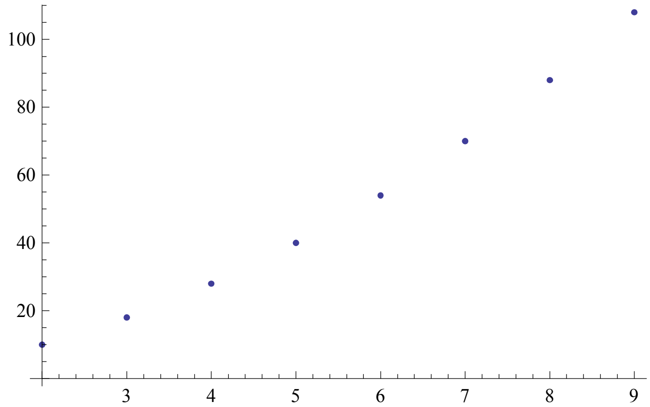
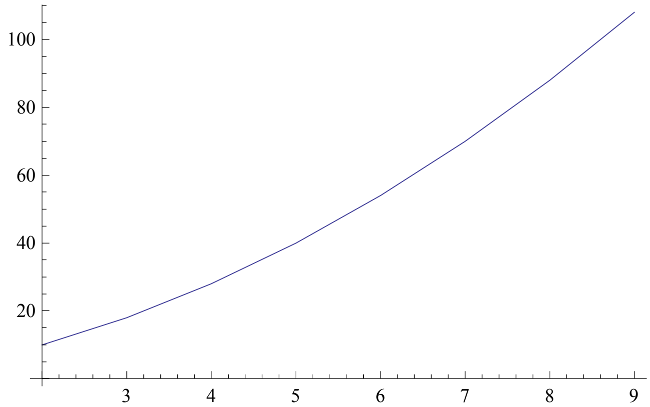
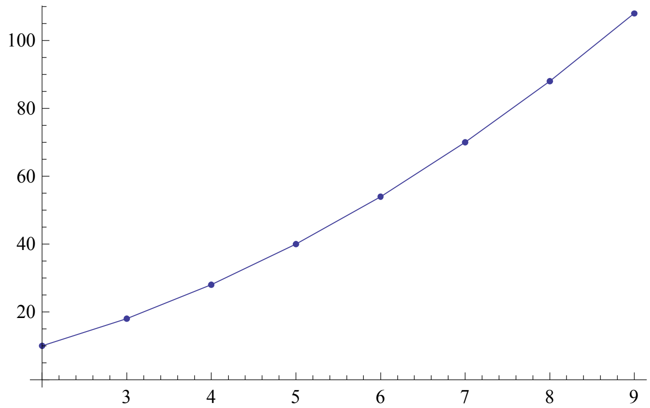
```
topic : Graphic data;

date : 27 / 3 / 2017;
Q1; graph i^2 + 3 i for i = 2, ....,
9 and also a set if lines joining them. display both graphs together;

a = Table[{i, i^2 + 3 i}, {i, 2, 9}]
g1 = ListPlot[a]
g2 = ListLinePlot[a]
Show[g1, g2]
```

```
{{2, 10}, {3, 18}, {4, 28}, {5, 40}, {6, 54}, {7, 70}, {8, 88}, {9, 108}}
```







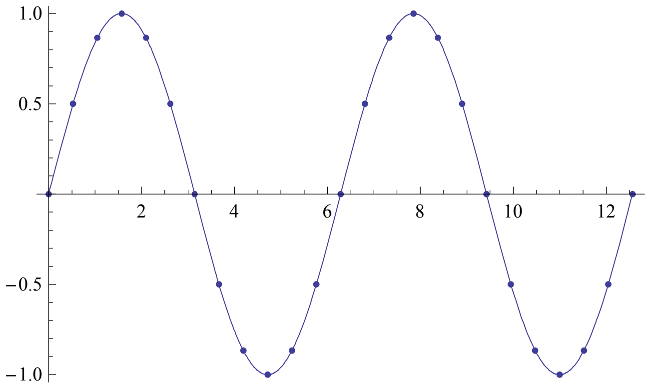

```
?ListPlot
```

ListPlot[{y_1, y_2, …}] plots points corresponding to a list of values, assumed to correspond to x coordinates 1, 2, ….
ListPlot[{{x_1, y_1}, {x_2, y_2}, …}] plots a list of points with specified x and y coordinates.
ListPlot[{$list_1$, $list_2$, …}] plots several lists of points. ≫

```
? Table
```

