

§3 Заключительный этап: индивидуальная часть

Заключительный этап олимпиады состоит из двух частей: индивидуальное решение задач по предметам (математика, физика, информатика) и командное решение инженерной задачи. На индивидуальное решение задач дается по 2 часа на один предмет. Для каждого из параллелей (9 класс или 10-11 класс) предлагается свой набор задач по физике, задачи по математике и информатике общие для всех участников. Решение каждой задачи дает определенное количество баллов (см. критерии оценки). По математике и физике за каждую задачу можно получить от 0 до указанного количества баллов. Решение задач по информатике предполагало написание программ, допускалось использовать один из двух языков программирования: Python или C++. Баллы по информатике зачисляются в полном объеме за правильное решение задачи. Участники получают оценку за решение задач в совокупности по всем предметам данного профиля (математика, физика и информатика) — суммарно от 0 до 38 баллов.

3.1. Задачи по математике

Задача 3.1.1 (4 балла)

Радиус Земли $R=6370$ км, масса Земли $M_3 = 1$, орбита спутника считается круговой. Спутник летит в поле тяготения Земли на высоте 2000 км над поверхностью. Какова его орбитальная скорость?

РЕШЕНИЕ:

Решение: $v = \sqrt{\frac{GM}{R+h}}$ Поскольку масса Земли нам дана не в килограммах, а в массах Земли, то и значение гравитационной постоянной нам понадобится не в системе Си, а в системе единиц, связанной с массой Земли.

$$G = 3986004418 \cdot 10^5 \frac{M^3}{M_3 \cdot c^2} \quad \text{Подставляя получаем} \quad v = 6900 \text{ м/с}$$

Примечание: разумеется не предполагается, что участники помнят наизусть гравитационную постоянную. Решения, в которых G не подставлено в формулу засчитывались. А вот решения, в которых подставлено G в системе Си не засчитывались.

Критерии оценки:

- верное решение — 4 балла
- приведена верная формула, но в расчете используются значения гравитационной

постоянной и массы Земли в различных системах — 2 балла

- приведена верная формула, но в расчете используются значения гравитационной постоянной и массы Земли в различных системах + допущены ошибки в вычислениях — 0 баллов
- ошибка в вычислениях на 1 порядок, возможно описка — 3 балла

Задача 3.1.2 (4 балла)

Радиус Земли $R=6370$ км, масса Земли $M_3 = 1$, орбита спутника считается круговой. Астроном Сережа наблюдает за спутником с поверхности Земли. Он знает, что спутник летит по орбите на высоте 2000 км над Землей. И что в момент времени $t=0$ спутник был над точкой на поверхности Земли, диаметрально противоположной Сереже. В какой момент времени Сережа увидит спутник на горизонте?

РЕШЕНИЕ:

Сережа видит спутник на горизонте, его взгляд направлен по касательной к

поверхности Земли. Спутник к этому моменту пролетел угол $\phi = 180^\circ - \arccos \frac{R}{R+h} \approx 140^\circ$, т.е. $7/18$ длины орбиты.

Путь равен $\frac{7}{9}\pi(R+h) \approx 2450$ км.

Поделим на скорость из предыдущей задачи $2450000 : 6900 = 355$ секунд.

Критерии оценки:

- верное решение — 4 балла
- найдено время, через которое спутник будет ровно над Сережей — 1 балл
- верная идея решения, не доведенная до конца (например, из-за ошибки в задаче 1) — 1 балл
- верное решение с вычислительной ошибкой — 2 балла
- верная идея решения, в процессе решения участник ищет не тот угол — 1 балл

Задача 3.1.3 (4 балла)

Среди всех решений системы:

$$x^2 + y^2 = 4,$$

$$z^2 + t^2 = 9,$$

$$xt + yz = 6$$

выберите те, для которых величина $x + z$ принимает наибольшее значение.

РЕШЕНИЕ:

Рассмотрим векторы $\mathbf{a} = (x, y)$ и $\mathbf{b} = (t, z)$. По условию $|\mathbf{a}| = 2$, $|\mathbf{b}| = 3$, $(\mathbf{a}, \mathbf{b}) = 6 = |\mathbf{a}||\mathbf{b}|$. Значит, векторы \mathbf{a} и \mathbf{b} направлены одинаково, то есть $\mathbf{a} = 2(u, v)$, $\mathbf{b} = 3(u, v)$.

Положим $u = \cos \varphi$, $v = \sin \varphi$, тогда:

$$x + z = 2 \cos \varphi + 3 \sin \varphi = \sqrt{13} \sin(\varphi + \alpha) \leq \sqrt{13}.$$

Максимальное значение $\sqrt{13}$ достигается, когда $\cos \varphi = \frac{2}{\sqrt{13}}$, $\sin \varphi = \frac{3}{\sqrt{13}}$.

Ответ: $x = \frac{4}{\sqrt{13}}$, $y = \frac{6}{\sqrt{13}}$, $z = \frac{9}{\sqrt{13}}$, $t = \frac{6}{\sqrt{13}}$.

Критерии оценки:

- полное решение — 4 балла
- сделана замена переменных, но система не решена — 1 балл

Задача 3.1.4 (4 балла)

Магнитометр выдает для каждого градуса долготы φ три координаты вектора магнитного поля $(\sin \varphi, \sin 2\varphi, \cos 2\varphi)$. Найдите долготу, на которой вектор магнитного поля имеет минимальную длину.

