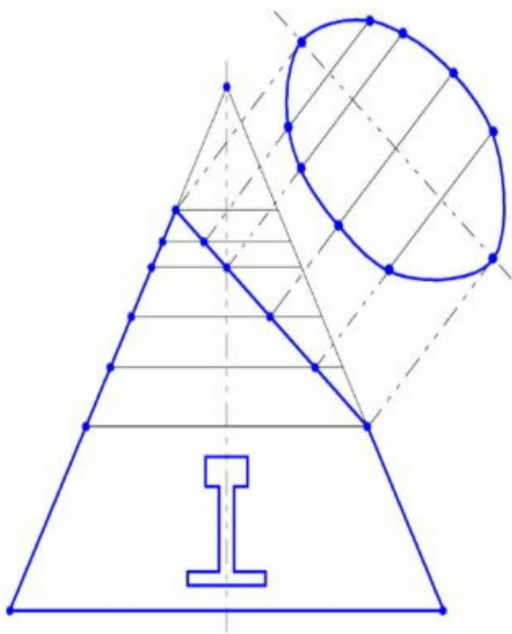


Сергей Гаврилов
Python-3
полезные программы
книга вторая



Сергей Фёдорович Гаврилов

Python-3. Полезные программы. Книга вторая

http://www.litres.ru/pages/biblio_book/?art=51798248

SelfPub; 2020

Аннотация

Эта книга для начинающих программистов, желающих начать программировать на языке Python 3. Книга будет полезна для студентов механиков и для цеховых конструкторов-механиков, а также для разметчиков. Предлагаемые в книге расчеты экономят рабочее время и уменьшают количество ошибок в расчетах. Язык программирования Python хорошо подходит для быстрого решения задач по определению оптимальных размеров и параметров. Программирование на Python требует меньше времени, чем на других языках. Конструкторам в цеху часто приходится быстро выполнять разнообразные расчеты – библиотека небольших программ позволяет значительно экономить время и уменьшить вероятность ошибок. Данная книга – сборник листингов рабочих программ. Все программы, в разное время, разработаны самим автором при возникновении необходимости данных расчетов. Все программы проверены и постоянно используются в работе конструктора-механика.

Листинги удобно использовать в качестве готовых блоков для вновь разрабатываемых программ.

Содержание

Введение.	6
Примечание:	7
Написание чисел с дробной десятичной частью.	10
Написание комментариев.	11
Линейная интерполяция	12
Расчет параметров численным интегрированием	17
Координаты точек на радиусной кривой	21
Хорда – Прогиб – Радиус	28
Расчет по Хорде – Радиусу	32
Расчет по Хорде и углу раствора	34
Вывод по Хорде Прогиб Радиус	36
Координаты радиусной кривой	39
Расчет по Хорде и длине Дуги	40
Расчет по Хорде – Прогибу	43
Расчет по Радиусу и Углу раствора	46
Расчет по Радиусу и Прогибу	48
Расчет по Радиусу и длине Дуги	50
Расчет по Прогибу и длине Дуги	52
Расчет по Прогибу и Углу раствора	55
Расчет по Углу раствора и длине Дуги	57
Вывод по Хорде Прогиб Радиус	59
Расчет Правильного многогранника	62
Овал (коробовая кривая)	68

Геометрия многогранника	76
Треугольник	83
Известны три стороны	86
Две известные стороны и угол между ними	89
Одна сторона – A и два угла B и C	92
Центр масс нескольких тел	100
Конец ознакомительного фрагмента.	103

Введение.

Книга вторая.

Инженер конструктор отдела гл.механика Механического завода Гаврилов Сергей Федорович написал эту книгу для начинающих программистов, желающих начать программировать на языке Python 3. Книга будет полезна для студентов механиков и для конструкторов механиков а также дл разметчиков. Данная книга содержит набор небольших рабочих программ облегчающих геометрические расчеты встречающиеся в практической работе цехового конструктора и разметчика. Предлагаемые в книге расчеты существенно экономят время и уменьшают вероятность ошибок. Все программы проверены и постоянно используются в работе конструктора механика. Листинги программ снабжены пояснениями. Листинги удобно использовать в качестве готовых блоков для вновь разрабатываемых программ. Для практического применения листингов программ проверьте – установлен ли на вашем компьютере Python 3.4. или более старшая версия Python 3.8..Python 3.4 распространяется свободно и бесплатно – например на сайте <https://soft-file.ru/python/> Перед скачиванием исходника Python 3.4 проверьте разрядность вашей ЭВМ. Посмотрите вкладку – Компьютер – Свойства системы. Бывают 32 разрядные и 64 разрядные системы. Для каждой системы свой Python 3.4..

Примечание:

Программы написаны для Python 3.4 и более новых версий. На Python 2 программы работать не будут, так как в Python 3.4 написание команд отличается от Python 2.

В программе на любом языке очень важен синтаксис – порядок написания символов в строке.

Малейшая ошибка, которую человек даже не заметит, – ЭВМ замечает и отказывается исполнять программу. Поэтому проще копировать блоки уже работающих программ в свою программу, при необходимости, подправить готовые блоки легче, чем писать блок заново – так будет меньше ошибок в программе и экономится время.

Программа в Python состоит из строк, написанных в простом текстовом редакторе.

Текстовый редактор должен быть именно простым, к примеру Блокнот. Редактор Microsoft Word не годится потому, что он вставляет в строку невидимые служебные символы, которые, искажают команды Python и программа отказывается работать. Если листинг программы вы скачали в формате «.doc» или подобном – следует создать пустой текстовый файл в программе «Блокнот» – выделить и скопировать листинг из файла «.doc» и вставить в пустой текстовый файл «Блокнота» с расширением «.txt» и сохранить его с именем программы. Затем открыть этот файл и сохранить

уже с расширением «.py». Вам будут нужны оба этих файла.

Важно, строки программы должны начинаться без пробела точно с начала строки.

Если в программе имеются логические операторы, например **while** или **for**, то за таким оператором следуют строки одного или нескольких блоков. Блоки выполняются программой,

или пропускаются без исполнения, в зависимости от условия в логическом операторе.

Блоки отделяются от основной программы пробелами в начале строки. Обычно в начале строки блока ставят четыре пробела. Число пробелов в начале строки – важная величина.

Если в компьютере Python 3.4 установлен правильно, то при двойном клике мышью на файле с расширением “.py” программа начнет исполняться. Если программа стопорится или вылетает – Запускаем файл редактора программы «IDLE(Python GUI)» в этом редакторе щелкаем левой кнопкой мыши на – File – затем Open. В выпавшем меню находим свой файл, выделяем его – щелкаем – Открыть. На экран выводится листинг вашего файла с выделением цветом элементов команд. Находим колонку – Run- щелкнув – открываем выпадающее меню и щелкаем на – Run Module F5 – Ваша программа начинает выполняться и останавливается на месте ошибки. Иногда она останавливается на следующей за ошибкой команде. Выводится комментарий к ошибке на ан-

глийском. В некоторых случаях помогает удаление строки в начале программы « # -*- coding: cp1251 -*- » и программа заработает.

»Программа редактор « IDLE(Python GUI) » идет в комплекте с программой Python 3.4.. и служит для нахождения ошибок в разрабатываемой программе.

При верстке книги « умные программы » что-то убирают, а что-то добавляют, искажая оригинал.

К примеру при верстке удаляются все « лишние» пробелы. Скопировав листинг программы в «Блокноте» в формате «.txt» необходимо восстановить утраченные пробелы. Перед знаком # расположенным после команды надо добавить два пробела. В блоках- там, где блок выделяется пробелами, перед строками блока написана фраза: # Далее Сдвиг – четыре пробела в начале каждой строки. Соответственно в листинг ставим четыре пробела в начало каждой строки. Там, где блок закончился, написано: # Далее Конец Сдвига – четыре пробела в начале каждой строки – и строки начинаются без пробелов в начале строки. Иногда меняется вид кавычек – тоже не дает работать программе – искать такие ошибки лучше в «IDLE(Python GUI)». Если при копировании программа не работает: обращайтесь gsf4610@yandex.ru к автору.

Написание чисел с дробной десятичной частью.

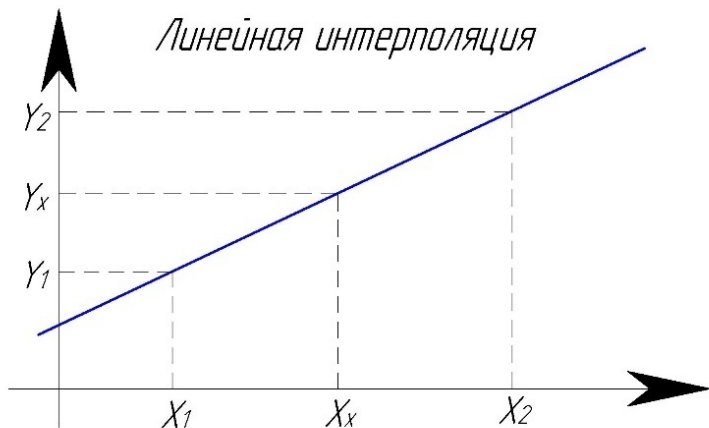
При вводе числа с дробной частью, необходимо отделять дробную часть от целой части числа точкой. Если части числа разделить запятой – программа вылетает без предупреждения.

