2021-2022 учебный год

Работа участника
ol2248217

№	Кратко суть задачи	Решение (есть или нет)	оценка	комментарий
1	Круговая дорога со столбами		6	Верно. Очень оригинальное решение! (+1 балл)
2	Ракетное топливо	нет	0	Решение отсутствует
3	Пуля попадает в шарик		4	допущен ряд незначительных ошибок
4	Кирпичи		5	Верно.
5	Квадрокоптер = осенний лист		5	Верно.

Файл прикреплен

Вопрос **1**

Выполнен

Балл: 5

Отметить вопрос

Редактировать вопрос

На круговой дороге расставлено 20 столбов, расстояние между которыми одинаково. Автомобиль проезжает расстояние между двумя столбами за 1 минуту, а пешеход проходит это же расстояние за 9 минут. Пусть от одного из столбов автомобиль и пешеход отправились одновременно, но в разные стороны. Сколько раз автомобиль и пешеход окажутся возле каких-нибудь столбов одновременно перед тем, как пешеход вернется к первому столбу?

решение загружено в файле и прикреплено к задаче №1

№1 Т.к. столбы находятся на одинаковом расстоянии (~~последовательно идущие~~), длины дуг, стягиваемых соседними столбами, равны. Тогда равны и центральные углы, на эти дуги опирающиеся. т.е. каждый центральный угол равен $\frac{360^\circ}{20} = 18^\circ$.

Пусть ϑ - скорость пешехода, V - автомобиля, тогда $\frac{V}{\vartheta} = \frac{\Delta t_{\text{пеш}} \cdot \cancel{d}}{\Delta t_{\text{авт}} \cdot \cancel{d}} = 9$, т.е. $V = 9\vartheta$

(d - длина дуги в 18°)

Тогда каждые 9 минут пешеход перемещается на 1 столб (угол 18°), а автомобиль - 9 столбов (угол 162°). Так как пешеход и автомобиль идут в разные стороны, ~~каждые~~ через 9 минут (после начала пути) они должны расходиться на 10 столбов (угол 180°), т.е. находиться на диаметрально противоположных точках круговой дороги. Тогда через следующие 9 минут они окажутся около одного столба. Следовательно, каждые 18 минут они встречаются.

т.к. пешеход вернётся к первому столбу через $20 \cdot 9 = 180$ мин после начала пути, они с автомобилем успеют встретиться $\frac{180}{18} = 10$ раз. Имеется в виду, что 10-ая встреча произойдёт к тому моменту, как пешеход окажется у первого столба снова. Ответ: если момент достижения пешеходом первого столба после начала пути учитывается - 10, иначе - 9.

Вопрос **2**

Выполнен

Баллов: 0,0 из 5,0

Отметить вопрос

Редактировать вопрос

Для того чтобы выйти на круговую орбиту Земли высотой 250 км, искусственный спутник должен набрать скорость равную как минимум 8360 м/с. Предполагается, что для запуска спутника будет использована двухступенчатая ракета-носитель с жидкостным ракетным двигателем. Пусть масса второй ступени такой ракеты равна 60 тонн, а масса спутника (то есть полезной нагрузки) равна 15 тонн. Также известно, что масса топлива составляет 88 % от массы каждой ступени.

4) Выберите подходящее ракетное топливо (пару горючее-окислитель) и рассчитайте общую массу данной ракеты. При расчетах используйте формулу Циолковского (отдельно для каждой ступени, считая, что полное сгорание топлива в любой ступени позволяет набрать половину от необходимой скорости):

$$\boldsymbol{V} = \boldsymbol{I} \times \ln(\boldsymbol{M_1}/\boldsymbol{M_2}),$$

где $\boldsymbol{V}$ — конечная скорость летательного аппарата;
$\boldsymbol{I}$ — удельный импульс ракетного двигателя (отношение тяги двигателя к секундному расходу массы топлива);
$\boldsymbol{M_1}$ — начальная масса летательного аппарата (полезная нагрузка + конструкция аппарата + топливо);
$\boldsymbol{M_2}$ — конечная масса летательного аппарата (полезная нагрузка + конструкция аппарата).

