (x_k - x_j)}{|x_k - x_j|^3}$$

$$F = F \cdot \frac{1}{4\pi\varepsilon_0}$$

$a = F/m$

$$y_k = x_k + \frac{a\, t_0^2}{2}$$

$Key = Key$ И $(|y_k - x_k| \leq d)$
$x_k = y_k$
Вывести x_k
Конец алгоритма

Запустив алгоритм, найдем ответ на третий вопрос.

5. (Только 10 класс.) Для получения ответа на 4-й вопрос необходимо провести вычислительный эксперимент. Будем запускать программу при различных значениях q_5 и следить за тем, когда координаты всех остальных точек станут не больше, чем $2L/3$.

Для этого можно, например, монотонно увеличивать величину q_5 с шагом $\Delta q = 10^{-3}$ мКл (можно даже автоматизировать процесс). Можно применить другую стратегию: подобрать любое значение q_5, при котором остальные точки группируются в нужной части отрезка, а затем подбирать (например, бисекцией) величину q_5 с заданной точностью. Возможны и другие подходы.

5. (Только 11 класс.) Для получения ответа на 4-й вопрос достаточно запустить алгоритм для четырех зарядов, т.е. при $N=4$. При грамотном кодировании это не должно требовать внесения изменений в текст программы.

6. Если провести расчеты по приведенным алгоритмам, то получим следующий округленный

Ответ:

1, 2. Координаты на первых шагах расчета (в метрах, округленно)

	x_1	x_2	x_3	x_4	x_5
начальный момент	0	3	6	9	12
через t_0	0	4,28	6,82	9,50	12,00
через $2t_0$	0	4,50	7,17	9,84	12,00

3. Окончательное положение (в метрах, округленно)

x_1	x_2	x_3	x_4	x_5
0	5,19	8,13	10,38	12,00

4. (10) Искомый заряд $Q = 0{,}91$ мКл.

4. (11) Новое окончательное положение (в метрах, округленно)

x_1	x_2	x_3	x_4
0	6,15	9,55	12,0