Table[$expr$, {i_{max}}] generates a list of i_{max} copies of $expr$.
Table[$expr$, {i, i_{max}}] generates a list of the values of $expr$ when i runs from 1 to i_{max}.
Table[$expr$, {i, i_{min}, i_{max}}] starts with $i = i_{min}$.
Table[$expr$, {i, i_{min}, i_{max}, di}] uses steps di.
Table[$expr$, {i, {i_1, i_2, …}}] uses the successive values i_1, i_2, ….
Table[$expr$, {i, i_{min}, i_{max}}, {j, j_{min}, j_{max}}, …] gives a nested list. The list associated with i is outermost. ≫

```
Q2 : graph sinx for 0 <= x ≤
   4 Pi radians and its discretized version with step size Pi / 6. display both graphs together;

g1 = Plot[Sin[x], {x, 0, 4 Pi}];
g2 = ListPlot[Table[{x, Sin[x]}, {x, 0, 4 Pi, Pi / 6}]]; Show[g1, g2]
```

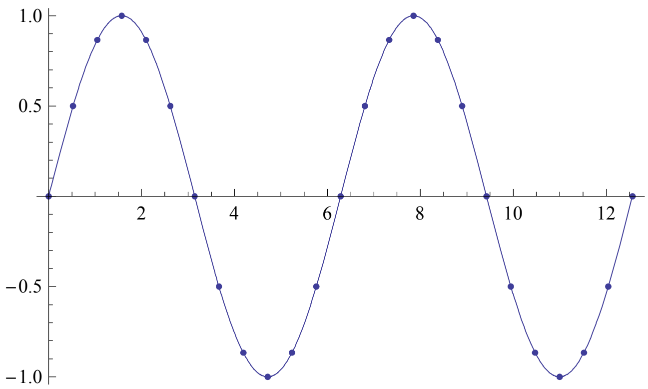



```
? Do
```

Do[$expr$, {i_{max}}] evaluates $expr$ i_{max} times.
Do[$expr$, {i, i_{max}}] evaluates $expr$ with the variable i successively taking on the values 1 through i_{max} (in steps of 1).
Do[$expr$, {i, i_{min}, i_{max}}] starts with $i = i_{min}$.
Do[$expr$, {i, i_{min}, i_{max}, di}] uses steps di.
Do[$expr$, {i, {i_1, i_2, …}}] uses the successive values i_1, i_2, ….
Do[$expr$, {i, i_{min}, i_{max}}, {j, j_{min}, j_{max}}, …] evaluates $expr$ looping over different values of j, etc. for each i. ≫

Hint : in this question for discretized values
we use the command Table[$expr$**, {**i**,** i_{min}**,** i_{max}**,** di**}] uses steps** di**. ;**

```
g1 = Plot[Sin[x], {x, 0, 4 Pi}];
g2 = ListPlot[Table[{x, Sin[x]}, {x, 0, 4 Pi, Pi / 6}]]; Show[g1, g2]
```



```
Q : Generate a data composed of first ten
   positive odd integers and their cubes . display this data ;
f = (2 i + 1) ; Table[{f, f^3}, {i, 0, 9}]
```

```
{{1, 1}, {3, 27}, {5, 125}, {7, 343}, {9, 729},
 {11, 1331}, {13, 2197}, {15, 3375}, {17, 4913}, {19, 6859}}
```

```
topic : Displaying specific shapes : Graphics ;
```

```
note : to display the graphics or shapes in mathematica we use the command Graphics
   which consists on primitives . in graphics the primitives are building blocks
   which are used to display the shapes like (circles , lines , rectangles , disks) ;
```

```
? Circle
```

Circle[{x, y}, r] is a two-dimensional graphics primitive that represents a circle of radius r centered at the point x, y.
Circle[{x, y}] gives a circle of radius 1.
Circle[{x, y}, r, {θ_1, θ_2}] gives a circular arc.
Circle[{x, y}, {r_x, r_y}] gives an ellipse with semi-axes of lengths r_x and r_y, oriented parallel to the coordinate axes. ≫

```
? Line
```

Line[{pt_1, pt_2, …}] is a graphics primitive which represents a line joining a sequence of points.
Line[{{pt_{11}, pt_{12}, …}, {pt_{21}, …}, …}] represents a collection of lines. ≫

```
? Disk
```

Disk[{x, y}, r] is a two-dimensional graphics primitive that represents a filled disk of radius r centered at the point x, y.
Disk[{x, y}] gives a disk of radius 1.
Disk[{x, y}, r, {θ_1, θ_2}] gives a segment of a disk.
Disk[{x, y}, {r_x, r_y}] gives an elliptical disk
with semi-axes of lengths r_x and r_y, oriented parallel to the coordinate axes. ≫

```
? Rectangle
```