РЕШЕНИЕ:

Квадрат длины вектора магнитного поля равен $\sin^2 \varphi + \sin^2 2\varphi + \cos^2 2\varphi$ и эту величину надо минимизировать. Сначала преобразуем это выражение, воспользовавшись формулами двойного аргумента

$$\begin{aligned} \sin^2 \varphi + \sin^2 2\varphi + \cos^2 2\varphi &= \sin^2 \varphi + 2 \sin^2 \varphi \cos^2 \varphi + (1 - 2 \sin^2 \varphi)^2 = \\ &= \sin^2 \varphi + 2 \sin^2 \varphi (1 - \sin^2 \varphi) + (1 - 2 \sin^2 \varphi)^2 = 2 \sin^4 \varphi - \sin^2 \varphi + 1 \end{aligned}$$

Пусть теперь $\sin^2 \varphi = t$

Наше выражение примет вид $2t^2 - t + 1$ Это парабола «рогами вверх», ее минимум

достигается на оси, то есть при $t = \frac{1}{4}$

Отсюда $\sin \varphi = \pm \frac{1}{2}$, т.е. $\varphi = 30^\circ; 150^\circ; 210^\circ; 330^\circ$

Ответ: $\varphi = 30^\circ; 150^\circ; 210^\circ; 330^\circ$

Критерии оценки:

- полное решение — 4 балла
- участник верно понял, какую величину надо минимизировать — 1 балл

3.2. Задачи по физике (9 класс)

Задача 3.2.1 (3 балла)

Спутник пролетая мимо астероида, за счет его гравитационного притяжения изменил направление своего движения на 90° , при этом скорость спутника в системе отсчета в которой астероид покоился уменьшилась в 20 раз. Как относятся массы астероида и спутника? Двигатели спутник не включал, влиянием других масс можно пренебречь.

РЕШЕНИЕ:

Система «космический корабль — астероид» является замкнутой и, в указанных приближениях, к ней можно применить закон сохранения импульса и закон сохранения энергии.

$$\begin{cases} \frac{mV_0^2}{2} = \frac{mV_1^2}{2} + \frac{MU^2}{2} \\ m\vec{V}_0 = m\vec{V}_1 + M\vec{U} \end{cases}$$

В системе отсчета, в которой астероид первоначально покоился, для модуля импульса можно записать, используя теорему Пифагора:

$$M^2U^2 = m^2V_0^2 + m^2V_1^2$$

используя это уравнение совместно с уравнением сохранения энергии можно получить уравнение на отношение масс:

$$\frac{M}{m} = \frac{V_0^2 + V_1^2}{V_0^2 - V_1^2} \approx 1.005$$

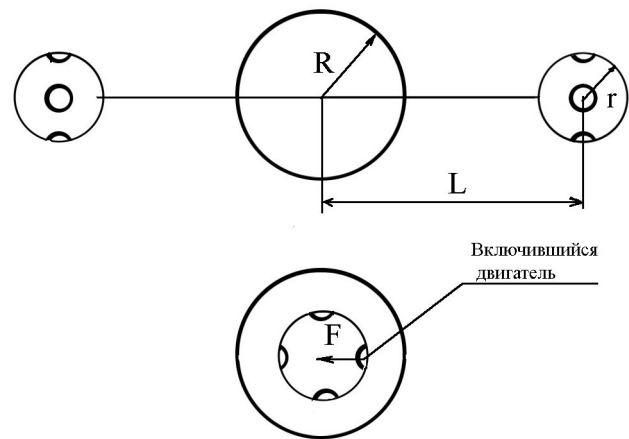
Критерии оценки:

- Записаны ЗСИ и ЗСЭ - по 0,5 балла
- Получено решение - 1 балл
- Получен ответ - 1 балл
- Если решения нет, но приведены разумные рассуждения в направлении решения - 0,5 балла

Задача 3.2.2 (3 балла)

Космический корабль, покоящийся в открытом космосе, вдалеке от планет

представляет из себя конструкцию, изображенную на рисунке. Корабль должен был выполнить маневр ускорения, включив на короткое время $\tau = 2\text{с}$ симметрично расположенные двигатели, однако, в результате технического сбоя сработал только один из них. Каков модуль скорости, которую приобрел после этого маневра корабль в системе отсчета, в которой исходно он покоился? На какой угол повернулся корабль на момент выключения двигателя относительно оси, перпендикулярной плоскости движения, проходящей через центр основного отсека корабля? Параметры корабля: масса корабля $M = 1200\text{кг}$, расстояние между центрами отсеков $L = 2\text{м}$, тяга двигателя $F = 710\text{Н}$, мощность двигателя $N = 50\text{кВт}$. Момент инерции корабля относительно оси, проходящей через центр масс перпендикулярно штангам равен $J = 903\text{кг}\cdot\text{м}^2$. Центры шаров лежат на прямой, проходящей вдоль соединяющей их штанги. Массой штанг, уменьшением массы корабля в результате сгорания топлива и гравитационным взаимодействием можно пренебречь. Считайте, что сила тяги не зависит от скорости, а масса в отсеках распределена однородно.



РЕШЕНИЕ:

Движение корабля можно разложить на движение центра масс и вращение вокруг него.

Запишем основное уравнение динамики вращения:

$$\frac{d(J\omega)}{dt} = \sum M_i, \text{ в нашем случае момент силы, которую производит двигатель,}$$

постоянен, также как и момент инерции.

$$\frac{Jd\omega}{dt} = FL, \text{ откуда в нашем случае (начальная угловая скорость } 0) - \omega = \frac{FL\tau}{J}, \text{ после}$$

этого, зная сколько времени работал двигатель можно легко рассчитать конечную угловую скорость и конечную кинетическую энергию вращения.

$$E_{\text{кр}} = \frac{J\omega^2}{2} = \frac{(FL\tau)^2}{2J}$$

Зная угловую скорость и то, что возрастает она линейно, можно рассчитать угол на который повернется корабль:

$$\varphi = \frac{\omega\tau}{2} \approx \pi$$

Полная работа совершенная двигателем равна: $A = N\tau$ пошла на приращение кинетической энергии вращения и кинетической энергии центра масс. Тогда кинетическая энергия центра масс:

$$E_{цм} = N\tau - \frac{(FL\tau)^2}{2J}, \text{ она же равна } E_{цм} = \frac{(M + 2m)V^2}{2}, \text{ откуда легко вычислить скорость}$$

поступательного движения. $V \approx 12.6 \text{ м/с}$.

