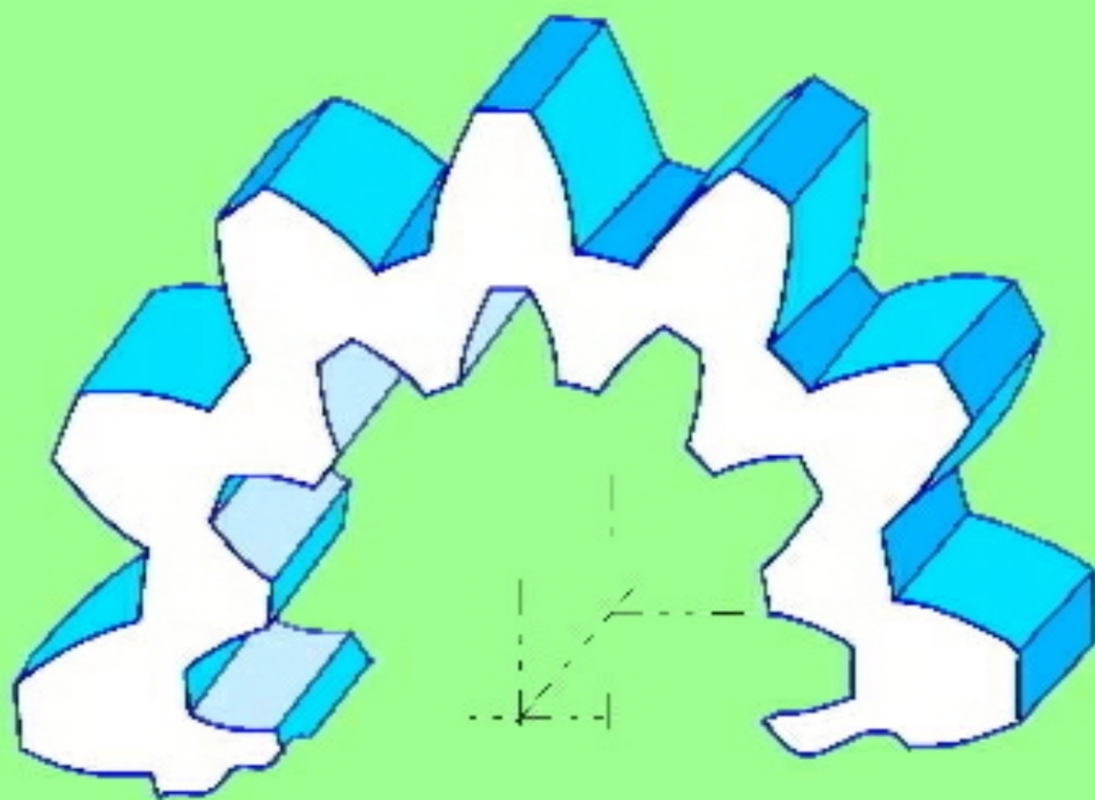


Сергей Гаврилов

Python-3

полезные программы

книга третья



12+

Сергей Гаврилов

**Полезные программы
Python-3. Книга третья**

«ЛитРес: Самиздат»

2020

Гаврилов С. Ф.

Полезные программы Python-3. Книга третья / С. Ф. Гаврилов —
«ЛитРес: Самиздат», 2020

Эта книга для начинающих программистов, желающих начать программировать на языке Python 3. Книга будет полезна для студентов механиков. Набор программ создан для цеховых конструкторов механиков. Предлагаемые в книге расчеты экономят рабочее время и уменьшают ошибки.. 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 программ снабжены пояснениями.

Содержание

Введение	5
Примечание	6
Написание чисел с дробной десятичной частью.	8
Написание комментариев.	9
Линейная интерполяция	10
Координаты точек на радиусной кривой	13
Хорда – Прогиб – Радиус	17
Расчет по Хорде и углу раствора	20
Расчет по Хорде и длине Дуги	21
Расчет по Хорде – Прогибу	23
Расчет по Радиусу и Прогибу	25
Расчет по Прогибу и длине Дуги	26
Расчет по Прогибу и Углу раствора	28
Расчет по Углу раствора и длине Дуги	30
Расчет Правильного многогранника	32
Овал (коробовая кривая)	36
Две известные стороны и угол между ними	44
Одна сторона – А и два угла В и С	46
Центр масс нескольких тел	51
Круглое трубное сечение	55
Конец ознакомительного фрагмента.	57

Введение

Книга третья.

Инженер конструктор отдела гл.механика Гаврилов Сергей Фёдорович написал эту книгу для начинающих программистов, желающих начать программировать на языке Python 3. Книга будет полезна для студентов механиков и для конструкторов механиков. Данная книга содержит листинги – набор небольших рабочих программ облегчающих расчеты встречающиеся в практической работе цехового конструктора. Листинги выполнены как есть, без сокращений, и после копирования готовы к работе. 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.

Сама программа начинается со строки «# -*- coding: cp1251 -*-» и заканчивается строкой «# Конец листинга программы». Копируя листинг в файл .txt – надо копировать только программу, любой текст до тела программы и после тела программы даст сбой в работе.

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 английском.

В некоторых случаях помогает удаление строки в начале программы

«# -*- coding: cp1251 -*-» и программа заработает. Иногда помогает удалить конфликтную строку и забить ее вновь в текстовом редакторе.