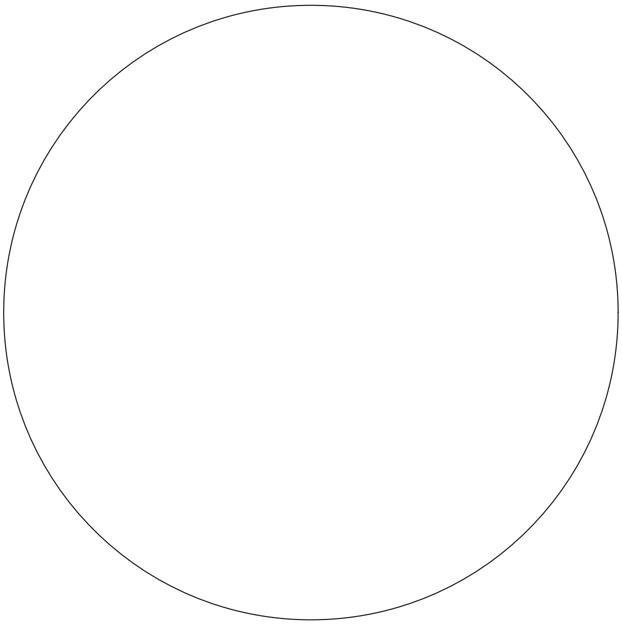
Rectangle[{x_{min}, y_{min}}, {x_{max}, y_{max}}] is a two-dimensional
graphics primitive that represents a filled rectangle, oriented parallel to the axes.
Rectangle[{x_{min}, y_{min}}] corresponds to a unit square. ≫

```
Q1 : Use Graphics to draw a circle of radius 3 meters centred at (-1, 4) ;
```

```
? Graphics
```

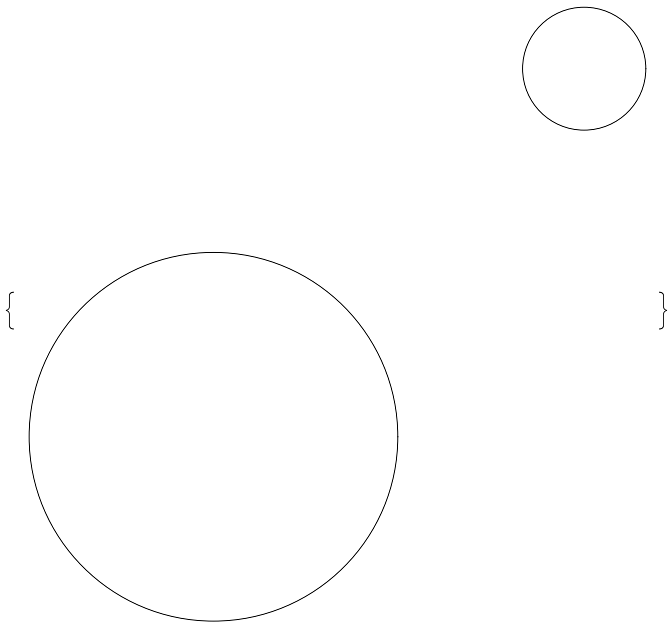
Graphics[*primitives*, *options*] represents a two-dimensional graphical image. ≫

```
Graphics[Circle[{-1, 4}, 3]]
```



→ Notes : if we change the centre and radius of the circle the output is not chnage . why ? the reason is that there is only one primitive in the Graphics . if we take more than one primitives in the Graphics the output will changes because one primitive compare the results to the other primitive . the output will be chnage ;

```
{Graphics[{Circle[{-1, 4}, 3], Circle[{5, 10}, 1]}]}
```



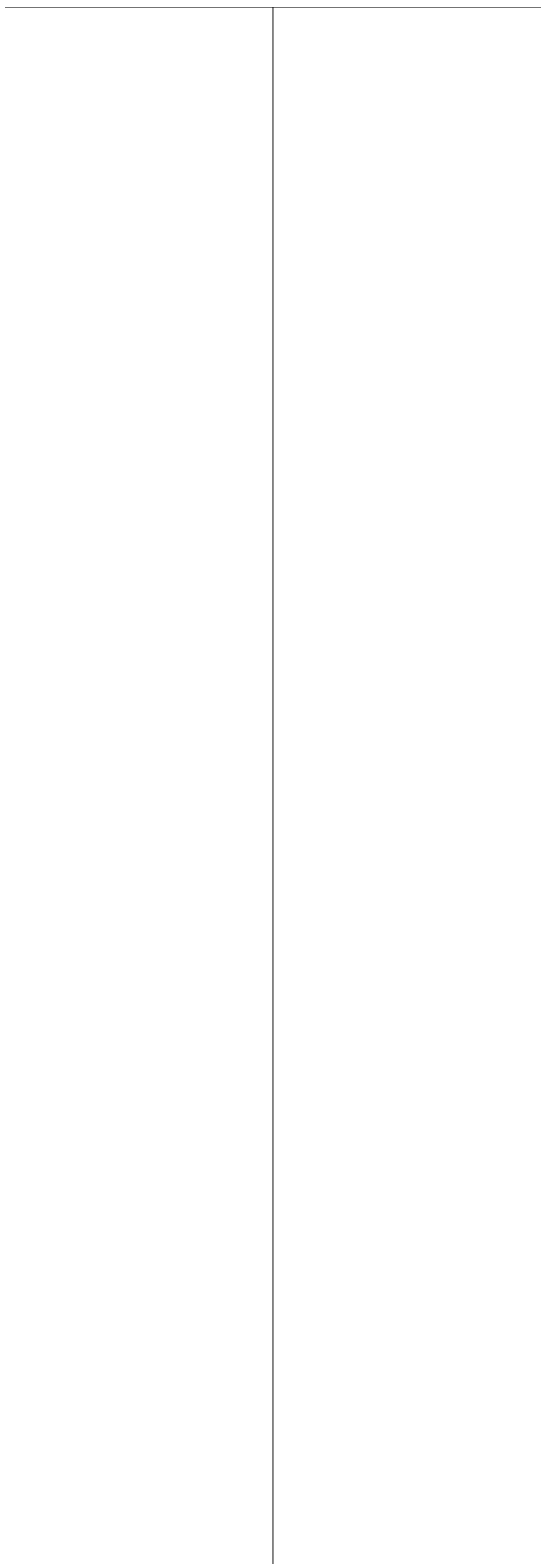
Q2 : draw a pendulume of unit length in its dowmmost position . take the radius of its ball to be 5 cm.include a hanging of 20 cm ;