Величины углов для расчета в программе необходимо задавать в радианах. Если программа в результат расчета выводит величину угла – то он выводится в радианах. Ответ на экран удобнее выводить в градусах – при выводе радианы переводят в градусы. При вводе угловых величин- их обычно вводят в градусах и, перед подачей в расчетную часть, переводят в радианы.

Написание комментариев.

Значок # предваряет начало комментариев. То, что следует за значком программа просто пропускает, переходя на следующую строку. Комментарий программист пишет себе для справки, чтобы потом ему или другому программисту было легче разобраться в работе программы.

Линейная интерполяция



Линейную интерполяцию применяем для определения значения коэффициента находящегося между двумя табличными значениями.

Листинг программы.

```
# -*- coding: cp1251 -*-
```

```
import math # Подключили математич модуль
```

```
# Эти первые две строки листинга определяют таблицу кодировки символов
```

и подключают библиотеку для математических расчетов.

Эти две строки, точно без изменения, применяют во всех листингах программ.

```
uu=" " # Далее идет вывод на экран заголовка программы.
```

```
print (uu)
```

```
print (uu)
```

```
u=" Расчет линейной интерполяции "
```

```
print (u)
```

```
print (uu)
```

Далее вводим в программу исходные данные для расчета.

```
print (uu)
```

```
u=" Введите X1 "
```

```
print (u)
```

```
print (uu)
```

```
x1=0.00000000
```

```
x1=input( ) # Вводим число
```

```
x1=float(x1) # Принудительно в вещественное число
```

```
print (uu)
```

```
u=" Введите X2 "
```

```
print (u)
```

```
print (uu)
```

```
x2=0.00000000
```

```
x2=input( ) # Вводим число
```

```
x2=float(x2) # Принудительно в вещественное число
```

```
print (uu)
u=" Введите Xx "
print (u)
print (uu)
xx=0.00000000
xx=input( ) # Вводим число
xx=float(xx) # Принудительно в вещественное число
print (uu)
u=" ..... "
print (u)
print (uu)
u=" Введите Y1 "
print (u)
print (uu)
y1=0.00000000
y1=input( ) # Вводим число
y1=float(y1) # Принудительно в вещественное число
print (uu)
u=" Введите Y2 "
print (u)
print (uu)
y2=0.00000000
y2=input( ) # Вводим число
y2=float(y2) # Принудительно в вещественное число
# Далее идет расчет по формуле..
ux=(((y2-y1)*(xx-x1))/(x2-x1))+y1
```

```

# Далее выводится на экран результат расчета
u=" ..... "
print (u)
print (u)
print (uu)
print (uu)
u=" X1 = ", x1," Y1 = ",y1," "
print (u)
print (uu)
u=" X2 = ", x2," Y2 = ",y2," "
print (u)
print (uu)
u=" Xx = ", xx," Yx = ",yx," "
print (u)
print (uu)
u=" ..... "
print (u)
print (u)
print (uu)
print (uu)
input( ) # Ожидание нажима Enter

```

Далее делаем еще один расчет на тех же исходных данных и выводим результаты.

```

Pii=math.pi # Вытащили число " Пи " и поместили в переменную Pii ...

```

```

cc=((x2-x1)*(x2-x1))+((y2-y1)*(y2-y1))

```

```
c=math.sqrt(cc) # Квадратный корень из " cc "  
cs=str(c) # Преобразуем число в строку  
u=" Длина линии x1;y1 – x2;y2 = "+cs  
print (uu)  
print (u)  
a=(y2-y1)/(x2-x1)  
aur=math.atan(a) # Угол aur в радианах через арс тангенс  
au=aur*180/Pii # Угол au в градусах ( переводим ).  
aus=str(au) # Преобразуем число в строку  
u=" Угол линии x1;y1 – x2;y2 с осью X = "+aus+" градус "  
print (uu) # Выводим пустую строку  
print (u) # Выводим результат – угол в градусах  
print (uu)  
print (uu)  
input( ) # Ожидание нажима Ентер что бы можно было  
прочитать результаты.  
# ..... Конец листинга программы .....
```

Расчет параметров численным интегрированием

Данная программа разработана для расчета момента инерции сложного сечения. Для расчета сечение было разбито по горизонтали на одинаковые по высоте участки (шаги). Длины участков были замеряны и внесены в таблицу. Затем запускаем программу и последовательно вводим из таблицы значения длинн участков. Программа рассчитывает момент инерции участка и площадь участка – затем суммирует эти величины. При вводе нуля на запрос длинны – программа показывает результат работы. Программу легко переделать на расчеты интеграции по другим законам.

```
# -*- coding: cp1251 -*-
```

```
# Алгоритм расчета момента инерции и площади интегрированием
```

```
import math # Подключили математич модуль
```

```
u=" Алгоритм временного расчета Jx интегрированием,"
```

```
uu=" "
```

```
uuu=" ..... " "
```

```
print (uu)
```

```
print (uu)
```

```
print (u)
```

```
print (uuu)
```

```
print (uu)
u=" Ввести Шаг вдоль оси Y-Y "
print(u)
t=0.00
t=input( ) # Вводим число
t=float(t) # Принудительно в вещественное число
print (uu)
J=0.0000
J=float(J)
Jx=0.0000
Jx=float(Jx)
S=0.0000
S=float(S)
Ss=0.0000
Ss=float(Ss)
d=0.00000001
d=float(d)
L=0.0
L=float(L)
n=1
n=float(n)
while d>0:
# Далее строки начинаются с четырех пробелов.
L=L+t # Суммируем смещение по оси
ss=str(n) # Преобразуем число в строку
Lt=str(L) # Преобразуем число в строку
```

```

u=" Ввести высоту по оси Y-Y Ввод № "
u=u+ss+" Смещение по оси = "+Lt
print(u)
d=0.00
d=input( ) # Вводим число
d=float(d) # Принудительно в вещественное число
print (uu)
J=t*d*d*d/12 # Находим момент инерции элемента
Jx=Jx+J # Суммируем J
S=t*d # Нашли площадь элемента
Ss=Ss+S # Суммируем S
n=n+1 # Счетчик числа вводов
# Далее строки начинаются без пробелов.
Pii=math.pi # Вытащили число " Пи "
# Rk=math.sqrt(a) # Квадратный корень из " a "
# sur=math.asin(x) # Угол A в радианах
# su=sur*180/Pii # Угол A в градусах
print (uu)
uuuu=" ,,,, ,,,, ,,,, ,,,, ,,,, ,,,, ,,,, ,,,, ,,,, ,,,, "
print (uuuu)
print (uu)
u=" Площадь сечения = "
ss=str(Ss) # Преобразуем число в строку
u=u+ss
print (u)
print (uu)

```

```
u=" Момент инерции по оси X-X = "  
ss=str(Jx) # Преобразуем число в строку  
u=u+ss  
print (u)  
print (uu)  
print (uu)  
u=" ..... Конец расчета ..... "  
print (u)  
print (uu)  
print (uu)  
input( ) # Ожидание нажима Ентер  
print (uu)  
# .....  
# Конец программы.
```

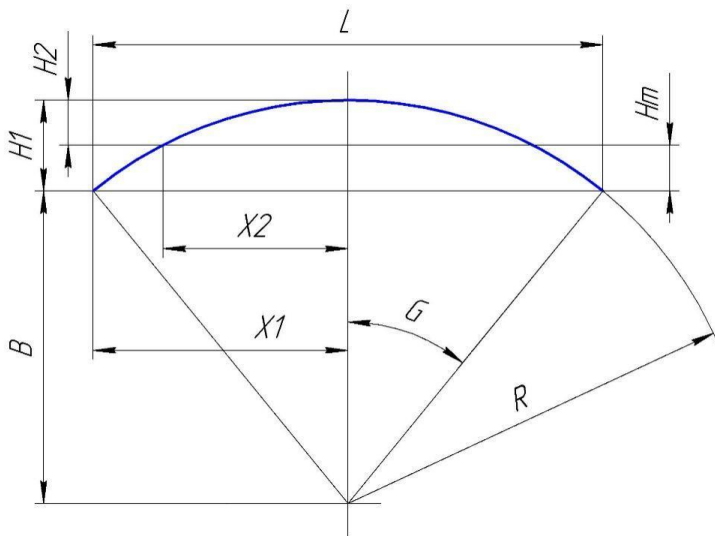
Координаты точек на радиусной кривой

Программа бывает необходима при построении сегмента с радиусом

большой величины – который построить непосредственной разметкой

сложно или невозможно из за мешающих препятствий.

Результаты расчета записываются в файл 'Rezult.txt'.