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

При верстке книги к печати «умные программы» что-то убирают, а что-то добавляют, искажая оригинал. Для программы маленькие буквы и большие буквы – это разные буквы, а «умные программы» при верстке в начале строки бывает, что заменяют маленькую букву большой или меняют форму ковычек – программа вылетает.

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. Строка комментариев после значка # не должна переноситься на следующую строку. В блоках- там, где блок выделяется пробелами, перед строками блока будет 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 – это тоже не дает работать программе – искать такие ошибки лучше в «IDLE(Python GUI)». Часть программ из книги первой и из книги второй в этой книге повторяются – это программы, часто применяемые в работе.

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. Для человека ответ на экран удобнее выводить в градусах – при выводе на экран радианы программно переводят в градусы. При вводе угловых величин- их обычно вводят в градусах и, перед подачей в расчетную часть программы, переводят программно в радианы.

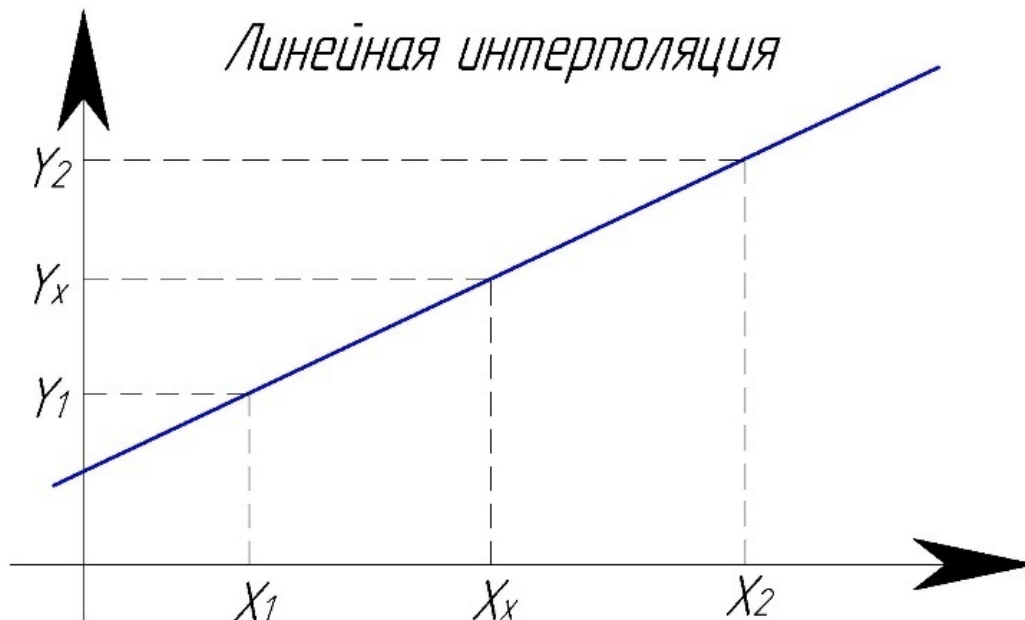
Написание комментариев.

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.

Листинги программ

:

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..
yx=((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)

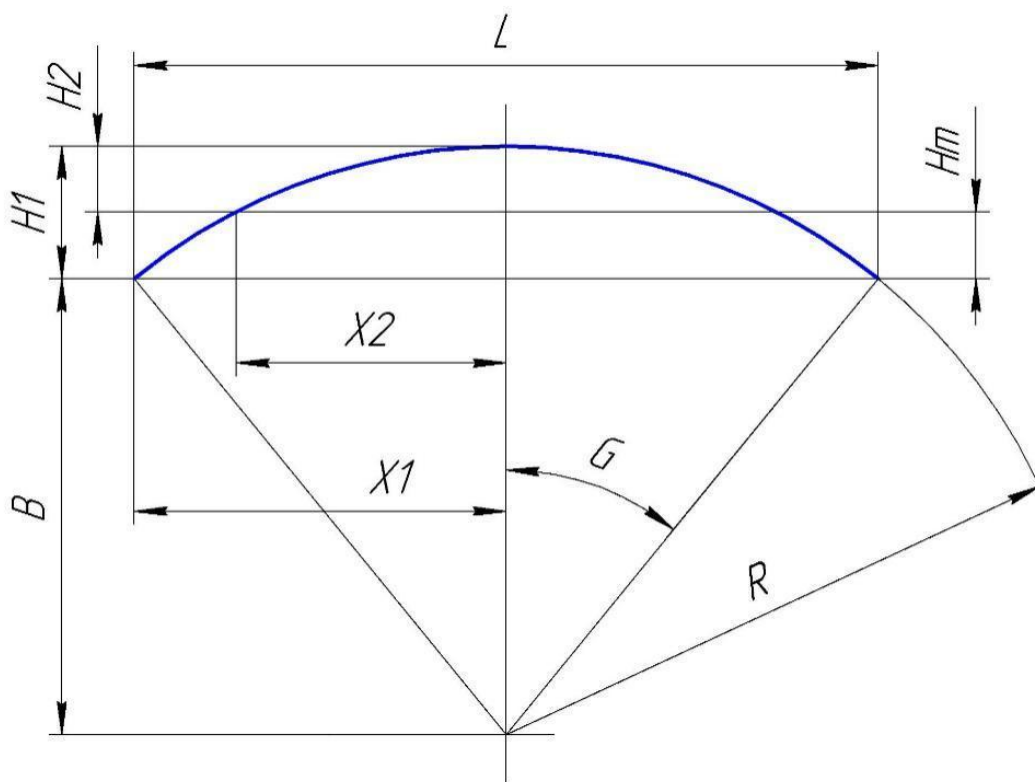
```

```
print (uu)
print (uu)
input( ) # Ожидание нажима Ентер
# 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 .....
```

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 результатов
# 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 хорды "
print (u)
u=" от точки X2 на хорде проводим перпендикуляр Нм рассчитанной величины. "
print (u)
u=" При вводе размера по горизонту = нулю – выход из программы "
print (u)
print (uu)
input( ) # Ожидание нажима Enter
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 X2 = 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"