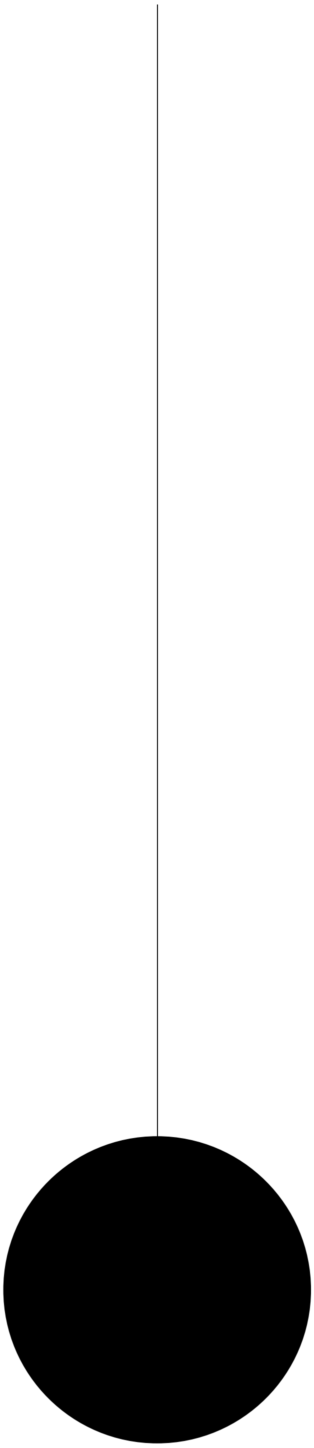
→ in this problem we have two lines one for the unit length of pendulum and second line for the hangging of the ball ;

→ one disk for the bal with radius 5 cm;

→ in this problem there are three primitives ;

```
Graphics[{Line[{{-0.1, 0}, {0.1, 0}}], Line[{{0, 0}, {0, -1}}], Disk[{0, -1}, 0.05]}]
```

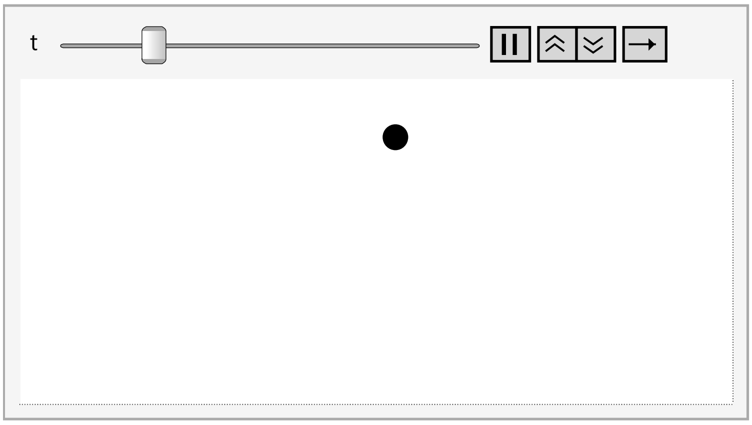




Q3 : animate a projectile for 0 ≤ t ≤ 1, v = 5 m / s,
theta = 60 Degree . take radius of ball 5 cm, . used fixed ranges horizontal 0 ≤ x ≤ 2.5, 0 ≤ y ≤ 1;

→ in this problem for fixed ranges we use the PlotRange → {{}, {}};

```
Animate[Graphics[Disk[{5 t * Cos[60 Degree], 5 t * Sin[60 Degree] - 1 / 2 * 9.8 * t^2}, 0.05],
  PlotRange → {{0, 2.5}, {0, 1}}], {t, 0, 1}]
```




```
topic : operations with polynomials;

→ for the operations of polynomials we find its Coefficient;

→ if you don ' t know about the what Coefficient
   is multiplied to its variable we use the command Expand;

Expand[(2 x + 4)^5]
```