Листинг программы.

```
# -*- coding: cp1251 -*-
```

```
import math # Подключили математический модуль
```

```
import sys # Подключили основные библиотеки
```

```
# ..... .....
```

```
Pii=math.pi # Вытащили число " Пи "
```

```
f = open('Rezult.txt', 'w') # Открыли файл для записи
```

```
# Записываем числа в текстовом виде
```

```
uu=" "
```

```
u=uu+"\n" # Добавим код перевода строки
```

```
f.write(u)
```

```
u1=" Расчет координат точек на радиусной кривой "
```

```
u=u1+"\n"
```

```
f.write(u)
```

```
uu=" "+" \n"
```

```
f.write(uu)
```

```
uuu=" ..... " + "\n"
```

```
f.write(uuu)
```

```
f.write(uu)
```

```
print (uu)
```

```
u=" Построение большого радиуса методом подъема "
```

```
print (u)
```

```
print (uu)
```

```
u=" Размер по горизонтали отсчитывается от центра хор-
```

```
ды "
```

```
print (u)
```

u=" от точки X2 на хорде проводим перпендикуляр Nm
расчитанной величины. "

```
print (u)
```

u=" При вводе размера по горизонту = нулю – выход из
программы "

```
print (u)
```

```
print (uu)
```

```
input( ) # Ожидание нажима Ентер
```

```
print (uu)
```

```
a=0.00000000
```

```
y=0.00000000
```

```
ug=0.00000000
```

```
ugg=0.00000000
```

```
R=0.00000000
```

```
hm=0.00000000
```

```
ht= 0.00000000
```

```
u=" Введите радиус "
```

```
print (u)
```

```
print (uu)
```

```
R=input( ) # Вводим число
```

```
R=float(R) # Принудительно в вещественное число
```

```
u=" Радиус заданной кривой = "
```

```
ss=str(R) # Преобразуем число в строку
```

```
Rad=u+ss
```

```
u=Rad+"\n"
```

```

f.write(u)
u=" Введите Хорду L "
print (u)
print (uu)
a=input( ) # Вводим число
a=float(a) # Принудительно в вещественное число
u=" Хорда максимальная заданная L = "
ss=str(a) # Преобразуем число в строку
Xord=u+ss
u=Xord+"\n"
f.write(u)
# Находим значения максимального прогиба при  $X^2 = 0...$ 
x=((R*R)-(a*a/4))
b=R-(math.sqrt(x)) # Квадратный корень из " x "
# b – максимальный прогиб...
u=" Подъем максимальный в центре хорды = "
ss=str(b) # Преобразуем число в строку
Progi=u+ss
u=Progi+"\n"
f.write(u)
x=(a/2)/R
y=math.asin(x)
v=math.cos(y)
aur=2*y
au=aur*180/Pii # Угол A в градусах
Lx= 0.0000001 # <Начальное значение>

```

```

while Lx>0:
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
u=" Введите расстояние от центра хорды до перпендикуля-
ра X2 "
print (u)
print (uu)
Lx=input( ) # Вводим число
Lx=float(Lx)
u=" От центра хорды до точки X2 по оси X-X = "
ss=str(Lx) # Преобразуем число в строку
Xord=u+ss
u=Xord+"\n"
f.write(uu)
f.write(u)
x=(R*R)-(Lx*Lx)
z=math.sqrt(x)
y=R-z # Прогиб при хорде = Lx*2
ht=b-y # Расчитали величину подъема
u=" На расстоянии от центра = "
ss=str(Lx) # Преобразуем число в строку
u=u+ss
print (u)
print (uu)
u=" Величина подъема ( перпендикуляра ) Nm = "
ss=str(ht) # Преобразуем число в строку

```

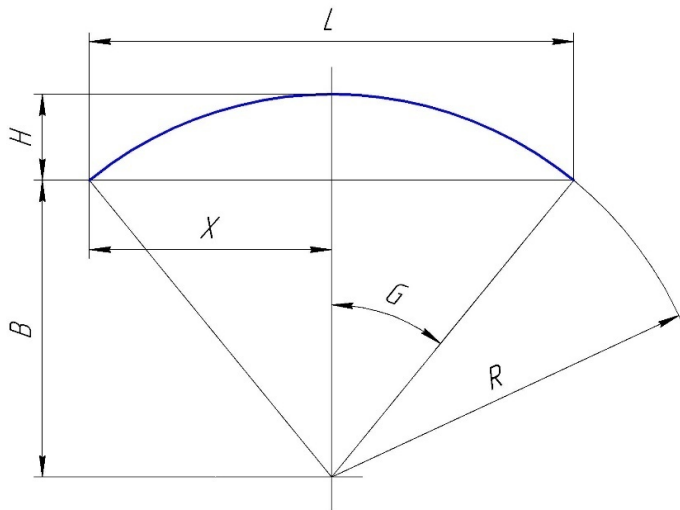
```
u=u+ss
print (u)
f.write(u)
f.write(uu)
print (uu)
u="
```

```
=====
"
print (u)
f.write(u)
print (uu)
input( ) # Ожидание нажима Enter
# Далее Конец Сдвига – четыре пробела в начале каждой
строки
print (uu)
input( ) # Ожидание нажима Enter
print (uu)
# .....
f.write(uu)
f.write(uuu)
f.close() # закрыли файл
# .....
u=" ..... Конец программы ..... "
print (u)
print (uu)
input( ) # Ожидание нажима Enter – Позволяет рассмот-
```

реть результаты расчета

..... Конец листинга программы ..

Хорда – Прогиб – Радиус



В этой программе объединены общим меню двенадцать небольших программ. Производится расчет при разном наборе исходных данных. Сейчас все привыкли к графическому интерфейсу – пункт меню вычирают мышкой и щелкают кнопкой. Здесь применен древний интерфейс: Пункты меню пронумерованы цифрами (номерами). Чтобы выбрать

пункт меню – введите цифру номера пункта меню и нажмите Enter.

Листинг программы.

```
# -*- coding: cp1251 -*-
```

```
import math # Подключили математич модуль
```

```
global Rad # Радиус – Объявили глобальную переменную
```

```
global Diam # Диаметр
```

```
global Xord # Хорда
```

```
global Progi # Прогиб
```

```
global UgSe # Угол раствора радиусов на хорду
```

```
global Dug # Длина дуги над хордой
```

```
global PISe # Площадь сектора на хорду
```

```
global PITr # Площадь Треугольника под хордой
```

```
global PIGo # Площадь Горбушки
```

```
u=" "
```

```
u=" Расчет элементов Хорда – Прогиб – Радиус "
```

```
print (uu)
```

```
print (u)
```

```
print (uu)
```

```
u=" ..... " "
```

```
print (u)
```

```
Pii=0.000000
```

```
Pii=math.pi # Вытащили число " Пи "
```

```
Kvz=0 # Флаг – Если =1 то работа уже выполнена.
```

```
print (uu)
```

```
u=" 1 – Расчет по Хорде – Радиусу "
```

print (u)

print (uu)

u=" 2 – Расчет по Хорде и Углу раствора "

print (u)

print (uu)

u=" 3 – Расчет координат Радиусной кривой "

print (u)

print (uu)

u=" 4 – Расчет по Хорде и длине Дуги "

print (u)

print (uu)

u=" 5 – Расчет по Хорде и Прогибу "

print (u)

print (uu)

u=" 6 – Расчет по Радиусу и Углу раствора "

print (u)

print (uu)

u=" 7 – Расчет по Радиусу и Прогибу "

print (u)

print (uu)

u=" 8 – Расчет по Радиусу и длине Дуги "

print (u)

print (uu)

u=" 9 – Расчет по Прогибу и длине Дуги "

print (u)

print (uu)

```
u=" 10 – Расчет по Прогибу и Углу раствора "  
print (u)  
print (uu)  
u=" 11 – Расчет по Углу раствора и длине Дуги "  
print (u)  
print (uu)  
u=" 12 – Расчет Правильного многогранника "  
print (u)  
print (uu)  
u=" ..... " "  
print (u)  
q=0  
q=input( ) # Вводим число  
q=float(q) # Принудительно в вещественное число  
# .....  
if q==1:
```

Расчет по Хорде – Радиусу

```
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
print (uu)
u=" Расчет по Хорде – Радиусу "
print (u)
print (uu)
u=" Введите Хорду "
print (u)
print (uu)
a=0.00000000
a=input( ) # Вводим число
a=float(a) # Принудительно в вещественное число
u=" Введите Радиус "
print (u)
print (uu)
R=0.00000000
R=input( ) # Вводим число
R=float(R) # Принудительно в вещественное число
x=0.00000000
y=0.00000000
v=0.00000000
Sk=0.00000000
St=0.00000000
```

S=0.000000000

x=((R*R)-(a*a/4))

b=R-(math.sqrt(x)) # Квадратный корень из " x "

x=(a/2)/R

y=math.asin(x)

v=math.cos(y)

aur=2*y

au=aur*180/Pii # Угол A в градусах

Kvz=1 # Коэф – Расчет уже выполнен

Далее Конец Сдвига – четыре пробела в начале каждой

строки

else:

Далее Сдвиг – четыре пробела в начале каждой строки

Kvz=0

.....

Далее Конец Сдвига – четыре пробела в начале каждой

строки

if q==2:

Расчет по Хорде и углу раствора

Угол раствора это угол между двумя радиусами идущими к концам хорды.

Далее Сдвиг – четыре пробела в начале каждой строки

```
print (uu)
```

```
print (uu)
```

```
u=" Расчет по Хорде и углу раствора "
```

```
print (u)
```

```
print (uu)
```

```
u=" Введите Хорду "
```

```
print (u)
```

```
print (uu)
```

```
a=0.00000000
```

```
a=input( ) # Вводим число
```

```
a=float(a) # Принудительно в вещественное число
```

```
u=" Введите Угол раствора в градусах "
```

```
print (u)
```

```
print (uu)
```

```
au=0.00000000
```

```
au=input( ) # Вводим число
```

```
au=float(au) # Принудительно в вещественное число
```

```
x=0.00000000
```

```
aur=0.00000000
```

```
v=0.00000000
```

y=0.00000000

Sk=0.00000000

St=0.00000000

S=0.000000000

aur=au*Pii/180

x=math.sin(aur/2)

R=(a/2)/x # Нашли радиус

y=(R*R)-(a*a/4)

v=math.sqrt(y) # Квадратный корень из " y "

b=R-v # Нашли прогиб

Kvz=1 # Коэф – Расчет уже выполнен

Далее Конец Сдвига – четыре пробела в начале каждой

строки

else:

Далее Сдвиг – четыре пробела в начале каждой строки

Kvz=0

.....