Критерии оценки:

- Записаны основные уравнения динамики вращения - 1 балл
- Получена угловая скорость - 0,5 балла
- Вычислен угол поворота вокруг своей оси - 0,5 балла
- Получено выражение для линейной скорости - 0,5 балла.
- Вычислена линейная скорость - 0,5 балла.
- Если решения нет, но приведены разумные рассуждения в направлении решения - 0,5 балла

Задача 3.2.3 (3 балла)

Известно, что мощность теплового излучения единицы поверхности абсолютно черного тела пропорциональна температуре $N = \sigma T^4$ (где $\sigma = 5,67 \cdot 10^{-8} \text{ Вт/(м}^2 \cdot \text{К}^4)$ – постоянная Стефана–Больцмана). Шарообразный спутник радиусом $R = 0.5 \text{ м}$ облетает Землю на геостационарной орбите высотой $H = 35\,786 \text{ км}$. Какова температура спутника, считая что внутренних источников тепла в нем нет? Солнечная постоянная $L = 1367 \text{ Вт/м}^2$, радиус орбиты Земли $R_{\text{земли}} = 150 \text{ млн. километров}$. *Солнечная постоянная - суммарный поток солнечного излучения, проходящий за единицу времени через единичную площадку, ориентированную перпендикулярно потоку, на расстоянии одной астрономической единицы от Солнца вне земной атмосферы.*

РЕШЕНИЕ:

Все получаемое спутником тепло должно переизлучаться наружу.

Тогда $\sigma T^4 S = LS$, откуда $T = 394 \text{ К}$.

Можно показать, что время, которое спутник проводит в тени Земли пренебрежимо мало.

Критерии оценки:

- Записаны выражение равенства излучения и падающего света - 1 балла
- Получен ответ - 2 балла.

- Если решения нет, но приведены разумные рассуждения в направлении решения - 0,5 балла

Задача 3.2.4 (3 балла)

Геостационарный спутник на короткое время включил тормозные двигатели и уменьшил свою скорость на 2 м/с. На сколько изменился период его обращения?

РЕШЕНИЕ:

Используя второй закон Ньютона найдем скорость спутника на геостационарной орбите. $V_0 = 3070$ м/с. Тогда на новой орбите в перигее $(V_0 - \Delta V)R = V_1 R_1$.

Используя закон сохранения энергии:

$$\frac{(V_0 - \Delta V)^2}{2} - \frac{GM}{R} = \frac{V_1^2}{2} - \frac{GM}{R_1}, \text{ откуда } \frac{R + R_1}{2R} \approx \left(1 - \frac{2\Delta V}{V}\right), \text{ тогда пользуясь третьим}$$

законом Кеплера, можно получить:

$$\frac{T_1}{T} = \left(\frac{(R + R_1)}{2R}\right)^{3/2}, \text{ откуда изменение времени оборота равно } 169 \text{ с.}$$

Критерии оценки:

- Получены параметры геостационарной орбиты, в том числе скорость - 1 балла
- Записан закон сохранения энергии - 1 балла.
- Записан третий закон Кеплера - 0,5 балла
- Получен ответ - 0,5 балла
- Если решения нет, но приведены разумные рассуждения в направлении решения - 0,5 балла

3.3. Задачи по физике (10-11 класс)

Задача 3.3.1 (3 балла)

Выполняя гравитационный маневр вокруг астероида, масса которого примерно в 4 раза больше массы корабля, корабль изменил свой курс на угол $\alpha = 60^\circ$. Во сколько раз изменилась скорость корабля в системе отсчета в которой астероид покоился до встречи с кораблем, если влиянием других массивных тел можно пренебречь, а двигатели корабль не включал?

РЕШЕНИЕ:

Система космический корабль - астероид является замкнутой и, в указанных

приближениях, к ней можно применить закон сохранения импульса и закон сохранения энергии.

$$\begin{cases} \frac{mV_0^2}{2} = \frac{mV_1^2}{2} + \frac{MU^2}{2} \\ m\vec{V}_0 = m\vec{V}_1 + M\vec{U} \end{cases}$$

В системе отсчета, в которой астероид первоначально покоился, для модуля импульса можно записать, используя теорему косинусов:

$$M^2U^2 = m^2V_0^2 + m^2V_1^2 - 2m^2V_0V_1 \cos \alpha$$

используя это уравнение совместно с уравнением сохранения энергии можно получить квадратное уравнение на отношение скоростей:

$$\left(\frac{V_1}{V_0}\right)^2 - 2\left(\frac{\cos \alpha}{1 + \frac{M}{m}}\right)\left(\frac{V_1}{V_0}\right) + \left(\frac{1 - \frac{M}{m}}{1 + \frac{M}{m}}\right) = 0,$$

решив которое, можно получить ответ:

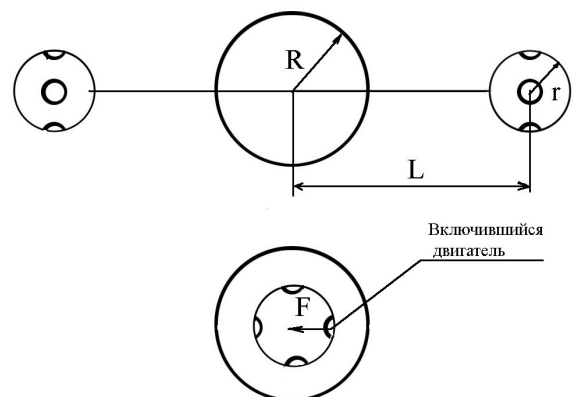
$$\frac{V_1}{V_0} \approx 0.119$$

Критерии оценки:

- Записаны ЗСИ и ЗСЭ по 0,5 балла
- Получено квадратное уравнение для модуля импульса 1 балл
- Получен ответ - 1 балл.
- Если решения нет, но приведены разумные рассуждения в направлении решения - 0,5 балла

Задача 3.3.2 (3 балла)