$1024 + 2560\,x + 2560\,x^2 + 1280\,x^3 + 320\,x^4 + 32\,x^5$

```
→ but our topic here is about the Coefficient. which is actually a Command of mathematica;

?Coefficient
```

Coefficient[*expr*, *form*] gives the coefficient of *form* in the polynomial *expr*.
Coefficient[*expr*, *form*, *n*] gives the coefficient of *form* ^ *n* in *expr*. ≫

```
Q1 : find what multiplies x^4 in (2 x + 4)^5,
write three slightly different programmes to do the same;

exp = (2 x + 4)^5; Coefficient[exp, x^4]
```

320

```
Coefficient[exp, x^2^2]
```

320

```
Coefficient[exp, x, 4]
```

320

```
Coefficient[exp, x, 2^2]
```

320

```
Coefficient[exp,  x, 2 * 2]
```

320

```
Coefficient[exp, x^2, 2]
```

320

```
Coefficient[exp, x^(2*2)]
```

```
320
```

Q2 : Find terms independent of x in (2 x + 4) ^5;

```
Coefficient[exp, x, 0]
```

```
1024
```

Q3 : Find what multiplies x^2 y^4 in (2 x + y) ^6;

```
exp = (2 x + y)^6; Coefficient[exp, x^2*y^4]
```

```
60
```

```
Coefficient[exp, (x*y^2)^2]
```

```
60
```

→ for partial fraction we Aprt each fraction, the command for it Apart;

Q4 : Find partial fractions of 1 / (1 + x) (5 + x);

```
? Apart
```

Apart[*expr*] rewrites a rational expression as a sum of terms with minimal denominators.
Apart[*expr*, *var*] treats all variables other than *var* as constants. >>

```
Apart[1/(1 + x)*(5 + x)]
```

$$1+\frac{4}{1+x}$$

topic : Elements of a list;

```
? [[
```

expr[[*i*]] or Part[*expr*, *i*] gives the i^{th} part of *expr*.
expr[[−*i*]] counts from the end.
expr[[*i*, *j*, …]] or Part[*expr*, *i*, *j*, …] is equivalent to *expr*[[*i*]][[*j*]] ….
expr[[{i_1, i_2, …}]] gives a list of the parts i_1, i_2, … of *expr*.
expr[[*m* ;; *n*]] gives parts *m* through *n*.
expr[[*m* ;; *n* ;; *s*]] gives parts *m* through *n* in steps of *s*. >>

→ The elements of the lists can be accessed by usinyhg their indicies by using [[]];

Q1 : select 4 th element of the list (3, i, -7, h, Pi, 7 / 11);

```
f = {3, i, -7, h, Pi, 7/11}; f[[4]]
```

```
h
```

Q2 : select the first row and then 2, 1 element of the matrix;

```
k = {{1, 0, Pi}, {8, 3, 2}, {i, -5, 1.5}}
"select first row"
k[[1]]
"select 2,1 element of the matrix"
k[[2, 1]]
```

$\{\{1, 0, \pi\}, \{8, 3, 2\}, \{i, -5, 1.5\}\}$

select first row

$\{1, 0, \pi\}$

select 2,1 element of the matrix

8

Topic : Solutions of algebric equations ;

→ to solve this problem we the equation of the quadratic formula ;
→ the symbol == is used to specify the euqation .
→ solution of symbolic equations is done with the function Solve [] ,
which returns a list of replacement rules for the solution variables ,

Q1 : find the quadratic formula ;

```
Solve[a * x^2 + b * x + c == 0, x]
```

$$\left\{\left\{x \to \frac{-b - \sqrt{b^2 - 4 a c}}{2 a}\right\}, \left\{x \to \frac{-b + \sqrt{b^2 - 4 a c}}{2 a}\right\}\right\}$$

Q2 : find a zero of 2 x^2 + 3 x - 1;
to find the zero of the equation which demands
that to find the value of x we use the replacement operator ;

```
a = Solve[2 x^2 + 3 x - 1 == 0, x]
a[[2]]
x /. a[[1]]
```

$$\left\{\left\{x \to \frac{1}{4}\left(-3 - \sqrt{17}\right)\right\}, \left\{x \to \frac{1}{4}\left(-3 + \sqrt{17}\right)\right\}\right\}$$

$$\left\{x \to \frac{1}{4}\left(-3 + \sqrt{17}\right)\right\}$$

$$\frac{1}{4}\left(-3 - \sqrt{17}\right)$$

```
(*in above question there are two list of solutions,
we choose one solution by using the index of the list,
to find the zero of the solution we use replacement opeator;*)

(*Q3:solve the simultaneously equations ax+3y==7 and -3x+5y==9 for x and y*)

Solve[{a * x + 3 y == 7, -3 x + 5 y == 9}, {x, y}]
```

$$\left\{\left\{x \to \frac{8}{9 + 5 a}, y \to \frac{3 (7 + 3 a)}{9 + 5 a}\right\}\right\}$$

```
(*Q4:solve the system of equations 2x-3y==4 and 6x+7y==
  1 and verify that the results satisfies these equations*)
a = Solve[{2 x - 3 y == 4, 6 x + 7 y == 1}, {x, y}]
{2 x - 3 y, 6 x + 7 y} /. a[[1]]
```