Далее Конец Сдвига – четыре пробела в начале каждой

строки

if q<3:

Вывод по Хорде Прогиб Радиус

```
# Далее Сдвиг – четыре пробела в начале каждой строки
D=R+R
Sk=Pii*D*D*au/(4*360) # Площадь сектора круга с углом
aur
St=(a/2)*(R-b) # Площадь треугольника в секторе
S = Sk-St # Площадь горбушки
L=Pii*D*au/360 # Длина дуги
print (uu)
u="      ,,,, ,,,, ,,,, ,,,,, ,,,,, ,,,, ,,,,, ,,,,, ,,,,, "
print (u)
print (uu)
print (uu)
u=" Хорда = "
ss=str(a) # Преобразуем число в строку
u=u+ss
print (u)
Xord=u
print (uu)
u=" Стрела прогиба = "
ss=str(b) # Преобразуем число в строку
u=u+ss
print (u)
Progi=u
```

```
print (uu)
```

```
u=" Радиус = "
```

```
ss=str(R) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
Rad=u
```

```
print (uu)
```

```
u=" Диаметр = "
```

```
ss=str(D) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
Diam=u
```

```
print (uu)
```

```
u=" Угол раствора хорды = "
```

```
ss=str(au) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
UgSe=u
```

```
print (uu)
```

```
u=" Длина дуги над хордой = "
```

```
ss=str(L) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
Dug=u
```

```
print (uu)
```

```
u=" Площадь сектора = "
```

```
ss=str(Sk) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
PISe=u
```

```
print (uu)
```

```
u=" Площадь треугольника под горбушкой = "
```

```
ss=str(St) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
PITr=u
```

```
print (uu)
```

```
u=" Площадь горбушки = "
```

```
ss=str(S) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
PIGo=u
```

```
print (uu)
```

```
# Далее Конец Сдвига – четыре пробела в начале каждой
```

```
строки
```

```
# ..... .....
```

```
if q==3:
```

Координаты радиусной кривой

Далее Сдвиг – четыре пробела в начале каждой строки

q=13 # Переброс в конец программы

.....

Далее Конец Сдвига – четыре пробела в начале каждой

строки

if q==4:

Расчет по Хорде и длине Дуги

```
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
print (uu)
u=" Расчет по Хорде и длине Дуги "
print (u)
print (uu)
u=" Введите Хорду "
print (u)
print (uu)
a=0.00000000
a=input( ) # Вводим число
a=float(a) # Принудительно в вещественное число
u=" Введите длину Дуги "
print (u)
print (uu)
L=0.00000000
L=input( ) # Вводим число
L=float(L) # Принудительно в вещественное число
R=0.00000000
dx=a/200000
rt=(a/2)+dx # Начальный радиус расчета
aa=a/2 # Половина хорды
dl=Pii*rt
```

```

while dl>L: # Расчет по Хорде и длине Дуги
# Далее Сдвиг – восемь пробелов в начале каждой строки
rt=rt+dx # Текущий радиус
x=aa/rt
y=2*(math.asin(x)) # Угол А в радианах через арс синус
dl=rt*y # Текущая длина дуги
# Далее Сдвиг – четыре пробела в начале каждой строки
yg=y*180/Pii # Угол " y " в градусах
au=yg
R=rt # Нашли радиус
y=(R*R)-(aa*aa)
v=math.sqrt(y) # Квадратный корень из " y "
b=R-v # Нашли прогиб
D=R+v
Sk=Pii*D*D*yg/(4*360) # Площадь сектора круга с углом
aur
St=aa*v # Площадь треугольника в секторе
S = Sk-St # Площадь горбушки
# Далее Конец Сдвига – четыре пробела в начале каждой
строки
else:
# Далее Сдвиг – четыре пробела в начале каждой строки
Kvz=0
# Далее Конец Сдвига – четыре пробела в начале каждой
строки
# .....

```

if q==5:

Расчет по Хорде – Прогибу

```
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
u=" Расчет по Хорде – Прогибу "
print (u)
print (uu)
u=" ..... "
print (u)
print (uu)
u=" Введите Хорду "
print (u)
print (uu)
a=0.00000000
a=input( ) # Вводим число
a=float(a) # Хорда – Принудительно в вещественное число
u=" Введите Стрелу Прогиба "
print (uu)
print (u)
print (uu)
b=0.00000000
b=input( ) # Вводим число
b=float(b) # Стрела прогиба – Принудительно в веще-
ственное число
x=0.00000000
```

$y=0.00000000$

$v=0.00000000$

$R=0.00000000$

$Sk=0.00000000$

$St=0.00000000$

$S=0.0000000000$

$Ex=1.0000000$

$k=a/2$

$Rt=k+(k/100000)$

$Rd= k/100000$

$t=k/100000000$

Уточнение шаговым подбором

while $Ex > t$:

Далее Сдвиг – восемь пробелов в начале каждой строки

$Rt=Rt+Rd$

$x=((Rt*Rt)-(k*k))$

$c=\text{math.sqrt}(x)$ # Квадратный корень из " x "

$Ex=(Rt-c)-b$

$E=\text{math.abs}(x)$

Конец подбора

Далее Сдвиг – четыре пробела в начале каждой строки

$R=Rt$

$D=R+R$

$x=k/Rt$

$\text{sur}=\text{math.asin}(x)$ # Угол A в радианах

$\text{su}=\text{sur}*180/\text{Pii}$ # Угол A в градусах

$au=2*su$

$yg=au$

$L=(R+R)*\text{Pii}*au/360$

.....

Далее Конец Сдвига – четыре пробела в начале каждой

строки

if q==6:

Расчет по Радиусу и Углу раствора

```
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
u=" Расчет по Радиусу и Углу раствора "
print (u)
print (uu)
u=" ..... "
print (u)
print (uu)
u=" Введите Радиус "
print (u)
print (uu)
R=0.00000000
R=input( ) # Вводим число
R=float(R) # Радиус – Принудительно в вещественное
число
u=" Введите Угол раствора в градусах "
print (uu)
print (u)
print (uu)
yg=0.00000000
yg=input( ) # Вводим число
yg=float(yg) # Угол раствора – Принудительно в веще-
ственное число
```

```
yr=yg*Pii/180 # Угол раствора В радианах
```

```
x=0.00000000
```

```
y=0.00000000
```

```
v=0.00000000
```

```
Sk=0.00000000
```

```
St=0.00000000
```

```
S=0.000000000
```

```
Ex=1.000000
```

```
x= math.cos(yr/2)
```

```
y=R*x
```

```
b=R-y
```

```
x=(R*R)-(y*y)
```

```
c=math.sqrt(x) # Квадратный корень из " x "
```

```
a=c+c
```

```
au=yg
```

```
D=R+R
```

```
# .....
```

```
# Далее Конец Сдвига – четыре пробела в начале каждой
```

```
строки
```

```
# .....
```

```
if q==7:
```

Расчет по Радиусу и Прогибу

```
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
u=" Расчет по Радиусу и Прогибу "
print (u)
print (uu)
u=" ..... "
print (u)
print (uu)
u=" Введите Радиус "
print (u)
print (uu)
R=0.00000000
R=input( ) # Вводим число
R=float(R) # Радиус – Принудительно в вещественное
число
u=" Введите Прогиб "
print (uu)
print (u)
print (uu)
b=0.00000000
b=input( ) # Вводим число
b=float(b) # Угол раствора – Принудительно в веществен-
ное число
```

k=R-b

x=(R*R)-(k*k)

a=(math.sqrt(x))*2

x=(a/2)/k

aur = (math.atan(x))*2

au=aur*180/Pii

yg = au

.....

Далее Конец Сдвига – четыре пробела в начале каждой

строки

if q==8:

Расчет по Радиусу и длине Дуги

```
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
u=" Расчет по Радиусу и длине Дуги "
print (u)
print (uu)
u=" ..... "
print (u)
print (uu)
u=" Введите Радиус "
print (u)
print (uu)
R=0.00000000
R=input( ) # Вводим число
R=float(R) # Радиус – Принудительно в вещественное
число
print (uu)
u=" Введите длину Дуги "
print (u)
print (uu)
L=0.00000000
L=input( ) # Вводим число
L=float(L) # Хорда – Принудительно в вещественное чис-
```

$ygr = (2 * \text{Pii}) * (L / ((R + R) * \text{Pii}))$ # Угол раствора хорды ради-
ан

$x = \text{math.sin}(ygr/2)$

$y = R * x$

$a = y + y$ # Хорда

$t = ((R * R) - (y * y))$

$z = \text{math.sqrt}(t)$ # Квадратный корень из " y "

$b = R - z$ # Прогиб

$yg = ygr * 180 / \text{Pii}$ # Угол раствора хорды град.

.....