```

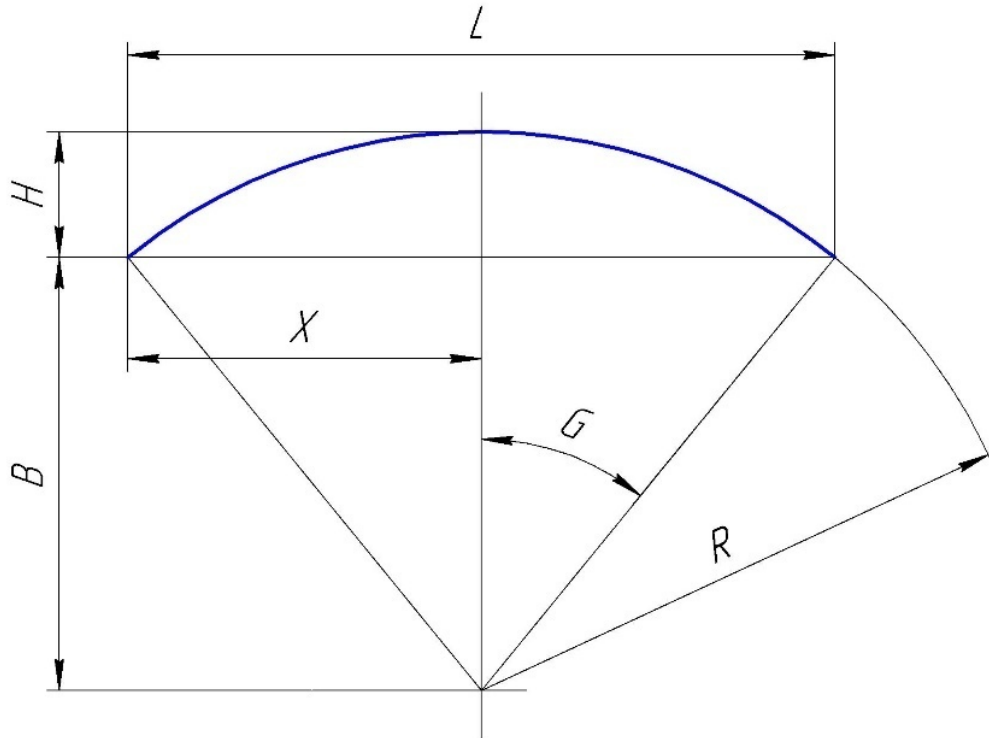
```

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 идет циклический расчет
# Далее Сдвиг – четыре пробела в начале каждой строки
print (uu)
u=" Введите расстояние от центра хорды до перпендикуляра 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
Gsf = 461030/2725231222..
u=" На расстоянии от центра = "
ss=str(Lx) # Преобразуем число в строку
u=u+ss
print (u)
print (uu)
u=" Величина подъема ( перпендикуляра ) Hm = "
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=" ..... Конеч программы ..... "
print (u)
print (uu)
input() # Ожидание нажима Ентер –
# Позволяет рассмотреть результаты расчета
# ..... Конеч листинга программы ..... ..
```

Хорда – Прогиб – Радиус



В этом разделе объединены двенадцать небольших программ. Производится расчет при разном наборе исходных данных.

Расчет по Хорде – Радиусу

Листинг программы.

```
# -*- coding: cp1251 -*-
import math # Подключили математический модуль
Pi=math.pi # Вытащили число " Пи "
uu=" "
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=0.0000000000
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
#
Вывод по Хорде Прогиб Радиус
# Этот вывод без изменений будет во всех программах
# « Хорда прогиб радиус » – поэтому в дальнейшем мы повторяться
# не будем – ограничимся только заголовком.
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)  
PlSe=u  
print (uu)  
u=" Площадь треугольника под горбушкой = "  
ss=str(St) # Преобразуем число в строку  
u=u+ss  
print (u)  
PlTr=u  
print (uu)  
u=" Площадь горбушки = "  
ss=str(S) # Преобразуем число в строку  
u=u+ss  
print (u)  
PlGo=u  
print (uu)  
input( ) # Ожидание нажима Ентер  
# Позволяет рассмотреть результаты расчета  
# ..... Конец листинга программы ..... ..
```

Расчет по Хорде и углу раствора

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

Листинг программы.

```
# -*- coding: cp1251 -*-
import math # Подключили математический модуль
Pii=math.pi # Вытащили число " Пи "
uu=" "
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=0.0000000000
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
# .....
# Вывод по Хорде Прогиб Радиус
# Далее следует блок вывода результата ( смотри выше ).
input( ) # Ожидание нажима Ентер
# Позволяет рассмотреть результаты расчета
# ..... Конец листинга программы .....
```