Характеристики пар двухкомпонентного топлива

Номер топлива	Окислитель	Горючее	Удельный импульс, м/с
1	Кислород	Водород	4194,4
2	Кислород	Керосин ($C_{10}H_{22}$)	3283,0
3	Кислород	Несимметричный диметилгидразин	3371,2
4	Кислород	Гидразин	3390,8
5	Кислород	Аммиак	3165,4
6	Тетраоксиддиазота	Керосин($C_{10}H_{22}$)	3028,2
7	Тетраоксиддиазота	Несимметричный диметилгидразин	3116,4
8	Тетраоксиддиазота	Гидразин	3155,6
9	Фтор	Водород	4400,2
10	Фтор	Гидразин	3939,6
11	Фтор	Пентаборан (B_5H_9)	3537,8

2) Напишите уравнения реакций горения каждого топлива из таблицы.

решения нет

Вопрос 3

Выполнен

Балл: 5

Отметить вопрос

Редактировать вопрос

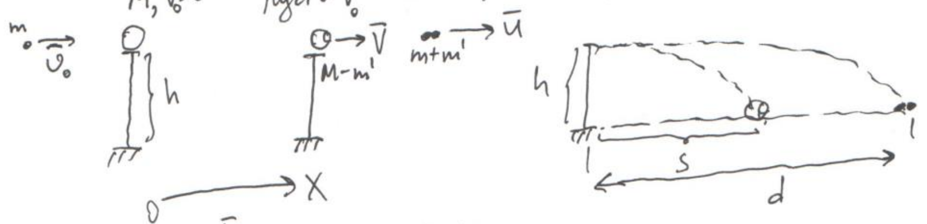
Небольшой шарик массой M покоится на вертикальном столбике высотой h. В центр шарика стреляют из ружья пулей массой m так, что она попадает в него, летя горизонтально со скоростью v_0. Пуля пробивает шарик насквозь и уносит с собой некоторую массу m' ($m' < M$), отстреленную у шарика. Определите расстояние d, на котором пуля и прилипшая к ней отстреленная масса шарика коснутся земли, если известно, что шарик упал на землю на расстоянии s от столбика? Сопротивлением воздуха и трением шарика о поверхность столбика пренебречь.

решение загружено в файле и прикреплено к задаче №1

№3
Дано: M, m, ϑ_0, S, m'

Найти: d

Пусть V_0 - нач. скорость шарика. $V_0 = 0$



$\bar{V}$ - скорость шарика после пробития, $\bar{u}$ - скорость пули и куска шарика после отстрела.

По закону сохранения импульса:

$$OX: \quad m\vartheta_0 = (M-m')V + (m+m')u \quad (1)$$

После отстрела и шарик, и пуля с куском свободно падают с высоты h, при этом двигаясь равномерно вдоль оси OX (принцип суперпозиции).

Тогда для шарика: $h = \frac{gt_1^2}{2} \Rightarrow t_1 = \sqrt{\frac{2h}{g}}$; $\quad S = Vt_1 \Rightarrow V = \frac{S}{t_1} = S\sqrt{\frac{g}{2h}} \quad (2)$

$$(2)\to(1): \quad m\vartheta_0 = (M-m')S\sqrt{\frac{g}{2h}} + (m+m')u$$

$$u = \frac{m\vartheta_0 - (M-m')S\sqrt{\frac{g}{2h}}}{m+m'}$$

$d = ut_2 = ut_1$ ($t_1 = t_2$, т.к. равны высоты, с которых падают шарик и пуля с куском)

$$d = \sqrt{\frac{2h}{g}} \cdot \frac{m\vartheta_0 - (M-m')S\sqrt{\frac{g}{2h}}}{m+m'} = \frac{1}{m+m'}\left(m\vartheta_0\sqrt{\frac{2h}{g}} - (M-m')\cdot S\right)$$

$$d = \frac{1}{m+m'}\left(\sqrt{\frac{2h}{g}}\cdot m\vartheta_0 - S\,(M-m')\right)$$

$\left\{ [d] = 1\cdot\frac{1}{кг}\left(\frac{м\cdot с^2}{м}\cdot\frac{м}{с^2}\cdot кг - \cancel{м\cdot кг}\right) = 1 м \right\}$

Ответ: $d = \frac{1}{m+m'}\left(\sqrt{\frac{2h}{g}}\, m\vartheta_0 - S\,(M-m')\right)$

Вопрос **4**

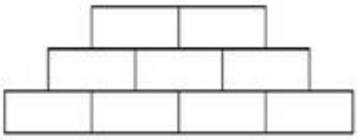
Выполнен

Балл: 5

Отметить вопрос

Редактировать вопрос

У Севы имеется всего N одинаковых кирпичей, из которых он должен сложить идеальную стену. Стена считается идеальной только в том случае, если ее основание — это ряд из m штук целых кирпичей, приставленных друг к другу торцами, а в каждом последующем верхнем слое кирпичей ровно на один меньше, чем в предыдущем нижнем (см. пример идеальной стены из $N = 9$ кирпичей на рисунке). Идеальные стены какой высоты H сможет сложить Вася так, чтобы у него не осталось лишних кирпичей, если $N = 1023$, а высота кирпича равна 7 см? Проведите аналитическое решение задачи и составьте компьютерную программу для получения ответа.