Далее Конец Сдвига – четыре пробела в начале каждой
строки

Расчет по Прогибу и длине Дуги "

if q==9:

Расчет по Прогибу и длине Дуги

```
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
u=" Расчет по Прогибу и длине Дуги "
print (u)
print (uu)
u=" ..... "
print (u)
print (uu)
u=" Введите стрелу Прогиба "
print (u)
print (uu)
b=0.00000000
b=input( ) # Вводим число
b=float(b) # Радиус – Принудительно в вещественное чис-
ло
print (uu)
u=" Введите длину Дуги "
print (u)
print (uu)
L=0.00000000
L=input( ) # Вводим число
L=float(L) # Хорда – Принудительно в вещественное чис-
```

```
print (uu)
```

```
u=" Подождите – идет расчет "
```

```
print (u)
```

```
print (uu)
```

```
bb=0.000000 # Текущий прогиб
```

```
Rt=L/Pii # Начальный Текущий радиус
```

```
xx=b/100000 # Приращение радиуса
```

```
xb=b/1000 # Допустимая ошибка..
```

```
db=10000000*b # Начальное значение ошибки
```

```
ugr=0.0000000
```

```
# Уточнение подбором
```

```
while db >xb:
```

```
# Далее Сдвиг – восемь пробелов в начале каждой строки
```

```
Rt=Rt+xx
```

```
ugr=(2*Pii)* (L/((Rt+Rt)*Pii)) # Угол раствора хорды ра-
```

```
диан
```

```
x=math.sin(ugr/2)
```

```
y=Rt*x
```

```
a=y+y # Хорда
```

```
t=((Rt*Rt)-(y*y))
```

```
z=math.sqrt(t) # Квадратный корень из " y "
```

```
bb=Rt-z # Прогиб
```

```
x=(b-bb)*(b-bb)
```

```
db=math.sqrt(x) # Квадратный корень из " x "
```

```
# Далее Сдвиг – четыре пробела в начале каждой строки
```

```
db=db+xb
```

```

xx=b/100000000 # Приращение радиуса
xb=b/1000000
# Уточнение подбором ( вторая ступень )
while db >xb:
# Далее Сдвига – восемь пробелов в начале каждой строки
Rt=Rt+xx
ygr=(2*Pii)* (L/((Rt+Rt)*Pii)) # Угол раствора хорды ра-
диан
x=math.sin(ygr/2)
y=Rt*x
a=y+y # Хорда
t=((Rt*Rt)-(y*y))
z=math.sqrt(t) # Квадратный корень из " y "
bb=Rt-z # Прогиб
x=(b-bb)*(b-bb)
db=math.sqrt(x) # Квадратный корень из " x "
# Далее Сдвига – четыре пробела в начале каждой строки
# Конец подбора .....
R=Rt
yg=ygr*180/Pii # Угол раствора хорды град.
# Далее Конец Сдвига – четыре пробела в начале каждой
строки
# .....
# .....
# Расчет по Углу раствора и Прогибу "
if q==10:

```

Расчет по Прогибу и Углу раствора

```
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
u=" Расчет по Прогибу и Углу раствора "
print (u)
print (uu)
u=" ..... "
print (u)
print (uu)
u=" Введите Угол раствора в градусах "
print (u)
print (uu)
yg=0.00000000
yg=input( ) # Вводим число
yg=float(yg) # Угол – Принудительно в вещественное чис-
ло
yr=yg*Pii/180
u=" Введите стрелу Прогиба "
print (u)
print (uu)
b=0.00000000
b=input( ) # Вводим число
b=float(b) # Радиус – Принудительно в вещественное чис-
```

ло

$z = \cos(\gamma/2)$

$R = b/(1-z)$

$x = 0.00000000$

$y = 0.00000000$

$v = 0.00000000$

$S_k = 0.00000000$

$S_t = 0.00000000$

$S = 0.000000000$

$E_x = 1.000000$

$x = \cos(\gamma/2)$

$y = R * x$

$x = (R * R) - (y * y)$

$c = \sqrt{x}$ # Квадратный корень из " x "

$a = c + c$

$au = yg$

$D = R + R$

.....

Далее Конец Сдвига – четыре пробела в начале каждой

строки

Расчет по Углу раствора и длине Дуги "

if $q == 11$:

Расчет по Углу раствора и длине Дуги

```
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
u=" Расчет по Углу раствора и длине Дуги "
print (u)
print (uu)
u=" ..... "
print (u)
print (uu)
u=" Введите Угол раствора в градусах "
print (u)
print (uu)
yg=0.00000000
yg=input( ) # Вводим число
yg=float(yg) # Радиус – Принудительно в вещественное
число
print (uu)
u=" Введите длину Дуги "
print (u)
print (uu)
L=0.00000000
L=input( ) # Вводим число
```

```

L=float(L) # Хорда – Принудительно в вещественное число
C=L*360/yr
D=C/Pii
R=D/2
x=0.00000000
y=0.00000000
v=0.00000000
Sk=0.00000000
St=0.00000000
S=0.000000000
Ex=1.000000
yr=yr*Pii/180
x= math.cos(yr/2)
y=R*x
b=R-y
z=(R*R)-(y*y)
c=math.sqrt(z) # Квадратный корень из " x "
a=c+c
# .....
# Далее Конец Сдвига – четыре пробела в начале каждой строки
# .....
if q>3: # Вывод по Хорде Прогиб Радиус
# Далее Сдвиг – четыре пробела в начале каждой строки
if q<12:

```

Вывод по Хорде Прогиб Радиус

Далее Сдвиг – восемь пробелов в начале каждой строки

D=R+R

L=D*Pii*yg/360

Sk=Pii*D*D*yg/(4*360) # Площадь сектора круга с углом

aur

St=a*(R-b)/2 # Площадь треугольника в секторе

S = Sk-St # Площадь горбушки

print (uu)

u=" ,,, ,,, ,,, ,,,, ,,,, ,,, ,,,, ,,,, ,,,, ,,,, "

print (u)

print (uu)

print (uu)

u=" Радиус = "

ss=str(R) # Преобразуем число в строку

u=u+ss

print (u)

Rad=u

print (uu)

u=" Диаметр = "

ss=str(D) # Преобразуем число в строку

u=u+ss

print (u)

Diam=u

```
print (uu)
```

```
u=" Стрела прогиба = "
```

```
ss=str(b) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
Progi=u
```

```
print (uu)
```

```
u=" Хорда = "
```

```
ss=str(a) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
Xord=u
```

```
print (uu)
```

```
u=" Угол раствора хорды = "
```

```
ss=str(yg) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
UgSe=u
```

```
print (uu)
```

```
u=" Длина дуги над хордой = "
```

```
ss=str(L) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
Dug=u
```

```
print (uu)
```

```
u=" Площадь сектора под дугой = "
```

```
ss=str(Sk) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
PISe=u
```

```
print (uu)
```

```
u=" Площадь треугольника под горбушкой = "
```

```
ss=str(St) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
PITr=u
```

```
print (uu)
```

```
u=" Площадь горбушки = "
```

```
ss=str(S) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
PIGo=u
```

```
print (uu)
```

```
# Далее Конец Сдвига – восемь пробелов в начале каждой
```

```
строки
```

```
# ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,, ,,,,,,
```

```
if q==12:
```

Расчет Правильного многогранника

```
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
u=" Расчет Правильного многогранника "
print (u)
print (uu)
u=" ..... "
print (u)
print (uu)
u=" Введите Описанный диаметр "
print (u)
u=" при вводе нуля – переход на вписанный диаметр "
print (u)
print (uu)
D=0.00000000
D=input( ) # Вводим число
D=float(D) # – Принудительно в вещественное число
x=0.00000000
y=0.00000000
v=0.00000000
Sk=0.00000000
St=0.00000000
```

```
S=0.000000000
```

```
Ex=10000000
```

```
vv=0 # Флаг расчета при заданном Описанном диаметре
```

```
vv=float(vv)
```

```
if D==0:
```

```
# Далее Сдвиг – восемь пробелов в начале каждой строки
```

```
u=" ..... " "
```

```
print (u)
```

```
print (uu)
```

```
u=" Введите Вписанный диаметр "
```

```
# Вписанный диаметр, при четном числе граней, является
```

размером под ключ.

```
print (u)
```

```
print (uu)
```

```
Sv=0.000000000
```

```
Sv=input( ) # Вводим число
```

```
Sv=float(Sv) # – Принудительно в вещественное число
```

```
u=" Введите число Граней "
```

```
print (uu)
```

```
print (u)
```

```
print (uu)
```

```
n=0.000000000
```

```
n=input( ) # Вводим число
```

```
n=float(n) # Принудительно в вещественное число
```

```
sur=2*Pii/n # Угол A в радианах
```

```
su=360/n # Угол A в градусах
```

au=su

yg=au

yr=sur

x=math.cos(sur/2) # Cos Угла A

y=Sv/2

R=y/x

D=R+R

vv=1 # Флаг расчета при заданном Вписанном диаметре
Далее Сдвиг – четыре пробела в начале каждой строки

if vv==0:

Далее Сдвиг – восемь пробелов в начале каждой строки

u=" Введите число Граний "

print (uu)

print (u)

print (uu)

n=0.00000000

n=input() # Вводим число

n=float(n) # Принудительно в вещественное число

R=D/2

sur=2*Pii/n # Угол A в радианах

su=360/n # Угол A в градусах

au=su

yg=au

yr=sur

x=math.cos(sur/2) # Cos Угла A

y=R*x

Далее Сдвиг – четыре пробела в начале каждой строки

Sh=y

Sv=y+y

b=R-y

x=(R*R)-(y*y)

c=math.sqrt(x) # Квадратный корень из " x "

a=c+c

Sm=(a*(R-b)/2)*n # Площадь многогранника

nn=0

Вывод по Многограннику

u=" Описанный диаметр = "

ss=str(D) # Преобразуем число в строку

u=u+ss

print (u)

print (uu)

u=" Число граней = "

ss=str(n) # Преобразуем число в строку

u=u+ss

print (u)

print (uu)

u=" Высота: Грань – Центр = "

ss=str(Sh) # Преобразуем число в строку

u=u+ss

print (u)

print (uu)

u=" Вписанный диаметр = "

```
ss=str(Sv) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
print (uu)
```

```
u=" Ширина грани = "
```

```
ss=str(a) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
print (uu)
```

```
u=" Площадь Многогранника = "
```

```
ss=str(Sm) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
print (uu)
```

```
# Далее Конец Сдвига – четыре пробела в начале каждой
```

```
строки
```

```
# ..... .....
```

```
if q==13:
```

```
# Координаты радиусной кривой
```

```
# Далее Сдвиг – четыре пробела в начале каждой строки
```

```
u1=" Расчет координат точек на радиусной кривой "
```

```
print (" ")
```

```
print (u1)
```

```
u1=" Программа этого расчета приведена ранее "
```

```
print (u1)
```

```
print (u)
```

```
input( ) # Ожидание нажима Ентер – Позволяет рассмотреть результаты расчета
```

```
# Далее Конец Сдвига – четыре пробела в начале каждой строки
```

```
q=111 # обход всего, что дальше, особенно записи в файл..
```

```
# .....
```

```
u=" ..... Конец программы ..... "
```

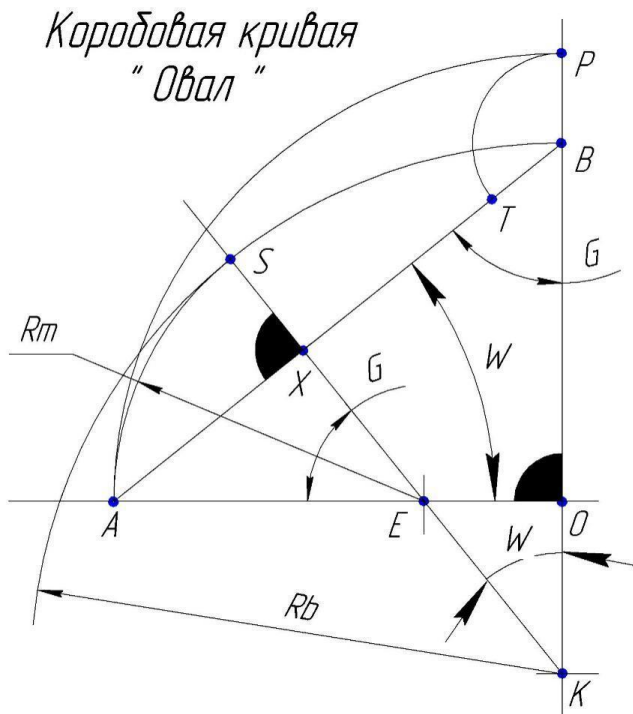
```
print (u)
```

```
print (uu)
```

```
input( ) # Ожидание нажима Ентер
```

```
# ..... Конец листинга программы .....
```

Овал (коробовая кривая)



На схеме показано построение коробовой кривой (ова-

ла).

Отрезок АО – половина большей оси овала. Отрезок ВО – половина меньшей оси овала.

Rb – Большой радиус овала. Rm – Малый радиус овала. Остальное понятно из чертежа.

Построенная коробовая кривая отличается от овала – но для большинства расчетов различие

является несущественным. Данный расчет применяется к определению формы резинового кольца при сжатии в осевом направлении.

Листинг программы.

```
# -*- coding: cp1251 -*-
```

```
import math # Подключили математич модуль
```

```
# Проверено и геометрически тоже 15-12-2015 г..
```

```
ug=0.000000
```

```
Uu=" "
```

```
u=" Расчет параметров овала ( коробовой кривой ) "
```

```
print (uu)
```

```
print (uu)
```

```
print (u)
```

```
u1=u
```

```
print (uu)
```

```
u=" .....
```

```
print (uu)
```

```
print (u)
```

```
print (uu)
```

```
u=" Вводим больший габаритный размер овала "  
print (u)  
print (uu)  
x1=0.00000000  
x1=input( ) # Вводим число  
x1=float(x1) # Принудительно в вещественное число  
u=" Вводим меньший габаритный размер овала "  
print (u)  
print (uu)  
y1=0.00000000  
y1=input( ) # Вводим число  
y1=float(y1) # Принудительно в вещественное число  
# .....  
Pii=math.pi # Вытащили число " Пи "  
oa=x1/2 # Большая полуось овала  
ob=y1/2 # Малая полуось овала  
pb=oa-ob  
tb=pb  
x=(oa*oa)+(ob*ob)  
ab=math.sqrt(x) # Квадратный корень из " x "  
at=ab-tb  
xt=at/2  
ao1=(xt*ab)/oa # Малый радиус  
bk=(ab*(xt+pb))/ob # Большой радиус  
x=ob/oa  
ua=math.atan(x) # АрксТангенс от X
```

ua – Угол четвертинки сектора большого круга
ub=(Pii/2)-ua # Угол четвертинки сектора малого круга
Sb=Pii*bk*bk # Площадь круга с Большим радиусом
Sm=Pii*ao1*ao1 # Площадь круга с Малым радиусом
Sbs=Sb*2*ua/Pii # Площадь секторов с Большим радиу-

сом

Sms=Sm*2*ub/Pii # Площадь секторов с Малым радиусом

ko=bk-ob

oo1=oa-ao1

Str=2*ko*oo1 # Площадь четырех вычитаемых треуголь-

ников

So=(Sms+Sbs)-Str

xu=ua*360/Pii # Угол раствора Б. радиусов

x=4*So/Pii

Ds=math.sqrt(x) # Диаметр равного по площади – круга

x=xu/2

Pp=((ao1*(90-x))+(bk*x))*Pii/45 # Периметр овала.

.....

u=""

print (uu)

print (u)

u2=u

print (uu)

u=" Большая ось овала = "

ss=str(x1) # Преобразуем число в строку

u=u+ss

```
print (u)
```

```
u3=u
```

```
print (uu)
```

```
u=" Меньшая ось овала = "
```

```
ss=str(y1) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
u4=u
```

```
print (uu)
```

```
u=" Большой радиус = "
```

```
ss=str(bk) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
u5=u
```

```
print (uu)
```

```
u=" От оси до центра Б. радиуса = "
```

```
ss=str(ko) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
u6=u
```

```
print (uu)
```

```
u=" Меньший радиус = "
```

```
ss=str(ao1) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
u7=u
```

```
print (uu)
```

```
u=" От оси до центра М. радиуса = "
```

```
ss=str(oo1) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
u8=u
```

```
print (uu)
```

```
u=" Угол раствора Б. радиусов = "
```

```
ss=str(xu) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
u9=u
```

```
print (uu)
```

```
u=" Периметр овала = "
```

```
ss=str(Pp) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
u10=u
```

```
print (uu)
```

```
u=" Площадь овала = "
```

```
ss=str(So) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
u11=u
```

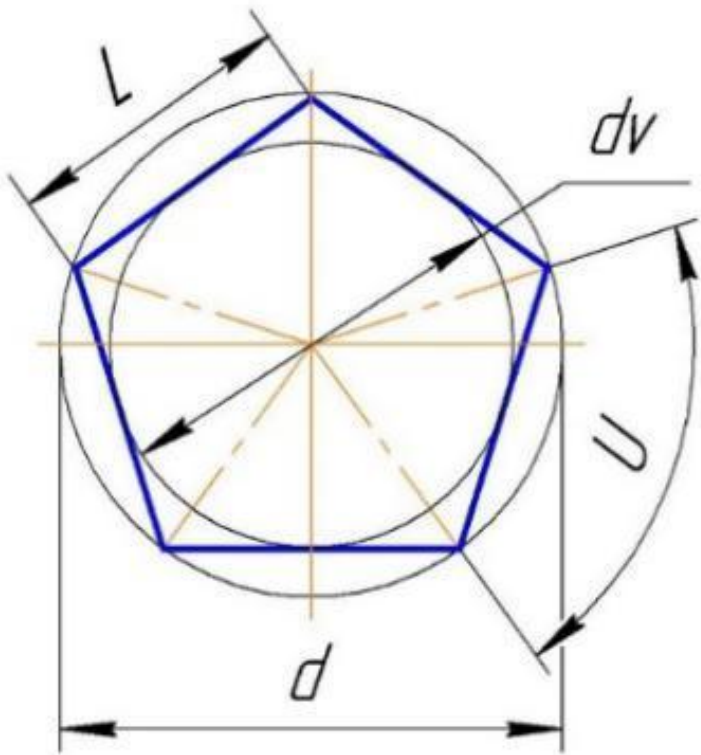
```
print (uu)
```

```
u=" Диаметр равного по площади – круга = "
```

```
ss=str(Ds) # Преобразуем число в строку
u=u+ss
print (u)
u12=u
print (uu)
print (uu)
# ..... ..... ..... .....
f = open('Rezult.txt', 'w') # Открыли файл для записи
# Записываем числа в текстовом виде
u=uu+"\n" # Добавим код перевода строки
f.write(u)
u=u1+"\n"
f.write(u)
uu=" "+ "\n"
f.write(uu)
u=u2+"\n"
f.write(u)
f.write(uu)
u=u3+"\n"
f.write(u)
u=u4+"\n"
f.write(u)
f.write(uu)
u=u5+"\n"
f.write(u)
u=u6+"\n"
```

```
f.write(u)
u=u7+"\n"
f.write(u)
u=u8+"\n"
f.write(u)
u=u9+"\n"
f.write(u)
u=u10+"\n"
f.write(u)
u=u11+"\n"
f.write(u)
u=u12+"\n"
f.write(u)
f.write(uu)
f.close() # закрыли файл
# .....
input( ) # Ожидание нажима Enter
u=" ..... Конец программы ..... "
print (u)
print (uu)
input( ) # Ожидание нажима Enter
# ..... Конец листинга программы .....
```