Расчет по Хорде и длине Дуги

Листинг программы.

```
# -*- coding: cp1251 -*-
import math # Подключили математический модуль
Pii=math.pi # Вытащили число " Пи "
uu=" "
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 – четыре пробела 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 Сдвига нет
yg=y*180/Pii # Угол " у " в градусах
au=yg
R=rt # Нашли радиус
y=(R*R)-(aa*aa)
v=math.sqrt(y) # Квадратный корень из " у "
b=R-v # Нашли прогиб
D=R+v
Sk=Pii*D*D*yg/(4*360) # Площадь сектора круга с углом auг
St=aa*v # Площадь треугольника в секторе
S = Sk-St # Площадь горбушки
# Вывод по Хорде Прогиб Радиус
# Далее следует блок вывода результата ( смотри выше ).
```

```
input() # Ожидание нажима Enter
# Позволяет рассмотреть результаты расчета
# ..... Конец листинга программы .....
# ..... .....
```

Расчет по Хорде – Прогибу

Листинг программы.

```
# -*- coding: cp1251 -*-
import math # Подключили математический модуль
Pi=math.pi # Вытащили число " Пи "
uu=" "
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) # Принудительно в вещественное число
u=" Введите Стрелу Прогиба "
print (uu)
print (u)
print (uu)
b=0.00000000
b=input( ) # Вводим число
b=float(b) # Принудительно в вещественное 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 – четыре пробела в начале каждой строки
Rt=Rt+Rd
x=((Rt*Rt)-(k*k))
c=math.sqrt(x) # Квадратный корень из " x "
Ex=(Rt-c)-b
# E=math.abs(x)
# Конец подбора .....
```

```
# Далее Сдвига нет ...
R=Rt
D=R+R
x=k/Rt
sur=math.asin(x) # Угол A в радианах
su=sur*180/Pii # Угол A в градусах
au=2*su
yg=au
L=(R+R)*Pii*au/360
# .....
# Вывод по Хорде Прогиб Радиус
# Далее следует блок вывода результата ( смотри выше ).
input() # Ожидание нажима Ентер
# Позволяет рассмотреть результаты расчета
# ..... Конец листинга программы ..... ..
# ..... ..
```

Расчет по Радиусу и Прогибу

Листинг программы.

```
# -*- coding: cp1251 -*-
import math # Подключили математический модуль
Pii=math.pi # Вытащили число " Пи "
uu=" "
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) # Принудительно в вещественное число
u=" Введите Прогиб "
print (uu)
print (u)
print (uu)
b=0.00000000
b=input( ) # Вводим число
b=float(b) # Принудительно в вещественное 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
# Вывод по Хорде Прогиб Радиус
# Далее следует блок вывода результата ( смотри выше ).
input( ) # Ожидание нажима Ентер
# Позволяет рассмотреть результаты расчета
# ..... Конец листинга программы .....
# .....
```

Расчет по Прогибу и длине Дуги

Листинг программы.

```
# -*- coding: cp1251 -*-
import math # Подключили математический модуль
Pii=math.pi # Вытащили число " Пи "
uu=" "
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) # Принудительно в вещественное число
print (uu)
u=" Введите длину Дуги "
print (u)
print (uu)
L=0.00000000
L=input( ) # Вводим число
L=float(L) # Принудительно в вещественное число
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 – четыре пробела в начале каждой строки
Rt=Rt+xx
ygr=(2*Pii)* (L/((Rt+Rt)*Pii)) # Угол раствора хорды радиан
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 нет..
db=db+xb
xx=b/100000000 # Приращение радиуса
xb=b/1000000
# Уточнение подбором ( вторая ступень )
while db >xb:
# Далее Сдвиг – четыре пробела в начале каждой строки
Rt=Rt+xx
ygr=(2*Pii)* (L/((Rt+Rt)*Pii)) # Угол раствора хорды радиан
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 – нет ...
# Конец подбора .....
# Gsf = 461030/2725231222...
R=Rt
ygr=ygr*180/Pii # Угол раствора хорды град.
# Далее Сдвига – нет....
x=a/(R+R)
k=math.asin(x)
au=(k+k)*180/Pii # Угол на хорду в градусах
# Вывод по Хорде Прогиб Радиус
# Далее следует блок вывода результата ( смотри выше ).
input( ) # Ожидание нажима Ентер
# Позволяет рассмотреть результаты расчета
# ..... Конец листинга программы .....
# .....

```

Расчет по Прогибу и Углу раствора

Листинг программы.

```
# -*- coding: cp1251 -*-
import math # Подключили математический модуль
Pii=math.pi # Вытащили число " Пи "
uu=" "
print (uu)
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)
au=0.00000000
au=input( ) # Вводим число
au=float(au) # Принудительно в вещественное число
aur=au*Pii/180
u=" Введите стрелу Прогиба "
print (u)
print (uu)
b=0.00000000
b=input( ) # Вводим число
b=float(b) # Принудительно в вещественное число
z=math.cos(aur/2)
R=b/(1-z)
x=0.00000000
y=0.00000000
v=0.00000000
Sk=0.00000000
St=0.00000000
S=0.0000000000
Ex=1.0000000
x= math.cos(aur/2)
y=R*x
x=(R*R)-(y*y)
c=math.sqrt(x) # Квадратный корень из " x "
a=c+c
D=R+R
# Вывод по Хорде Прогиб Радиус
# Далее следует блок вывода результата ( смотри выше ).
input( ) # Ожидание нажима Enter
# Позволяет рассмотреть результаты расчета
# ..... Конец листинга программы .....
```