Космический корабль, покоящийся в открытом космосе, вдалеке от планет представляет из себя конструкцию, изображенную на рисунке. Корабль должен был выполнить маневр ускорения, включив на $\tau = 2$ с симметрично расположенные двигатели. Однако,



в результате технического сбоя сработал только один из них. Каков модуль скорости, которую приобрел после этого маневра корабль в системе отсчета, в которой исходно он покоился? На какой угол повернулся корабль на момент выключения двигателя относительно оси,

перпендикулярной плоскости движения, проходящей через центр основного отсека корабля? Параметры корабля: масса основного блока $M = 1000 \text{ кг}$ масса двигателя $m = 100 \text{ кг}$, радиус основной гондолы $R = 0.5 \text{ м}$, радиус гондол двигателей $r = 0.2 \text{ м}$, расстояние между центрами отсеков $L = 2 \text{ м}$, тяга двигателя $F = 710 \text{ Н}$, мощность двигателя $N = 50 \text{ кВт}$. Центры шаров лежат на прямой, проходящей вдоль соединяющей их штанги. Массой штанг, уменьшением массы корабля в результате сгорания топлива и гравитационным взаимодействием пренебрегите. Считайте, что сила тяги не зависит от скорости, а масса в отсеках распределена однородно. Момент инерции однородного шара относительно оси, проходящей через его центр $J = \frac{2}{5}mR^2$, где m - масса шара, а R - его радиус).

РЕШЕНИЕ:

Движение корабля можно разложить на движение центра масс и вращение вокруг него. Запишем основное уравнение динамики вращения:

$$\frac{d(J\omega)}{dt} = \sum M_i, \text{ в нашем случае момент силы, которую производит двигатель,}$$

постоянен, также как и момент инерции.

$$\frac{Jd\omega}{dt} = FL, \text{ откуда в нашем случае (начальная угловая скорость } 0) - \omega = \frac{FL\tau}{J}$$

Момент инерции можно рассчитать, воспользовавшись теоремой Штейнера:

$$J = \frac{2}{5}MR^2 + 2\left(\frac{2}{5}mr^2 + mL^2\right), \text{ после этого, зная сколько времени работал двигатель}$$

можно легко рассчитать конечную угловую скорость и конечную кинетическую энергию вращения.

$$E_{\text{вр}} = \frac{J\omega^2}{2} = \frac{(FL\tau)^2}{2J}$$

Зная угловую скорость и то, что возрастает она линейно, можно рассчитать угол на который повернется корабль:

$$\varphi = \frac{\omega\tau}{2} \approx \pi$$

Полная работа совершенная двигателем равна: $A = N\tau$ пошла на приращение кинетической энергии вращения и кинетической энергии центра масс. Тогда кинетическая энергия центра масс:

$$E_{\text{цм}} = N\tau - \frac{(FL\tau)^2}{2J}, \text{ она же равна } E_{\text{цм}} = \frac{(M + 2m)V^2}{2}, \text{ откуда легко вычислить скорость}$$

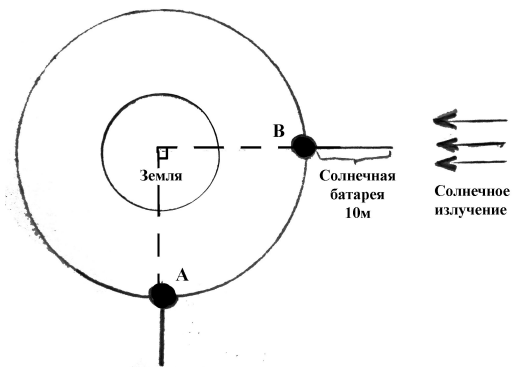
поступательного движения. $V \approx 12.6 \text{ м/с}$.

Критерии оценки:

- Записаны основные уравнения динамики вращения - 1 балл
- Вычислен момент инерции и получен угловая скорость - 0,5 балла
- Вычислен угол поворота вокруг своей оси - 0,5 балла
- Получено выражение для линейной скорости - 0,5 балла.
- Вычислена линейная скорость = 0,5 балла.
- Если решения нет, но приведены разумные рассуждения в направлении решения - 0,5 балла

Задача 3.3.3 (3 балла)

Спутник облетает Землю на геостационарной орбите высотой $H = 35\,786$ км. У спутника есть одна солнечная батарея $2\text{ м} \times 10\text{ м}$ направленная всегда перпендикулярно орбите спутника длинной стороной вдоль прямой соединяющей центры Земли и спутника (см. рисунок, плоскость батареи перпендикулярна плоскости рисунка).



Коэффициент отражения света от батареи $\beta = 75\%$ (остальной свет батареями поглощается). На какое время нужно запустить двигатель гиродина с крутящим моментом $2\text{ Н}\cdot\text{м}$, чтобы скомпенсировать вращение спутника, полученное из за солнечного излучения за время движения от положения А до положения В по орбите? Солнечная постоянная $L = 1367\text{ Вт/м}^2$, радиус орбиты Земли $R_{\text{Земли}} = 150$ млн. километров. Считайте, что коэффициент отражения не зависит от угла падения света. **Гиродин** — вращающееся инерциальное устройство, применяемое для высокоточной ориентации и стабилизации.

РЕШЕНИЕ:

Дополнительное вращение спутник приобретет в результате действия солнечного давления. Оценим силу давления.

$$F = \frac{\Delta p}{\Delta t}, \text{ причем изменение импульса света будет разным для отраженного и}$$

поглощенного света. Для отраженного света $\Delta p_{\text{отр}} = 2 \frac{E}{c} \cos \alpha$, для поглощенного

$\Delta p_{\text{отр}} = \frac{E}{c} \cos \alpha$, где c - скорость света, а α - угол падения, $E = LS\Delta t$ - энергия падающего

света.