Геометрия многогранника



Листинг программы.

```
# -*- coding: cp1251 -*-
```

```
# Проверено 24-06-2014 г..
```

```
import sys # Подключили основные библиотеки
```

```
import math # Подключили математич модуль
```

```
import os # Подключили модуль работы с файлами
```

```
Pii=math.pi # Вытащили число " Пи "
```

```
uu=" "
```

```
print (uu)
```

```
print (uu)
```

```
print (uu)
```

```
u1=" Расчет геометрии многогранника "
```

```
print (u1)
```

```
print (uu)
```

```
u=" Введите число граней "
```

```
print (uu)
```

```
print (u)
```

```
n=0.00000
```

```
n=input( )
```

```
n=float(n) # Принудительно в вещественное число
```

```
ur=Pii/n # Половина угла на грань
```

```
u=" Введите описанный диаметр "
```

```
print (uu)
```

```
print (u)
```

```
u=" при вводе нуля – переход на вписанный диаметр "
```

```
print (uu)
```

```
print (u)
d=0.00000
dv=0.0000
d=input( )
d=float(d) # Принудительно в вещественное число
if d>0:
# Далее Сдвиг – четыре пробела в начале каждой строки
x=math.cos(ur)
r=(d/2)*x
dv=r+r
R=d/2
# Далее Конец Сдвига – четыре пробела в начале каждой
строки
if d==0:
# Далее Сдвиг – четыре пробела в начале каждой строки
u=" Введите Вписанный диаметр "
print (uu) # Вписанный диаметр при четном числе граней
это размер под ключ
print (u)
dv=input( )
dv=float(dv) # Принудительно в вещественное число
x=math.cos(ur)
R=(dv/2)/x
d=R+R
r=dv/2
# Далее Конец Сдвига – четыре пробела в начале каждой
```

строки

```
print (uu)
```

```
print (uu)
```

```
u=" .... .. " "
```

```
print (u)
```

```
print (uu)
```

```
u=" Описанный диаметр = "
```

```
t=str(d) # Преобразуем число в строку
```

```
u2=u+t # сложим строки
```

```
print (uu)
```

```
print (u2)
```

```
u=" Вписанный диаметр = "
```

```
# При четном числе граней вписанный диаметр = размеру
```

под ключ.

```
t=str(dv) # Преобразуем число в строку
```

```
u3=u+t # сложим строки
```

```
print (uu)
```

```
print (u3)
```

```
u=" Центральный угол опирающийся на грань = "
```

```
x=ur*360/Pii
```

```
t=str(x) # Преобразуем число в строку
```

```
u4=u+t # сложим строки
```

```
print (uu)
```

```
print (u4)
```

```
u=" Ширина грани = "
```

```
x=(R*R)-(r*r)
```

```
sg=2*(math.sqrt(x))
t=str(sg) # Преобразуем число в строку
u5=u+t # сложим строки
print (uu)
print (u5)
u=" Площадь многогранника = "
x=(sg*r*n)/2
t=str(x) # Преобразуем число в строку
u6=u+t # сложим строки
print (uu)
print (u6)
# .....
print (uu)
u=" ..... "
print (u)
print (uu)
print (uu)
# .....
f = open('Rezult.txt', 'w') # Открыли файл для записи
# Записываем числа в текстовом виде
u=uu+"\n" # Добавим код перевода строки
f.write(u)

u=u1+"\n" # Добавим код перевода строки
f.write(u1)
uu=" "+ "\n"
```

```
f.write(uu)
uuu=" ..... "+"\n"
f.write(uuu)
u=u2+"\n"
f.write(u)
u=u3+"\n"
f.write(u)
u=u4+"\n"
f.write(u)
f.write(uuu)
u=u5+"\n"
f.write(u)
u=u6+"\n"
f.write(u)
f.write(uuu)
f.close() # закрыли файл
input( ) # Ожидание нажима Enter
# .....
input( ) # Ожидание нажима Enter
# Контрольный расчет:
# Расчет геометрии многогранника
# .....
# Описанный диаметр = 100.0
# Вписанный диаметр = 80.90169943749474
# Центральный угол опирающийся на грань = 72.0
# .....
```

Ширина грани = 58.778525229247315

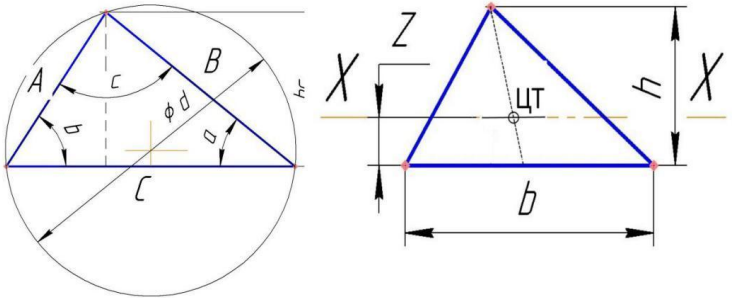
Площадь многогранника = 5944.10322684471

.....

.....

Конец листинга программы

Треугольник



Центр описанной окружности на пересечении перпендикуляров проведенных от середин сторон треугольника. Центр вписанной окружности на пересечении медиан углов треугольника. Центр тяжести треугольника лежит на линии X-X параллельной основанию и расположенной на 1/3 высоты – в точке пересечения медиан углов треугольника. Медиана делит противоположную сторону пополам. Сейчас все привыкли к графическому интерфейсу – пункт меню вычирают мышкой и щелкают кнопкой. Здесь применен древний интерфейс: Пункты меню пронумерованы цифрами (номерами). Чтобы выбрать пункт меню – введите цифру номера пункта меню и нажмите Enter.

Листинг программы.

```
# -*- coding: cp1251 -*-
```

```
import math # Подключили математич модуль.
```

```
uu=" "
```

```
u=" Расчет элементов треугольника "
```

```
print (uu)
```

```
print (uu)
```

```
print (u)
```

```
print (uu)
```

```
u=" Напротив сторон треугольника лежат одноименные
```

```
углы "
```

```
print (u)
```

```
print (uu)
```

```
u=" ..... " "
```

```
print (u)
```

```
print (uu)
```

```
Pii=0.000000
```

```
Pii=math.pi # Вытащили число " Пи "
```

```
Kvz=0 # Флаг – Если =1 то работа уже выполнена.
```

```
print (uu)
```

```
print (uu)
```

```
input( ) # Ожидание нажима Enter
```

```
print (uu)
```

```
print (uu)
```

```
u=" 1 – Известны три стороны "
```

```
print (u)
```

```
print (uu)
u=" 2 – Известны две стороны и угол меж ними "
print (u)
print (uu)
u=" 3 – Известны два угла и сторона меж ними "
print (u)
print (uu)
u=" 4 – Известны три угла "
print (u)
print (uu)
u=" ..... "
print (u)
print (uu)
q=0
q=input( ) # Вводим число
q=float(q) # Принудительно в вещественное число
```

Известны три стороны

```
if q==1: # Значит имеем три известные стороны
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
print (uu)
u=" Известны три стороны "
print (u)
print (uu)
u=" Введите сторону А "
print (u)
print (uu)
a=0.00000000
a=input( ) # Вводим число
a=float(a) # Принудительно в вещественное число
u=" Введите сторону В "
print (u)
print (uu)
b=0.00000000
b=input( ) # Вводим число
b=float(b) # Принудительно в вещественное число
u=" Введите сторону С "
print (u)
print (uu)
c=0.00000000
```

```

c=input( ) # Вводим число
c=float(c) # Принудительно в вещественное число
print (uu)
print (uu)
u=" ..... "
print (u)
print (uu)
x=0.00000000
x=((b*b)+(c*c)-(a*a))/(2*b*c)
aur=0.00000000
aur=math.acos(x) # Угол А в радианах
au=aur*180/Pii # Угол А в градусах
x=((a*a)+(c*c)-(b*b))/(2*a*c)
bur=0.00000000
bur=math.acos(x) # Угол В в радианах
bu=bur*180/Pii # Угол В в градусах
cur=Pii-(aur+bur)
cu=cur*180/Pii # Угол С в градусах
Kvz=1 # Флаг – Если =1 то работа уже выполнена.
# Далее Конец Сдвига – четыре пробела в начале каждой
строки
else:
# Далее Сдвиг – четыре пробела в начале каждой строки
Kvz=0
# Далее Конец Сдвига – четыре пробела в начале каждой
строки

```

if $q=2$:

Две известные стороны и угол между ними

```
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
print (uu)
u=" Известны две стороны и угол меж ними "
print (u)
print (uu)
u=" Введите сторону А "
print (u)
print (uu)
a=0.00000000
a=input( ) # Вводим число
a=float(a) # Принудительно в вещественное число
u=" Введите сторону В "
print (u)
print (uu)
b=0.00000000
b=input( ) # Вводим число
b=float(b) # Принудительно в вещественное число
u=" Введите угол С в градусах "
print (u)
print (uu)
```

```

cu=0.00000000
cu=input() # Вводим число
cu=float(cu) # Принудительно в вещественное число
print (uu)
print (uu)
u=" ..... "
print (u)
print (uu)
# Значит имеем две известные стороны А и В
# и угол между ними С
x=0.00000000
y=0.00000000
cur=cu*Pii/180 # Угол С в радианах
y=math.cos(cur) # COS угла С
x=((a*a)+(b*b))-(2*a*b*y)
c=math.sqrt(x) # Квадратный корень из " x "
x=((b*b)+(c*c)-(a*a))/(2*b*c)
aur=0.00000000
aur=math.acos(x) # Угол А в радианах
au=aur*180/Pii # Угол А в градусах
x=((a*a)+(c*c)-(b*b))/(2*a*c)
bur=0.00000000
bur=math.acos(x) # Угол В в радианах
bu=bur*180/Pii # Угол В в градусах
Kvz=1 # Флаг – Если =1 то работа уже выполнена.
# Далее Конец Сдвига – четыре пробела в начале каждой

```

строки

else:

Далее Сдвиг – четыре пробела в начале каждой строки

Kvz=0

Далее Конец Сдвига – четыре пробела в начале каждой строки

if q==3:

Одна сторона – А и два угла В и С

Далее Сдвиг – четыре пробела в начале каждой строки

```
print (uu)
```

```
print (uu)
```

```
u=" Известны два угла при одной стороне "
```

```
print (u)
```

```
print (uu)
```

```
u=" Введите сторону А "
```

```
print (u)
```

```
print (uu)
```

```
a=0.00000000
```

```
a=input( ) # Вводим число
```

```
a=float(a) # Принудительно в вещественное число
```

```
u=" Введите угол В в градусах "
```

```
print (u)
```

```
print (uu)
```

```
bu=0.00000000
```

```
bu=input( ) # Вводим число
```

```
bu=float(bu) # Принудительно в вещественное число
```

```
u=" Введите угол С в градусах "
```

```
print (u)
```

```
print (uu)
```

```
cu=0.00000000
```

```
cu=input( ) # Вводим число
```

```
cu=float(cu) # Принудительно в вещественное число
```

```
print (uu)
```

```
print (uu)
```

```
u=" ..... " "
```

```
print (u)
```

```
print (uu)
```

```
# Известны два угла В и С при одной стороне А
```

```
x=0.00000000
```

```
y=0.00000000
```

```
z=0.00000000
```

```
bur=bu*Pii/180 # Угол В в радианах
```

```
cur=cu*Pii/180 # Угол С в радианах
```

```
aur=Pii-(bur+cur)
```

```
au=aur*180/Pii
```

```
z=math.sin(aur) # Sin угла А
```

```
x=math.sin(bur) # Sin угла В
```

```
y=math.sin(cur) # Sin угла С
```

```
b=(a*x)/z # Сторона В
```

```
c=(b*y)/x # Сторона С
```

```
Kvz=1 # Флаг – Если =1 то работа уже выполнена.
```

```
# Далее Конец Сдвига – четыре пробела в начале каждой
```

```
строки
```

```
else:
```

```
# Далее Сдвиг – четыре пробела в начале каждой строки
```

```
Kvz=0
```

```
# Далее Конец Сдвига – четыре пробела в начале каждой
```

строки

```
if q==4:
```

```
Имеем три угла А, В и С
```

```
# Далее Сдвиг – четыре пробела в начале каждой строки
```

```
print (uu)
```

```
print (uu)
```

```
u=" Известны три угла ( определяем пропорции сторон ) "
```

```
print (u)
```

```
print (uu)
```

```
u=" Введите угол А в градусах "
```

```
print (u)
```

```
print (uu)
```

```
au=0.00000000
```

```
au=input( ) # Вводим число
```

```
au=float(au) # Принудительно в вещественное число
```

```
u=" Введите угол В в градусах "
```

```
print (u)
```

```
print (uu)
```

```
bu=0.00000000
```

```
bu=input( ) # Вводим число
```

```
bu=float(bu) # Принудительно в вещественное число
```

```
cu=180-(au+bu)
```

```
cu=float(cu) # Принудительно в вещественное число
```

```
print (uu)
```

```
print (uu)
```

```
u=" ..... " "
```



```
print (u)
print (uu)
u=" Угол А = "
ss=str(au) # Преобразуем число в строку
u=u+ss
print (u)
print (uu)
u=" Угол В = "
ss=str(bu) # Преобразуем число в строку
u=u+ss
print (u)
print (uu)
u=" Угол С = "
ss=str(cu) # Преобразуем число в строку
u=u+ss
print (u)
print (uu)
u=" ..... "
print (u)
print (uu)
print (uu)
input( ) # Ожидание нажима Enter
print (uu)
u=" Площадь = "
ss=str(S) # Преобразуем число в строку
u=u+ss
```

```
print (u)
```

```
print (uu)
```

```
u=" Радиус описанной окружности. = "
```

```
ss=str(R) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
print (uu)
```

```
u=" Радиус вписанной окружности = "
```

```
ss=str(rv) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
print (uu)
```

```
u=" Высота из угла A = "
```

```
ss=str(ha) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
print (uu)
```

```
u=" Высота из угла B = "
```

```
ss=str(hb) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
print (uu)
```

```
u=" Высота из угла C = "
```

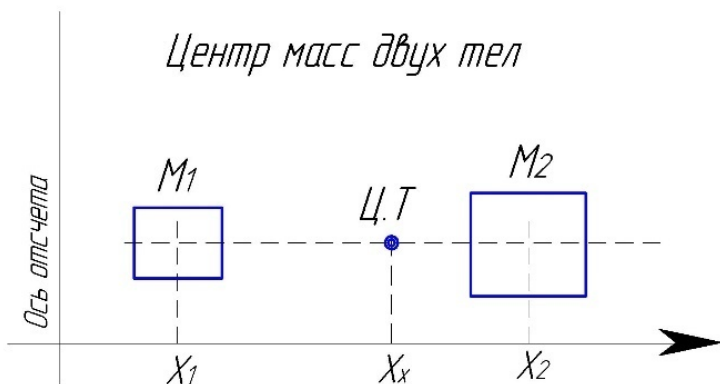
```
ss=str(hc) # Преобразуем число в строку
```

```
u=u+ss
```

```
print (u)
```

```
print (uu)
u=" Центр тяжести на одной трети высоты "
print (u)
print (uu)
input( ) # Ожидание нажима Enter
print (uu)
u=" ..... Конеч программы ..... "
print (u)
print (uu)
input( ) # Ожидание нажима Enter
print (uu)
# ..... Конеч листинга программы ..... ..
```

Центр масс нескольких тел



Определение координат центра масс нескольких тел производим методом последовательного добавления к уже определенной суммарной массе нескольких тел – еще одной массы. Начинаем с определения центра масс двух тел M_1 и M_2 . Находим суммарную массу двух тел и координаты центра масс этих двух тел в точке X_x . Далее считаем, что сумма масс двух тел сосредоточена в точке центра масс с координатами X_x . Считаем что получившееся суммарное тело – это тело № M_3 . Затем добавляем к нему еще одно тело № M_4 . Снова получаем задачу по определению центра масс двух тел № M_3 и

№ М4. Находим координаты новой точки центра масс. Программа позволяет добавлять к уже определенному суммарному телу еще одно тело столько раз – сколько необходимо.

По аналогичной методике можно разработать программу по вычислению момента инерции, координат нейтральной оси и момента сопротивления тела сложной формы. Сечение тела сложной формы разбиваем на элементарные прямоугольники и рассчитываем их как составное тело. Если при разработке данной программы возникнут сложности – можно обратиться к автору.

Листинг программы.

```
# -*- coding: cp1251 -*-
```

```
import sys # Подключили основные библиотеки
```

```
import math # Подключили математич модуль
```

```
uu=" "
```

```
print (uu)
```

```
print (uu)
```

```
print (uu)
```

```
u=" Расчет центра масс нескольких тел "
```

```
print (u)
```

```
u=" Введите массу первого тела М1 "
```

```
print (uu)
```

```
print (u)
```

```
m=0.0000
```

```
m=input( )
```

```
print (uu)
```

```
u=" Введите расстояние до нулевой оси отсчета X1 "  
print (uu)  
print (u)  
r=0.0000  
r=input( )  
print (uu)  
u=" ..... "  
print (u)  
u=" Масса первого тела M1 = "  
t=str(m) # Преобразуем число в строку  
u=u+t # сложим строки  
print (uu)  
print (u)  
u=" Расстояние до нулевой оси отсчета X1 = "  
t=str(r) # Преобразуем число в строку  
u=u+t # сложим строки  
print (uu)  
print (u)  
m=float(m) # Принудительно в вещественное число  
r=float(r) # Принудительно в вещественное число  
mm=0.00000  
mm=m*r  
u=" Момент массы относительно нулевой оси = "
```

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.