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

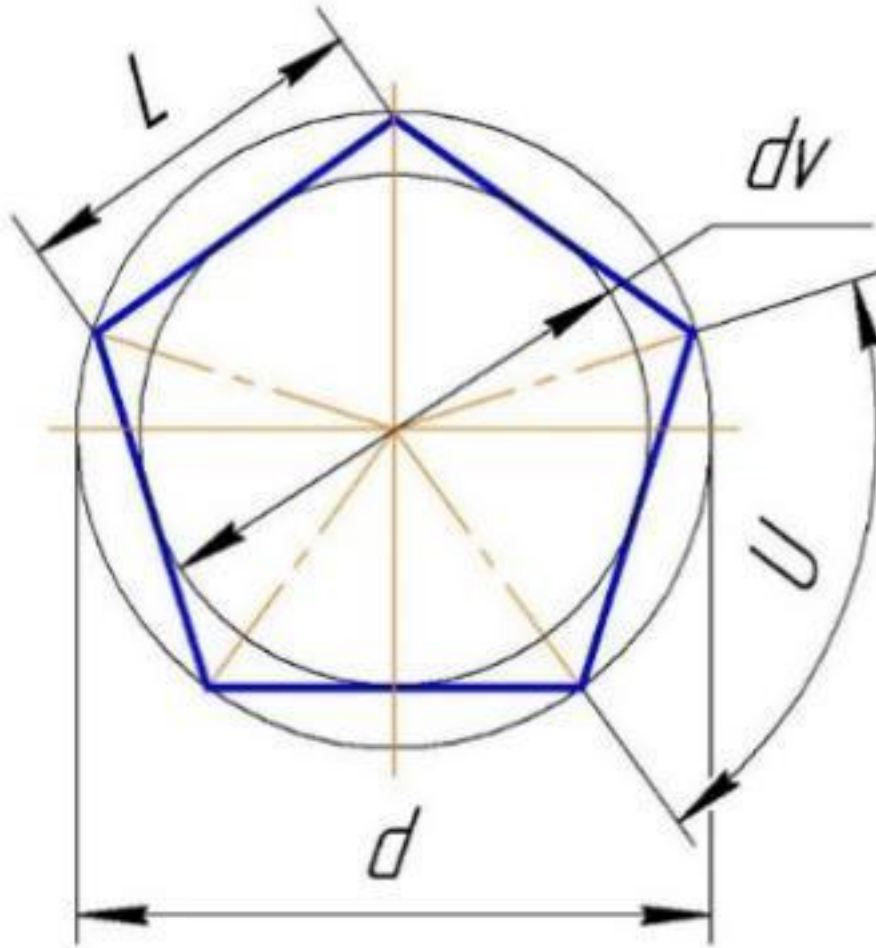
Расчет по Углу раствора и длине Дуги

Листинг программы.

```
# -*- coding: cp1251 -*-
import math # Подключили математический модуль
Pii=math.pi # Вытащили число " Пи "
uu=" "
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) # Принудительно в вещественное число
print (uu)
u=" Введите длину Дуги "
print (u)
print (uu)
L=0.00000000
L=input( ) # Вводим число
L=float(L) # Принудительно в вещественное число
C=L*360/yg
D=C/Pii
R=D/2
x=0.00000000
y=0.00000000
v=0.00000000
Sk=0.00000000
St=0.00000000
S=0.0000000000
Ex=1.00000000
yr=yg*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
au=yg
# Вывод по Хорде Прогиб Радиус
# Далее следует блок вывода результата ( смотри выше ).
input( ) # Ожидание нажима Ентер
```

```
# Позволяет рассмотреть результаты расчета
# ..... Конец листинга программы .....
# ..... .....
```