Тогда: $F = \frac{LS\Delta t}{c\Delta t} (2\beta + (1 - \beta)) \cos \alpha = \frac{LS}{c} (\beta + 1) \cos \alpha$, тогда момент этой силы

действующий на солнечную батарею $M = \left(\frac{LS}{c} (\beta + 1) \cos \alpha \right) \frac{h}{2}$, т.к. силу можно считать приложенной к середине батареи. Усреднив это значение за время движения и умножив на это время мы получим значение момента импульса, полученного спутником.

Среднее значение от косинуса за от 0 до $\pi/2$ равно $\frac{2}{\pi} \int_0^{\pi/2} \cos \alpha d\alpha = \frac{2}{\pi}$, тогда

$\langle M \rangle = \frac{LS(\beta + 1)h}{\pi c}$, и за четверть оборота он сообщит момент импульса:

$\langle M \rangle \frac{T}{4} = \frac{LS(\beta + 1)hT}{4\pi c}$, где T - период оборота, т.е. одни сутки.

Гироскоп должен скомпенсировать этот момент:

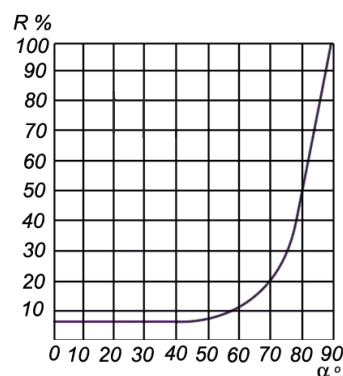
$M_{\text{гироскопа}} \tau = \langle M \rangle \frac{T}{4} = \frac{LS(\beta + 1)hT}{4\pi c}$, откуда искомое время $\tau \approx 5.5$ с.

Критерии оценки:

- Вычислена сила давления - 0,5 балл
- Записана зависимость момента силы от угла - 0,5 балла
- Произведено усреднение по углу - 1 балл
- Записано выражение для закона сохранения момента импульса - 0,5 баллов
- Вычислено время - 0,5 балла.
- Если решения нет, но приведены разумные рассуждения в направлении решения - 0,5 балла

Задача 3.3.4 (3 балла)

Солнечная батарея спутника из задачи 3 имеет коэффициент трансформации солнечной энергии в электрическую 10%, а электрическая цепь спутника с помощью аккумулятора позволяет усреднить полученную за время оборота вокруг Земли энергию от солнечных батарей.



Оцените, какова может быть максимальная мощность полезной нагрузки в цепи спутника. Солнечные батареи спутника двусторонние. Как изменится ответ, если учесть, что

зависимость коэффициента отражения от угла падения имеет вид изображенный на рисунке?

РЕШЕНИЕ:

Из соображений приведенных в решение задачи 3.3.3 ясно, что солнечная энергия попавшая на спутник за единицу времени равна $W = (1 - \beta)LS \cos \alpha$, полезную мощность можно получить умножив на КПД.

Для того, чтобы получить среднюю мощность за период, нужно снова усреднить по времени. Нужно только учесть, что из-за того, что батареи двусторонние, усреднение будет происходить по модулю косинуса, т.е. снова даст $\frac{2}{\pi}$, тогда средняя мощность:

$$W = \frac{2}{\pi}(1 - \beta)LS \approx 435 \text{ Вт.}$$

В данном случае мы пренебрегли временем, которое спутник проводит в тени Земли. Легко можно показать, что это время существенно меньше времени на свету.

Видно, что если использовать реальную кривую для коэффициента отражения, то среднее значение будет меньше, чем 0,75 практически на всем пути, таким образом мощность будет выше.

Критерии оценки:

- Записана мощность от угла - 1 балл
- Произведено усреднение по углу- 1 балл
- Вычислена максимальная мощность - 0,5 баллов
- Приведены рассуждения о случае реальной зависимости коэффициента отражения. - 0,5 балла.
- Если решения нет, но приведены разумные рассуждения в направлении решения - 0,5 балла

3.4. Задачи по информатике

Задача 3.4.1 «SMS со спутника» (1 балл)

Коля и Миша приехали на олимпиаду, чтоб хорошо проводить время и решать интересные задачи. Обычно они делают оба этих дела одновременно. Вот и сегодня, изучая документацию системы, мальчики обнаружили интересную возможность!

Оказывается, у спутника есть модуль `sputnik.transmitter` отвечающий за высокопроизводительную связь. Модуль позволяет пересылать сообщения трех типов: SMS, фотографии и сообщения с телеметрией. Всё это настолько заинтересовало ребят, что они

решили не терять время зря и прямо сейчас написать свою первую программу для спутника. Вот описание модуля из документации:

Класс `Transmitter.sputnik.transmitter` — подсистема полезной нагрузки (высокопроизводительной связи).

Методы:

- `send_data(msg_type, data, receiver, sender, timeout)` — отправить блок данных `data` типа `msg_type` на наземный измерительный пункт `receiver` (номер) от передатчика `sender` (номер) со сроком передачи `timeout` (с). могут быть переданы сообщения следующих типов:
 - `MESSAGE_PHOTO` — сообщение со снимком камеры аппарата;
 - `MESSAGE_SMS` — сообщение от системы обмена короткими сообщениями;
 - `MESSAGE_TELEMETRY` — сообщение с телеметрией.

Необязательными параметрами являются:

- `receiver` — тогда сообщение передается для всех наземных пунктов (`receiver == -1`);
- `sender` — тогда считается, что источником сообщения является сам аппарат (`sender == 0`);
- `timeout` — тогда считается, что сообщение имеет бесконечное время доставки.

Функция не возвращает данных.

Коля с Мишей поставили перед собой такую задачу: они наугад выбирают номер передатчика S и номер единственного измерительного пункта R . Потом берут какое-нибудь предложение M и пересылают его как SMS. Мальчики считают, что сообщения должны быть бессрочными. Помогите ребятам по известным данным составить команду, которую они должны будут использовать в своей программе

Формат входных данных:

В первой строке записано натуральное число S — номер передатчика. Во второй натуральное R — номер измерительного пункта. Оба числа не превосходят 10000. В третьей строке дано сообщение M , которое необходимо передать через SMS. Сообщение состоит из строчных букв английского алфавита и символов нижнего подчеркивания, длина сообщения M не больше 500 символов. Сообщение не может быть пустым.