решение загружено в файле и прикреплено к задаче №1

№ 4 Заметим, что числа кирпичей в каждом слое составляют арифметическую прогрессию с разностью 1. Т.е. если на верхнем слое i кирпичей и всего слоёв $i+n-1$, в основании будет ~~$i+n-1$~~ n кирпичей, а всего в стене будет $\sum_{k=i}^{n} k$ кирпичей (сумма чисел от i до n). Известно, что $\sum_{j=1}^{n} j = \frac{n(n+1)}{2}$.

Тогда
$$\sum_{k=i}^{n} k = \sum_{j=1}^{n} j - \sum_{q=1}^{i-1} q = \frac{n(n+1)}{2} - \frac{(i-1)i}{2} = \frac{n^2+n-i^2+i}{2} = \frac{(n-i)(n+i)+(n+i)}{2} =$$
$$= \frac{(n+i)(1+n-i)}{2} = N \text{ по условию}$$

$$\underset{\in \mathbb{Z},>0}{(n+i)}\underset{\in \mathbb{Z},>0}{(1+n-i)} = 2N$$

для нахождения n и i можно перебрать делители в разложении числа $2N$. При этом $1+n-i$ – число слоёв в стене.

Тогда $H = h\cdot(1+n-i)$ h - высота кирпича

Имеет место система $\begin{cases} n+i=d_1 \\ 1+n-i=d_2 \end{cases}$, откуда $\begin{cases} n = \frac{d_1+d_2-1}{2} \\ i = \frac{d_1-d_2+1}{2} \end{cases}$ при $d_1=d_2$ $i=\frac{1}{2}$ - неуд. т.е. $d_1>d_2$, $\sqrt{2N} > d_2 \geq 1$

d_1 и d_2 - делители $2N$

$\Rightarrow$ $H = h\cdot d_2$, т.е. можно взять любой делитель числа $2N$ меньше $\sqrt{2N}$

Для $N=1023$ и $h=7$ см: $(n+i)(1+n-i) = 2046 = 2\cdot 1023 = 2\cdot 3\cdot 341\cdot 1$

$H = 7\cdot 1 = 7$ см $(1023 = N)$

Подойдут делители 1, 2, 3, 6.

$H = 7\cdot 2 = 14$ см $(511+512=N)$

$H = 7\cdot 3 = 21$ см $(340+341+342=N)$

~~$H = 7\cdot 341 = 2387$ см $(340+341+342=N)$~~

Тогда $H = \{7 \text{ см}; 14 \text{ см}; 21 \text{ см}; 42 \text{ см}\}$.

$H = 7\cdot 6 = 42$ см $(168+169+170+171+172+173=N)$

~~$H = 7\cdot 682$~~

```
#include <iostream>
int main(){
    int N, h, d(1);
    std::cin>>N>>h; N*=2;
    while(d*d<N){ //перебор до корня из 2N
        if(N%d==0) //d- делитель 2N
            cout<<h*d<<endl;
        d++;
    } return 0;
}
```

— программа на C++

Вопрос **5**

Выполнен

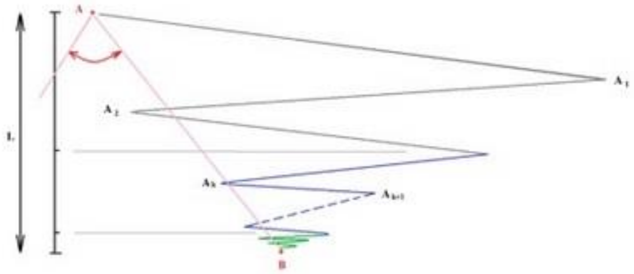
Баллов: 0,0 из 5,0

Отметить вопрос

Редактировать вопрос

На квадрокоптере установлен датчик, который смотрит вертикально вниз и сканирует поверхность на наличие точки B. Угол обзора датчика 30 градусов. Квадрокоптер летит горизонтально до тех пор пока на границе угла обзора датчика не попадает точка B. В этот момент он начинает снижаться.