Расчет Правильного многогранника



Листинг программы.

```
# -*- coding: cp1251 -*-
import math # Подключили математический модуль
Pii=math.pi # Вытащили число " Пи "
uu=" "
print (uu)
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 "
```

```

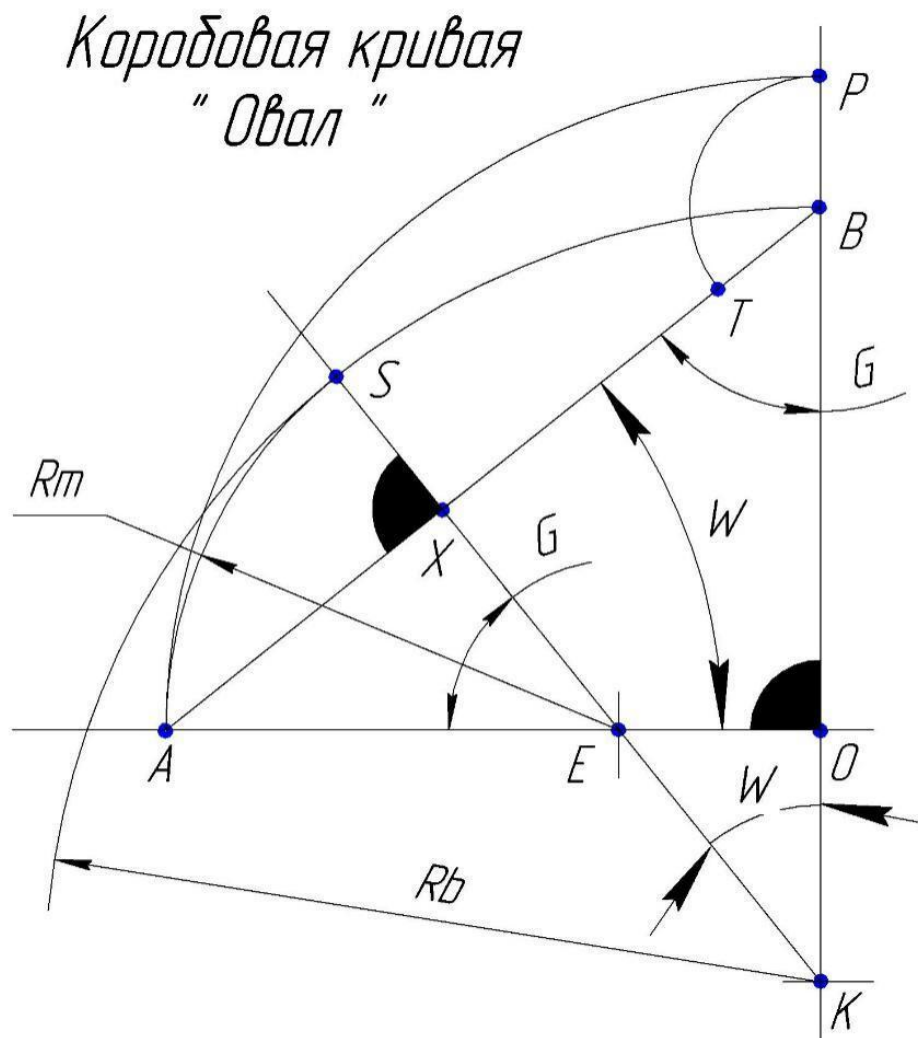
print (u)
print (uu)
D=0.00000000
D=input( ) # Вводим число
D=float(D) # – Принудительно в вещественное число
x=0.00000000
y=0.00000000
v=0.00000000
Sk=0.00000000
St=0.00000000
S=0.0000000000
Ex=10000000
vv=0 # Флаг расчета при заданном Описанном диаметре
vv=float(vv)
if D==0:
# Далее Сдвиг – четыре пробела 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=0.00000000
Sv=input( ) # Вводим число
Sv=float(Sv) # – Принудительно в вещественное число
u=" Введите число Граней "
print (uu)
print (u)
print (uu)
n=0.00000000
n=input( ) # Вводим число
n=float(n) # Принудительно в вещественное число
sur=2*Pii/n # Угол А в радианах
su=360/n # Угол А в градусах
au=su
yg=au
yr=sur
x=math.cos(sur/2) # Cos Угла А
y=Sv/2
R=y/x
D=R+R
vv=1 # Флаг расчета при заданном Вписанном диаметре
# Далее Сдвига нет...
if vv==0:
# Далее Сдвиг – четыре пробела в начале каждой строки
u=" Введите число Граней "
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 Сдвига нет ....
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)
input() # Ожидание нажима Ентер
# Позволяет рассмотреть результаты расчета
# ..... Конец листинга программы .....
# ..... .....
```

Овал (коробовая кривая)



На схеме показано построение коробовой кривой (овала).

Отрезок AO – половина большей оси овала. Отрезок BO – 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=" "
```

```
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 радиусом
Sms=Sm*2*ub/Pii # Площадь секторов с Малым радиусом
ko=bk-ob
oo1=oa-ao1
Str=2*ko*oo1 # Площадь четырех вычитаемых треугольников
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
Pr=((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

```

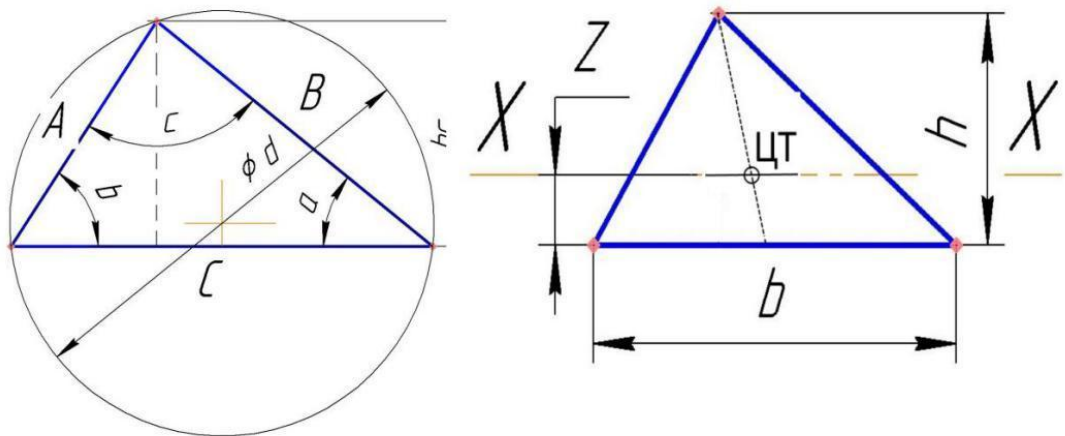
```
print (uu)
u=" Периметр овала = "
ss=str(Pp) # Преобразуем число в строку
u=u+ss
print (u)
u10=u
print (uu)
u=" Площадь овала = "
ss=str(So) # Преобразуем число в строку
u=u+ss
print (u)
u11=u
print (uu)
u=" Диаметр равного по площади – круга = "
ss=str(Ds) # Преобразуем число в строку
u=u+ss
print (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 Ентер
u=" ..... Конеч программы ....."
print (u)
print (uu)
input() # Ожидание нажима Ентер
# . ..... Конеч листинга программы .....
```

Треугольни

к



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 противоположащую сторону пополам.

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 "
#
Известны три стороны
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
# .....
# Это « добавочный расчет » во все
# подпрограммы Треугольника
# И вывод результатов расчета
x=0.00000000