Формат выходных данных:

Выведите в единственной строке команду, которая будет выполнять данную задачу, если её добавить в код.

СПОСОБ ОЦЕНКИ РАБОТЫ:

Для проверки результата используется следующий код на языке Python:

```
def check(reply, clue):
    stripper = lambda s: re.sub(r'^\s*|\s*$', '', s)
    squeezer = lambda s: re.sub(r'\s+', ' ', s)
    reply = reply.replace(' ', '').replace('\n', '').replace('None', '')
    clue = clue.replace(' ', '').replace('\n', '').replace('None', '')
    lines_reply = list(map(lambda s: squeezer(stripper(s)), reply.splitlines()))
    lines_clue = list(map(lambda s: squeezer(stripper(s)), clue.splitlines()))
    return lines_reply == lines_clue
```

РЕШЕНИЕ:

Задача требует умения работать со строками. Формирование и вывод ответа может выполнить такой код на языке Python:

```
01. sender, receiver, message = input().splitlines()
02. print("sputnik.transmitter(MESSAGE_SMS, '{}', {}, {}, )".format(message,
receiver, sender))
```

Основные ошибки:

- Вывод строки без кавычек.
- Вывод чисел в кавычках

Задача 3.4.2 «Конфиденциальность» (2 балла)

Научившись посылать сообщения через модуль высокопроизводительной связи, мальчики задумались, а что будет, если они будут посылать сверхсекретные сообщения, а их кто-нибудь перехватит. В итоге они решили ввести собственный шифр. Сложных шифров Коля с Мишей ещё не знают, поэтому решили для начала сделать шифр простой замены.

Миша составил алфавит, в котором каждой букве английского алфавита и каждой цифре сопоставляется другой знак. Коля же взял на себя задачу написать программу, которая будет по тексту M выводить его закодированную версию. От тебя же мы просим выполнить обратную процедуру и написать программу, которая сможет расшифровать сообщения по известному алфавиту.

Формат входных данных:

В первой строке задано $N < 100$ — число различных знаков, которые могут встречаться в сообщении. В следующих N строках указаны пары символов w_1 и w_2 записанные через знак '>'. При этом w_1 — символ в исходном сообщении, а w_2 — символ, который используется в шифре мальчиков вместо w_1 . На $N+2$ строке записано сообщение, которое получилось после применения кодирующей программы. Длина сообщения $L \leq 1000$

В исходном сообщении и в шифровке могут встречаться знаки пробела, но не могут встречаться символы переноса строки.

Формат выходных данных:

В единственной строке выведите исходное сообщение.

СПОСОБ ОЦЕНКИ РАБОТЫ:

Для генерации набора тестов и проверки результата используется следующий код на языке Python:

```
def generate():
    num_tests = 10
    tests = [("9\nh>g\ne>q\nl>o\no>j\n >t\nw>b\nr>r\nd>p\n!>3\ngqoojtbjrop3\n",
"hello world!\n"),]
    symbols = "qwertyuiopasdfghjklzxcvbnm"
    for i in range(num_tests):
        mes_length = random.randint(1, 1000)
        mes = []
        for j in range(mes_length):
            mes.append(random.choice(symbols))
        alph = list(set(mes))
        l_alphabet = len(alph)
        sym = list("qwertyuiopasdfghjklzxcvbnm")
        random.shuffle(sym)
        res = []
        decoder_dict = dict()
        for j in range(l_alphabet):
            res.append("{}>{}".format(sym[j], alph[j]))
            decoder_dict[alph[j]] = sym[j]
        test_quest = "{}\n{}\n{}\n".format(l_alphabet, "\n".join(res),
"".join(mes))
        test_ans= ''.join([decoder_dict[i] for i in mes])
        tests.append((test_quest, test_ans))
    return tests

def check(reply, clue):
    return reply.replace('\n', '') == clue.replace('\n', '')
```

РЕШЕНИЕ:

Задача на реализацию простого шифра замены. Для легкого преобразования букв в зашифрованном сообщении в буквы расшифрованного сообщения можно использовать стандартные коллекции языка Python список (list) или, что более удачно, словарь (dict).

Сначала необходимо построчно считать входной поток и разделить на символ и зашифрованный символ. В словарь по ключу, равному зашифрованной букве добавляем значение, равное исходной букве. После того как словарь сформирован, считываем сообщение и буква за буквой расшифровываем его. Результаты записываем в список, так как строка является неизменяемым типом данных. Последним действием преобразуем список в строку и выводим на экран.

Программа на языке Python, реализующая этот алгоритм:

```
01. n = int(input())
02. decoder_dict = dict()
03. for i in range(n):
04.     letter, coder = input().split('>')
```

```

05. decoder_dict[coder] = letter
06. message = input()
07. res = list()
08. for i in message:
09.     res.append(decoder_dict[i])
10. print(''.join(res))

```

Задача 3.2.3 «Восстановление чисел» (3 балла)

В свободное время Коля и Миша играют в следующую игру:

Миша выбирает и говорит Коле положительное число N , после чего он выписывает все целые числа от 1 до N по порядку. Затем вычеркивает 2 числа из полученной последовательности. Потом Миша считает число A , равное сумме квадратов оставшихся чисел, и B , равное сумме четвертых степеней оставшихся чисел. Эти две суммы Миша говорит Коле. Известно, что Миша очень внимателен во время игры и никогда не ошибается.

Задача Коли восстановить, что же за числа зачеркнул Миша. Он хочет научиться быстро отвечать на Мишины задачки. Помогите Коле справиться с этой проблемой и напишите программу, которая будет находить числа, которые Миша зачеркнул.

Формат входных данных:

В первой строке дано одно натуральное число $2 \leq N \leq 50000$, во второй строке следуют числа A и B . Где A — сумма квадратов оставшихся в строке чисел, а B — сумма четвертых степеней оставшихся чисел.