$\left\{\left\{x \to \frac{31}{32},\ y \to -\frac{11}{16}\right\}\right\}$

{4, 1}

```
(*Q5:find sin[x]+x if x is a zero of 2x^2-3x+1;*)
k = Solve[2 x^2 - 3 x + 1 == 0]
x = x /. k[[1]]
Sin[x] + x
```

$\left\{\left\{x \to \frac{1}{2}\right\},\ \{x \to 1\}\right\}$

$\frac{1}{2}$

$\frac{1}{2} + \text{Sin}\left[\frac{1}{2}\right]$

```
(*in above question we first solve for 2x^2-
 3x+1 and find its any value and put in the Sin[x]+x*)
```

```
(*Q6:write a programme to calculate the range of
  a projectile of given (symbolic)  initial speed and angle*)
```

```
t =. ; p = Solve[v0 * t * Sin[th] - 1 / 2 * g * t^2 == 0, t]
t = t /. p[[2]]
v0 * t * Cos[th] // Simplify
```

$\left\{\{t \to 0\},\ \left\{t \to \frac{2\ \text{v0 Sin[th]}}{g}\right\}\right\}$

$\frac{2\ \text{v0 Sin[th]}}{g}$

$\frac{\text{v0}^2\ \text{Sin[2 th]}}{g}$

```
Clear[t, th]
```

```
(*Q7:write a subprogramme to calculate the range of a projectile
  as a function of its initial speed and angle. call the subprogramme
  for initial speed of 4.5 m/s and an arbitrary initial angle. use
  it to calculate the range of projectiles thrown with angles 0,
Pi/8,pi/4,....,Pi rad. with initial speed as 2 m/s;*)
```

```
R[v0_, th_] := (q = Solve[v0 * t * Sin[th] - 1 / 2 * g * t^2 == 0, t];
  t0 = t /. q[[2]]; v0 * t0 * Cos[th] // Simplify); R[5, th];
Do[Print[R[2, th]], {th, 0, Pi, Pi / 8}]
```

0

$\frac{2\sqrt{2}}{g}$

$\frac{4}{g}$

$\frac{2\sqrt{2}}{g}$

0

$-\frac{2\sqrt{2}}{g}$

$-\frac{4}{g}$

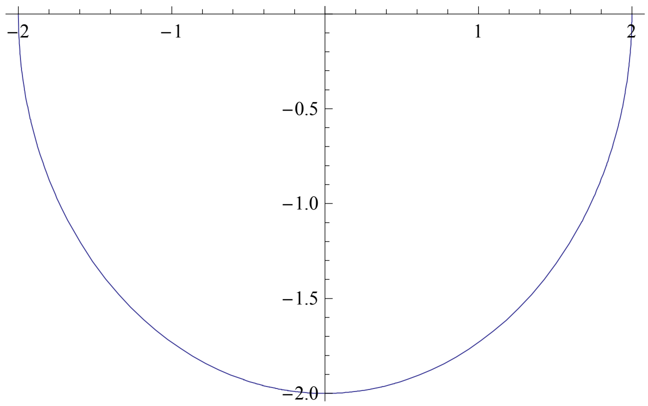
$-\frac{2\sqrt{2}}{g}$

0

```
(*Q7:graph a solution of x^2+y^2==4 for -2≤x≤2;*)

(*to solve this question we first solve it for y and then graph the solutionb for x*)

g = Solve[x^2 + y^2 == 4, y]
t = y /. g[[1]]
Plot[t, {x, -2, 2}]
```

$\left\{\left\{y \to -\sqrt{4 - x^2}\right\}, \left\{y \to \sqrt{4 - x^2}\right\}\right\}$

$-\sqrt{4 - x^2}$




```
(*Q8:find the cpacitance of two capacitors which have an equavilent capacitance 0.4
   microfarad when combined in series and 2 microfarad when combined in parallel;*)
```

```
Solve[{1 / c1 + 1 / c2 == 1 / (0.4 * 10 ^ -6), c1 + c2 == 2 * 10 ^ -6}, {c1, c2}]
```