```



```
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=" ..... Конец программы ..... "
print (u)
print (uu)
input( ) # Ожидание нажима Enter
print (uu)
# .... Конец листинга программы .....
```

Две известные стороны и угол между ними

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 "

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=ci*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
Gsf = 461030/2725231222...
# Далее вставляем блок добавочных расчетов
# И вывода результатов ( смотри выше ).
# .... Конец листинга программы .....
```

Одна сторона – А и два угла В и С

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 "
print (uu)
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
# Далее вставляем блок добавочных расчетов
# И вывода результатов ( смотри выше ).
# .... Конец листинга программы .....
Имеем три угла А, В и С
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 "
print (uu)
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)

```

```

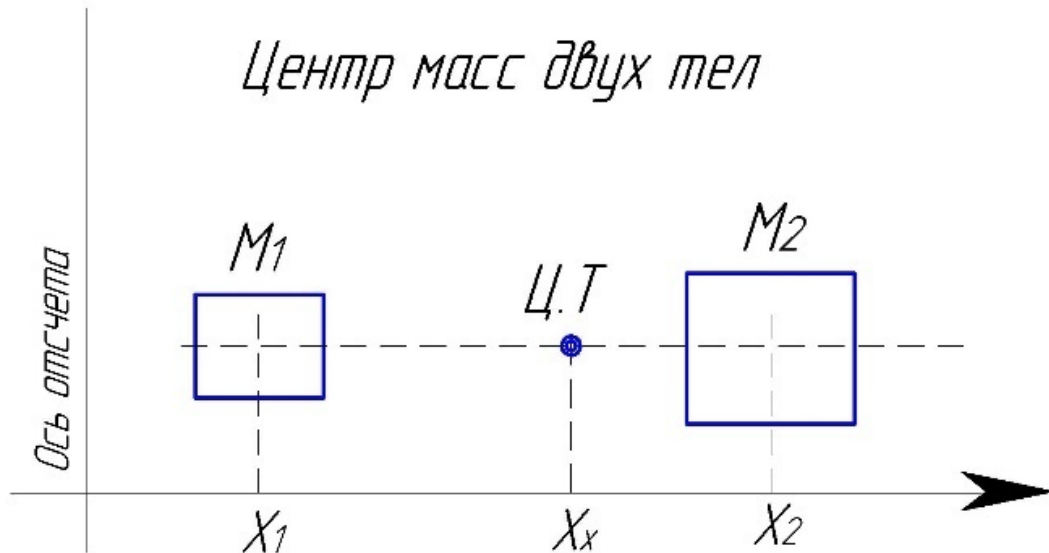
bu=0.00000000
bu=input( ) # Вводим число
bu=float(bu) # Принудительно в вещественное число
cu=180-(au+bu)
cu=float(cu) # Принудительно в вещественное число
print (uu)
print (uu)
u=" .... ..... .... .... "
print (u)
print (uu)
aur=au*Pii/180 # Угол С в радианах
bur=bu*Pii/180 # Угол С в радианах
cur=cu*Pii/180 # Угол С в радианах
a=1000*(math.sin(aur))
b=1000*(math.sin(bur))
c=1000*(math.sin(cur))
# " ,,,, ,,,, ,,,, ,,,,, ,,,,, ,,,, ,,,,, ,,,,, ,,,,, "
# Это « добавочный расчет »
# во все подпрограммы Треугольника
x=0.00000000
z=0.00000000
R=0.00000000
ha=0.00000000
hb=0.00000000
hc=0.00000000
z=math.sin(aur) # Sin угла А
R=a/(2*z) # R Описанной окружности
hc=b*z # Высота из угла С
z=math.sin(cur) # Sin угла С
hb=a*z # Высота из угла В
z=math.sin(bur) # Sin угла В
ha=c*z # Высота из угла А
S=a*ha/2 # Площадь треугольника
Pe=a+b+c # Периметр
rv=(S+S)/Pe # Радиус вписанной окружности
u=" ,,,, ,,,, ,,,, ,,,,, ,,,,, ,,,, ,,,,, ,,,,, ,,,,, "
print (uu)
print (u)
print (uu)
print (uu)
u=" Сторона А = "
ss=str(a) # Преобразуем число в строку
u=u+ss
print (u)
print (uu)
u=" Сторона В = "
ss=str(b) # Преобразуем число в строку
u=u+ss
print (u)

```

```
print (uu)
u=" Сторона С = "
ss=str(c) # Преобразуем число в строку
u=u+ss
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 А = "
ss=str(ha) # Преобразуем число в строку
u=u+ss
print (u)
print (uu)
u=" Высота из угла В = "
ss=str(hb) # Преобразуем число в строку
```

```
u=u+ss
print (u)
print (uu)
u=" Высота из угла С = "
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=" ..... ..... ..... Конец программы ..... ..... "
print (u)
print (uu)
input( ) # Ожидание нажима Enter
print (uu)
# ..... ..... ..... Конец 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 № M_4 . 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.

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 M1 "
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 = "
t=str(mm) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
print (uu)
u=" ..... "
print (u)
Pii=math.pi # Вытащили число " Пи "
ms=m # Сумма масс.
mms=mm # Сумма моментов масс.
n=1 # Порядковый номер прямоугольника
n=int(n) # Переведем в целое число
# " ..... "
x=0
x=int(x) # Переведем в целое число
y=int(r) # Переведем в целое число
while y>x:
# Далее Сдвиг – четыре пробела в начале каждой строки
u=" ..... "
print (u)
print (uu)
n=n+1
u=" Ввести значение массы № "
t=str(n) # Преобразуем число в строку