Формат входных данных:

В единственной строке выведите через пробел два числа, которые Миша вычеркнул.

Если решений несколько, выведите любое из них.

СПОСОБ ОЦЕНКИ РАБОТЫ:

Для генерации набора тестов и проверки результата используется следующий код на языке Python:

```

def generate():
    num_tests = 15
    tests = []
    for test in range(num_tests):
        n = random.randrange(50000)
        x1 = random.randint(2, n)
        x2 = random.randint(1, x1)
        sum2, sum4 = 0, 0
        for i in range(1, n+1):
            if i != x1 and i != x2:
                sum2 += i**2
                sum4 += i**4
        test_case = "{}\n{} {}\n".format(n, sum2, sum4)
        tests.append(test_case)
    return tests

def check(reply, clue):

```



```

a1, b1 = map(int, reply.split(' '))
a2, b2 = map(int, reply.split(' '))
if a1**2 + b1**2 == a2**2 + b2**2 and a1**4 + b1**4 == a2**4 + b2**4:
    return True
return False

```

РЕШЕНИЕ:

Задача описывает систему из двух уравнений с двумя неизвестными.

Обозначим числа, которые Миша вычеркнул, как x_1 и x_2 . Так же вычислим суммы квадратов и четвертых степеней от 1 до n , обозначим их sum_2 и sum_4 соответственно.

Тогда система уравнений будет иметь вид

$$x_1^2 + x_2^2 = sum_2 - a = nsum_2$$

$$x_1^4 + x_2^4 = sum_4 - b = nsum_4$$

Введя замену $x_1^2 = y_1, x_2^2 = y_2$ получим

$$y_1 + y_2 = sum_2 - a = nsum_2$$

$$y_1^2 + y_2^2 = sum_4 - b = nsum_4$$

После чего выразим из первого уравнения y_1

$$y_1 = nsum_2 - y_2$$

$$(nsum_2 - y_2)^2 + y_2^2 - nsum_4 = 0$$

$$2y_2^2 - 2nsum_2 y_2 + nsum_2^2 - nsum_4 = 0$$

Полученное квадратное уравнение по условию имеет ровно два решения. В конце необходимо из y_1, y_2 получить x_1, x_2 .

Программа на языке Python:

```

1.     n = int(input())
2.     a, b = map(int, input().split(' '))
3.     sum2, sum4 = 0, 0
4.     for i in range(1, n+1):
5.         sum2 += i**2
6.         sum4 += i**4
7.     n_sum2 = sum2 - a
8.     n_sum4 = sum4 - b
9.     # y1 = B - y2
10.    # y2**2 - n_sum2*y2 + (n_sum2**2 - n_sum4) / 2 = 0
11.    D = (n_sum2)**2 - 2 * (n_sum2**2 - n_sum4)
12.    sq_D = D ** 0.5
13.    y1 = (n_sum2 + sq_D) // 2
14.    y2 = (n_sum2 - sq_D) // 2
15.    x1, x2 = int(y1**0.5), int(y2**0.5)
16.    print("{} {}".format(x1, x2))

```

Можно заметить, что задача опирается на идею квадратного уравнения, которое заведомо имеет решение. Следовательно возможно быстрое решение без циклов.

Задача 3.2.4 «Фотография на память» (1 балл)

Когда с модулем высокопроизводительной связи было покончено, юные инженеры решили заняться изучением возможностей камеры. Миша, как более опытный, уже отправил модулю камеры запрос о его состоянии и даже получил ответ. Из документации он знает, что эти ответы-константы значат:

- `STATE_OFF` — Устройство выключено.
- `STATE_ON` — Устройство включено.
- `STATE_SLEEP` — Устройство находится в режиме экономии энергии (сна).
- `STATE_DEAD` — Устройство неисправно.

Школьник решил что он будет делать для каждого состояния системы.

Если устройство выключено, его следует включить. После чего сделать фотографию и записать в переменную `block_num` номер блока памяти, куда был помещен снимок.

Если устройство включено, то Миша сразу хочет сделать фотографию и записать в переменную `block_num` номер блока памяти, куда был помещен снимок.

Если устройство находится в режиме сна, то Миша делать ничего не будет.

Если устройство неисправно, Миша сообщает это Коле и выводит на экран сообщение «Camera is dead».

Если в качестве состояния системы поступила какая-то другая строка, Миша выдает сообщение «State is incorrect».

Вот ещё некоторая информация из документации:

Класс `Camera`. Подсистема полезной нагрузки (камеры).

Методы:

- `get_state` (без параметров) — получить текущее состояние камеры. Возвращает состояние (см. выше).
- `set_state` (`state`) — задать новое состояние `state` камере. Не возвращает данных.
- `take_photo` (без параметров) — сделать мгновенный снимок камерой. Возвращается номер блока памяти, в который был помещен снимок.

Вывод сообщений осуществляется при помощи команды `print`, например, может потребоваться вывести на экран команду `print("Hello!")`

Строка сообщения берется в кавычки!

Формат входных данных:

В единственной строке дан набор символов. Которые, как Миша предполагает, характеризуют состояние модуля камеры.

Формат выходных данных:

В выходном файле выведите команды, которые Миша должен добавить в свой код, чтоб запрограммировать те действия, которые соответствуют данному состоянию системы. Если потребуется больше одной команды, каждую команду выводите с новой строки.