$\{\{c1 \to 5.52786 \times 10^{-7},\ c2 \to 1.44721 \times 10^{-6}\},\ \{c1 \to 1.44721 \times 10^{-6},\ c2 \to 5.52786 \times 10^{-7}\}\}$

topic : Numerically Solving;

ther are two command for numerically solving 1) NSolve 2) FindRoot;

a useful feature of NSolve [] is that an initial guess for the root does not have to be given.NSolve[] is has a limited cababilities to solve some simple nonlinear equations;

the function FindRoot[] is designed for finding roots of nonlinear algebric equations or set of euqations. when using FindRoot[], an initial guess must be given for the position of the root;

```
(*Q1:find the equilibrium position of a particle under 2-3x+x^2+0.01e^-Sqrt[Sin[x]]
    newtons if it starts displacing away from equilibrium under 2-3x+x^2 newtons;*)
```

```
x0 = Solve[2 - 3 x + x^2 == 0, x]; a = x /. x0[[1]];
FindRoot[2 - 3 x + x^2 + 0.01 E^(-Sqrt[Sin[x]]), {x, a}]
```

$\{x \to 1.00401\}$

topic : power series;

```
(*to command for the power series is Series[]*)
```

```
(*Q1:Expand Sin[x] around pi/4 till order
   3. Find the numerical value of the expansion for x=1.1*)
```

```
? Series
```

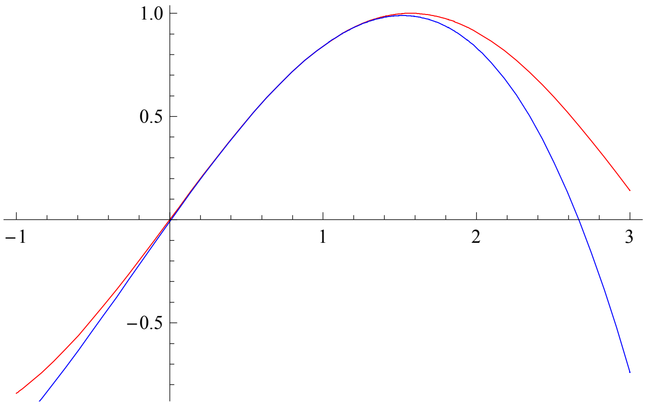
Series[f, {x, x_0, n}] generates a power series expansion for f about the point $x = x_0$ to order $(x - x_0)^n$.
Series[f, {x, x_0, n_x}, {y, y_0, n_y}] successively finds series expansions with respect to x, then y. ≫

```
a = Normal[Series[Sin[x], {x, Pi / 4, 3}]]
k = a /. x -> 1.1
```

$$\frac{1}{\sqrt{2}} + \frac{-\frac{\pi}{4} + x}{\sqrt{2}} - \frac{\left(-\frac{\pi}{4} + x\right)^2}{2\sqrt{2}} - \frac{\left(-\frac{\pi}{4} + x\right)^3}{6\sqrt{2}}$$

0.890902

```
(*graph the original function as red and its power series expansion for -1≤
 x≤3 as blue, and display together*)
g1 = Plot[Sin[x], {x, -1, 3}, PlotStyle → Red];
g2 = Plot[a, {x, -1, 3}, PlotStyle → Blue];
Show[g1, g2]
```




```
(*Q2:find the expression for the position dependence of a
   potential energy near equilibrium position of a particle upto second
   order. extract the effective spring constant using a coefficient;*)
v'[0] = 0; a = Normal[Series[v[x], {x, 0, 2}]]
(*by choosing origin at equilibrium*)
"effective spring constant"
Coefficient[a, x^2 / 2]
```

$\frac{1}{2} x^2 v''[0]$

```
effective spring constant
```

```
v''[0]
```

```
Clear[a]
```

```
topic : Integration;
```

```
? Integrate
```

Integrate[f, x] gives the indefinite integral $\int f\, dx$.

Integrate[f, {x, x_{min}, x_{max}}] gives the definite integral $\int_{x_{min}}^{x_{max}} f\, dx$.

Integrate[f, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}, …] gives the multiple integral $\int_{x_{min}}^{x_{max}} dx \int_{y_{min}}^{y_{max}} dy \ldots f$. ≫

```
(*Q1:integrate 1/x wiht respect to x;*)
```

```
Integrate[1 / x, x]
```

```
Log[x]
```

```
(*Q2:integrate 1/x wiht respect to x from 1 to 4*)
```

```
Integrate[1 / x, {x, 1, 4}]
```

```
Log[4]
```

```
(*Q3:find e^-ax^2 for positive a;*)
```

```
f = Integrate[Exp[-a * x^2], {x, -Infinity, Infinity}]; Simplify[f, a > 0]
```

$$\frac{\sqrt{\pi}}{\sqrt{a}}$$

```
(*Q4:find e^-4x^2 for l=.2,.4,.6,....,
4 and compare with e^-4x^2 from - infinity to infinity*)

Do[Print[Integrate[Exp[-4 x^2], {x, -l, l}]], {l, .2, 4, .2}]
"Compare"
Integrate[Exp[-4 x^2], {x, -Infinity, Infinity}] // N
```

```
0.379653
0.65767
0.806745
0.865266
0.882081
0.885617
0.88616
0.886222
0.886227
0.886227
0.886227
0.886227
0.886227
0.886227
0.886227
0.886227
0.886227
0.886227
0.886227
0.886227
```