```

```

u=u+t # сложим строки
print (u)
print (uu)
m=0.0000
m=input( )
print (uu)
u=" Расстояние до нулевой оси отсчета "
print (uu)
print (u)
r=0.000000
r=input( )
r=float(r) # Принудительно в вещественное число
m=float(m) # Принудительно в вещественное число
ms=ms+m # Сумма масс.
mm=m*r
mms=mms+mm # Сумма моментов масс.
rx=0.000000
rx=mms/ms # Расстояние от нулевой оси до Ц.Т. системы
y=r
print (uu)
if y == x:
# Далее Сдвиг – восемь пробелов в начале каждой строки
    break # Выход из цикла
# Далее Сдвиг – четыре пробела в начале каждой строки
u=" ..... "
print (u)
print (u)
print (uu)
u1=" Добавили массу № "
u2=" = "
t1=str(n) # Преобразуем число в строку
t2=str(m) # Преобразуем число в строку
u=u1+t+u2+t2
print (uu)
print (u)
u1=" Расстояние до нулевой оси = "
t=str(r) # Преобразуем число в строку
u=u1+t # сложим строки
print (uu)
print (u)
u1=" Добавленный момент = "
t=str(mm) # Преобразуем число в строку
u=u1+t # сложим строки
print (uu)
print (u)
u1=" Сумма масс = "
t=str(ms) # Преобразуем число в строку
u=u1+t # сложим строки
print (uu)

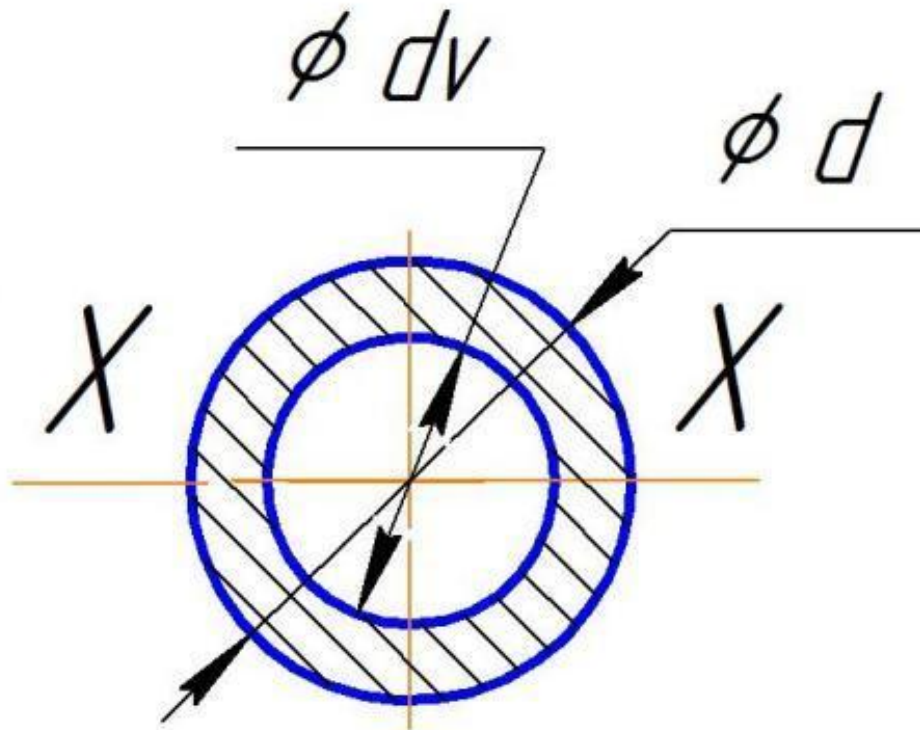
```

```

print (u)
u1=" Сумма моментов = "
t=str(mms) # Преобразуем число в строку
u=u1+t # сложим строки
print (uu)
print (u)
print (uu)
u=" Расстояние Ц.Т. от нулевой оси = "
t=str(rx) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
print (uu)
u=" ..... "
print (u)
print (u)
print (uu)
# Далее Сдвига нет ...
u=" ..... Конец работы программы ..... "
print (u)
print (uu)
u=" ..... "
print (u)
print (u)
print (uu)
# .....
input() # Ожидание нажима Ентер
# ..... Конец листинга программы .....

```

Круглое трубное сечение



```
# Листинг программы ..
# -*- coding: cp1251 -*-
import sys # Подключили основные библиотеки
import math # Подключили математич модуль
uu=" "
print (uu)
print (uu)
print (uu)
u=" Расчет параметров круглого трубного сечения "
print (u)
u=" Введите наружный диаметр сечения , "
print (uu)
print (u)
d=0.0000
d=input( )
print (uu)
# .....
u=" Введите внутренний диаметр сечения , "
print (uu)
print (u)
dv=0.0000
```

```
dv=input( )
print (uu)
print (uu)

u=" ..... "
print (u)
l=1 # Зададим длину сечения = 1 мм.
d=float(d) # Принудительно в вещественное число
l=float(l) # Принудительно в вещественное число
# .....
u=" ..... "

u=" Диаметр наружный сечения = "
t=str(d) # Преобразуем число в строку
u=u+t # сложим строки
print (uu)
print (u)
u=" Диаметр внутренний сечения = "
t=str(dv) # Преобразуем число в строку
u=u+t # сложим строки
```

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.