СПОСОБ ОЦЕНКИ РАБОТЫ:

Для генерации набора тестов и проверки результата используется следующий код на языке Python:

```
def generate():
    num_tests = 6
    tests = ["STATE_OFF\n", "sputnik.camera.set_state(STATE_ON)\nblock_num =
sputnik.camera.take_photo()\n"),
            ("Camera is ON\n", "print('State is incorrect')\n"),
            ("STATE_ON\n", "block_num = sputnik.camera.take_photo()"),
            ("\n", "print('State is incorrect')\n"),
            ("STATE_DEAD\n", "print('Camera is dead')\n"),
            ("STATE_SLEEP\n", "\n")]
    return tests

def check(reply, clue):
    return reply.replace('\n', '').replace(' ', '').replace('None',
    '').replace('""', "''") == clue.replace('\n', '').replace(' ',
    '').replace('None', '').replace('""', "''")
```

РЕШЕНИЕ:

Задача схожа с первой задачей «SMS со спутника», но требует дополнительно умения использовать условные конструкции if-elif-else. Код решающий задачу может быть таким:

```
01. msg = input()
02. if msg == "STATE_OFF":
03.     print("sputnik.camera.set_state(STATE_ON)\nblock_num =
sputnik.camera.take_photo()")
04. elif msg == "STATE_ON":
05.     print("block_num = sputnik.camera.take_photo()")
06. elif msg == "STATE_DEAD":
07.     print("print('Camera is dead')")
08. elif msg == "STATE_SLEEP":
09.     print("")
10. else:
11.     print("print('State is incorrect')")
```

Задача 3.4.5 «Механизм» (3 балла)

Олимпиада это, конечно, здорово и интересно, но дома у Коли осталась компьютерная игра-головоломка. На новый уровень Коля перешел буквально перед самым отъездом из дома и теперь в уме размышляет, как будет проходить новый уровень.

Задача заключается в следующем: Коля собирает какой-то заумный механизм. Для

этого он должен последовательно устанавливать на стержень высоты h детали механизма. Каждая деталь представляет собой диск некоторой высоты. Все детали задаются двумя параметрами: высота детали k_i и радиус детали r_i . Известно, что все детали, несмотря на то, что имеют разный вид, вносят равный вклад в мощность механизма. Коля же должен сделать самый мощный механизм из возможных. При сборке механизма действуют следующие правила

Детали укладываются на стержень последовательно. Радиус детали, которая лежит выше, должен быть строго меньше радиуса любой детали, которая лежит ниже.

Суммарная высота всех деталей должна быть строго меньше высоты стержня.

Помогите Коле написать программу, которая сможет выдать план сборки самого мощного механизма.

Формат входных данных:

В первой строке задано число деталей $1 \leq N \leq 5000$ и высота стержня $h \leq 20000$. В следующих N строках записаны тройки целых неотрицательных чисел id_i — номер детали, $0 \leq k_i \leq 1000$ — высота i -ой детали и $1 \leq r_i \leq 10000$ — радиус i -ой детали. Гарантируется что номера деталей уникальны.

Формат выходных данных:

В первой строке выведите через пробел M — количество деталей в самом мощном механизме и высоту стержня h . В следующих M строках выведите информацию о деталях от самой нижней детали на стержне до самой верхней. Каждая строка должна содержать три числа, разделенные пробелами: номер детали, её высоту и её радиус.

Если существует несколько вариантов собрать мощнейший механизм, выведите любой из планов сборки.

СПОСОБ ОЦЕНКИ РАБОТЫ:

Для генерации набора тестов и проверки результата используется следующий код на языке Python:

```
def generate():
    num_tests = 3
    tests = []
    for test in range(num_tests):
        n = random.randint(1, 10)
        h = 0
        details = []
        for i in range(n):
            id = i + 1
            r = random.randint(1, 1000)
            k = random.randint(1, 100)
            h += k
            details.append("{} {} {}".format(id, k, r))
        h = int(0.7 * h)
```

```

        test_case = "{} {} \n{} \n".format(n, h, '\n'.join(details))
        tests.append(test_case)

num_tests = 10
for test in range(num_tests):
    n = random.randint(1, 5000)
    h = random.randint(1, 1000)
    details = []
    for i in range(n):
        ids = i + 1
        r = random.randint(1, 10000)
        k = random.randint(1, 1000)
        details.append("{} {} {}".format(ids, k, r))
    test_case = "{} {} \n{} \n".format(n, h, '\n'.join(details))
    tests.append(test_case)
return tests

def check(reply, clue):
    lines1 = reply.splitlines()
    len1, h1 = map(int, lines1[0].split(' '))

    lines2 = reply.splitlines()
    len2, h2 = map(int, lines2[0].split(' '))
    if len1 != len2 or h1 != h2:
        return False

    details = []
    ids, k, r = map(int, lines2[1].split(' '))
    h_n = h1
    details.append({'k': k, 'r': r})
    h_n -= details[0]['k']
    for i in range(1, len2):
        ids, k, r = map(int, lines2[i + 1].split(' '))
        details.append({'k': k, 'r': r})
        h_n -= details[i]['k']

        if details[i-1]['r'] < details[i]['r'] or h_n < 0:
            return False
    return True

```

РЕШЕНИЕ:

Это самая сложная из задач, она подразумевает написание жадного алгоритма. Для этого сортируем детали в порядке увеличения высоты детали.

Далее выбираем все детали от меньшей высоты к большей и проверяем, что у нас ещё не было детали такого радиуса.

Полученный набор деталей сортируем в порядке увеличения радиуса и выводим на экран.

Пример программы на языке Python, реализующей этот алгоритм:

```

1. from operator import itemgetter
2.
3. n, h = map(int, input().split())
4. details = []
5. for i in range(n):
6.     ids, k, r = map(int, input().split(' '))
7.     details.append({'id': ids, 'k': k, 'r': r})
8.

```

```
9. new_list = sorted(details, key=itemgetter('k', 'r'))
10. n_h = h
11. used_detail_r = set()
12. results = []
13. for detail in new_list:
14.     if n_h < detail['k']:
15.         break
16.     if n_h >= detail['k'] and detail['r'] not in used_detail_r:
17.         n_h -= detail['k']
18.         used_detail_r.add(detail['r'])
19.         results.append(detail)
20. new_res = sorted(results, key=itemgetter('r'))
21. res = []
22. for i in new_res[::-1]:
23.     res.append("{} {} {}".format(i['id'], i['k'], i['r']))
24. print("{} {} \n{}".format(len(new_res), h, '\n'.join(res)))
```