Алгоритм автоматической посадки квадрокоптера с высоты L в определенную точку B можно описать следующим образом (все участки пути прямолинейные и проходят точно над точкой B):



1. Когда в угол обзора датчика попадает точка B (она оказывается на границе угла обзора) то он начинает прямолинейное снижение под углом 20 градусов к горизонту со скоростью 10 м/с, проходит точно над точкой B и продолжает движение до тех пор пока точка B не окажется на границе угла обзора датчика с противоположной стороны.
2. В тот момент, когда точка B покидает угол обзора датчика, попадая на его границу, квадрокоптер мгновенно останавливается и начинает снижение в противоположную сторону (т.е. в сторону точки B), сохраняя прежние угол (20 градусов) и скорость (10 м/с).
3. Специальный датчик отслеживает все точки изменения траектории (A_1, A_2, …, A_k, A_{k+1}, ….), а бортовой вычислитель считает время прохождения между двумя последними точками. В том случае, если время между двумя последними пройденными точками изменения траектории становится менее 5 секунд, квадрокоптер меняет и угол снижения (теперь он становится 10 градусов), и свою скорость движения (5 м/с).
4. Квадрокоптер продолжает снижение с новыми параметрами и с прежним алгоритмом до тех пор пока время прохождения между двумя точками не станет менее 3 секунд. В этот момент опять меняется угол (становится равным 5 градусам) и скорость (3 м/с).
5. Процесс снижения продолжается до тех пор пока время прохождения между двумя точками не станет менее 2 секунд. В этот момент квадрокоптер начинает вертикальное снижение до высоты 0 метров со скоростью 0,5 м/с.

Необходимо написать программу, реализующую вычисление основных параметров снижения (для L=1000м):

А) на какой высоте от земли произойдет каждое изменение скорости (описанное в пунктах 3, 4 и 5).

Б) Сколько потребует времени для снижения квадрокоптера из точки A в точку B по данному алгоритму? Как далеко от точки B приземлится квадрокоптер?

В) До какого интервала времени (прохождения между двумя точками) надо продолжать выполнение п.4, чтобы при переходе к п.5 отклонение приземлившегося аппарата от точки B было менее 1 метра?

Примечание: программа должна содержать комментарии, обеспечивающие понимание алгоритма работы программы.

решение загружено в файле и прикреплено к задаче №1

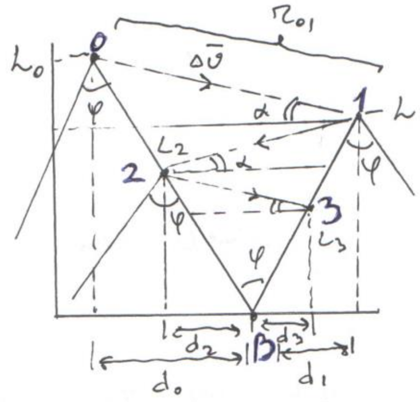
```
//решение представлено на языке С++
#include <iostream>
#include <vector>

using namespace std;
const double EPS=1e-8;
int main(){
	double l(1000),l1,d,d1,phi(30),a(20),v(10),t, time(0), tau(0);
t=tan(phi/2);
d=l*t;
while(tau>=5){
l1=l*(1-tan(a)*t)/(1+tan(a)*t);
tau=t*(l+l1)/(v*cos(a));
l=l1;
d=l*t;
time+=tau;
}
cout<<l<<endl;
a=10, v=5;
while(tau>=3){
l1=l*(1-tan(a)*t)/(1+tan(a)*t);
tau=t*(l+l1)/(v*cos(a));
l=l1;
d=l*t;
time+=tau;
}
cout<<l<<endl;
a=5, v=3;
double A=5, V=3, L=l, D=d, Tau=tau, Time=time;
while(D>=1){
L1=L*(1-tan(A)*t)/(1+tan(A)*t);
Tau=t*(L+L1)/(V*cos(A));
L=L1;
D=L*t;
Time+=Tau;
}

while(tau>=2){
l1=l*(1-tan(a)*t)/(1+tan(a)*t);
tau=t*(l+l1)/(v*cos(a));
l=l1;
d=l*t;
time+=tau;
}
cout<<l<<endl;
cout<<time<<endl;
cout<<d<<endl;
cout<<Tau<<endl;

return 0;
}
```

№ 5



$\varphi = 30^\circ$

$L_0 = L = 1000\text{м}$

$\tau_0 = 0$

L - высота точки от Земли,
d - расстояние от проекции точки на Землю до B, τ - время полёта от пред. точки к новой

$$d_0 = L_0 \operatorname{tg}\frac{\varphi}{2}$$

$$L_0 - L_1 = (d_0 + d_1)\operatorname{tg}\alpha, \quad d_1 = L_1 \operatorname{tg}\varphi/2$$

$$L_0 - L_1 = d_0 \operatorname{tg}\alpha + L_1 \operatorname{tg}\frac{\varphi}{2}\operatorname{tg}\alpha$$

$$L_1 = \frac{-d_0 \operatorname{tg}\alpha + L_0}{1 + \operatorname{tg}\frac{\varphi}{2}\operatorname{tg}\alpha} = \frac{L_0 - L_0 \operatorname{tg}\frac{\varphi}{2}\operatorname{tg}\alpha}{1 + \operatorname{tg}\frac{\varphi}{2}\operatorname{tg}\alpha} = L_0 \frac{1 - \operatorname{tg}\frac{\varphi}{2}\operatorname{tg}\alpha}{1 + \operatorname{tg}\frac{\varphi}{2}\operatorname{tg}\alpha}$$

$$\cos\alpha = \frac{d_0 + d_1}{r_{01}} \Rightarrow r_{01} = \frac{d_0 + d_1}{\cos\alpha}$$

$$\tau_1 = \frac{r_1}{\Delta\upsilon} = \frac{d_0 + d_1}{\Delta\upsilon\cos\alpha} = \frac{(L_0 + L_1)\operatorname{tg}\frac{\varphi}{2}}{\Delta\upsilon\cos\alpha}$$

Обобщим для точки $i > 0$:
$$\begin{cases} L_i = L_{i-1}\left(\dfrac{1 - \operatorname{tg}\alpha \operatorname{tg}\frac{\varphi}{2}}{1 + \operatorname{tg}\alpha \operatorname{tg}\frac{\varphi}{2}}\right) \\ d_i = L_i \operatorname{tg}\frac{\varphi}{2} \\ \tau_i = \operatorname{tg}\frac{\varphi}{2} \cdot \dfrac{L_{i-1} + L_i}{\Delta\upsilon\cos\alpha} \end{cases}$$

Если $\operatorname{tg}\frac{\varphi}{2} = \operatorname{tg}15^\circ = t$,
то
$$\begin{cases} L_i = L_{i-1} \cdot \dfrac{1 - \operatorname{tg}\alpha\, t}{1 + \operatorname{tg}\alpha\, t} \\ d_i = L_i t \\ \tau_i = t \cdot \dfrac{L_{i-1} + L_i}{\Delta\upsilon\cos\alpha} \end{cases}$$

~~#include <iostream> // решение на C++~~
~~int main() {~~
~~double L(1000), L1, d, d1, phi(30), a(20), v(10), t,~~
~~time(0), tau(0);~~

Решение на C++ см. на стр. 5

№5 (код)

```
#include <iostream>
using namespace std;
int main() {
 double l(1000), l1, d, d1, phi(30), a(20), V(10), t, time(10), tau(0);
  t = tan(phi/2);
  d = l * t;  //d[0]
 while(tau>=5) {   // промежуток времени не менее 5 сек
   l1 = l * (1-tan(a)*t)/(1+tan(a)*t);
   tau = t *(l+l1)/(V*cos(a));
    l=l1;
   d = l*t;
   time += tau; //общее время
 }
 cout << l << endl; // смена скорости
   a=10, V=5;
while(tau>=3) {
  l1=l*(1-tan(a)*t)/(1+tan(a)*t);
  tau = t*(l+l1)/(V*cos(a));
  l=l1;
   d=l*t;
   time += tau;
 }
 cout << l << endl;
  a=5, V=3;
 double A=5, V=3, L=l, D=d, Tau=tau, Time=time  //для п.B)
while(D>=1) {
   L1=L*(1-tan(A)*t)/(1+tan(A)*t);
     Tau = t*(L+L1)/(V*cos(A));
     L=L1; D = L*t; Time += Tau;
  }
  // п.B) решён
While(tau>=2) {
        l1=l*(1-tan(a)*t)/(1+tan(a)*t);
         tau = t*(l+l1)/(V*cos(a));
           l=l1; d = l*t; time += tau; }
 cout << l << endl; cout << time << endl; cout << d << endl; cout << Tau << endl;
                              // п.Б)                        // п.B)
return 0;
}
```