Compare

0.886227

```
(*Q5:
  calculate the work done by electrostatic force n increasing their separation ri to rf
    along the direction of the force. assume ri and rf to be positive and rf<ri;*)

Simplify[Integrate[k * q1 * q2 / r^2, {r, ri, rf}], {rf < ri, rf > 0, ri > 0}]
```

$$k\ q1\ q2\left(-\frac{1}{rf}+\frac{1}{ri}\right)$$

```
(*Q6:numerically calculate the work done by a spring force with such a spring constant
   that the force magnitude for an extension of 3cm is 5N from 7 cm to 11 cm;*)
```

```
k =. ; g = Solve[5 == k * 3 * 10^-2, k]; k = k /. g[[1]];
NIntegrate[-k * x, {x, 7 * 10^-2, 11 * 10^-2}]
```

-0.6

```
(*Q7:calculate Sqrt[x^2+y^2]*)

Integrate[Sqrt[x^2 + y^2], {y, 0, x}, {x, 0, 1}]
```

$$\frac{1}{6}\left(\sqrt{2}+\text{Log}\left[1+\sqrt{2}\right]\right)$$

```
(*Q8:calculate total charge in the region -1≤x≤2,
0<=y≤1 meters, if the charge density thereis 2.3Sin[x*y]c/m^2*)

Integrate[2.3 Sin[x * y], {y, 0, 1}, {x, -1, 2}]
```

1.39741

```
(*Q9:calculate total charge in the region 0≤x≤y,
0<=y≤1 meters, if the charge density thereis 2.3Sin[x*y]c/m^2*)

Integrate[2.3 Sin[x * y], {y, 0, 1}, {x, 0, y}]
```

0.275784

```
topic : Sums and products;

? Sum
```

Sum[*f*, {*i*, i_{max}}] evaluates the sum $\sum_{i=1}^{i_{max}} f$.

Sum[*f*, {*i*, i_{min}, i_{max}}] starts with $i = i_{min}$.

Sum[*f*, {*i*, i_{min}, i_{max}, *di*}] uses steps di.

Sum[*expr*, {*i*, {i_1, i_2, …}}] uses successive values i_1, i_2, ….

Sum[*f*, {*i*, i_{min}, i_{max}}, {*j*, j_{min}, j_{max}}, …] evaluates the multiple sum $\sum_{i=i_{min}}^{i_{max}} \sum_{j=j_{min}}^{j_{max}} \ldots f$. »

```
Q : find summations 5 (-1)^n / 4^n;

Sum[5 * (-1)^n / 4^n, {n, 0, Infinity}]
```

4

```
(*Q2:find the total electric potential energy of a system of charges qn=
 0.0003n^2(-1)n+1in C, with n=1,
2,.....20. the charges are kept at positions n/20 in meter*)

Sum[9 * 10^9 * 0.0003 n^2 * (-1)^(n + 1) * 0.0003 m^2 * (-1)^(m + 1) / Abs[n / 20 - m / 20],
 {n, 1, 19}, {m, n + 1, 20}]
```

-7.41201×10^{9}

```
(*Q3:find the numerical value of e^Sin[x] from x=1 to 10 for each x being an integer*)
? Product
```

Product[f, {i, i_{max}}] evaluates the product $\prod_{i=1}^{i_{max}} f$.
Product[f, {i, i_{min}, i_{max}}] starts with $i = i_{min}$.
Product[f, {i, i_{min}, i_{max}, di}] uses steps di.
Product[$expr$, {i, {i_1, i_2, ...}}] uses successive values i_1, i_2, ...
Product[f, {i, i_{min}, i_{max}}, {j, j_{min}, j_{max}}, ...] evaluates the multiple product $\prod_{i=i_{min}}^{i_{max}} \prod_{j=j_{min}}^{j_{max}} \ldots f$. ≫

```
Product[Exp[Sin[x]], {x, 1, 10}] // N
```

4.10083

```
topic : optimization;

? Max
```

Max[x_1, x_2, ...] yields the numerically largest of the x_i.
Max[{x_1, x_2, ...}, {y_1, ...}, ...] yields the largest element of any of the lists. ≫

```
? Min
```

Min[x_1, x_2, ...] yields the numerically smallest of the x_i.
Min[{x_1, x_2, ...}, {y_1, ...}, ...] yields the smallest element of any of the lists. ≫

```
(*Q2:find the total electric potential energy of a system of charges qn=
 0.0003n^2(-1)n+1in C, with n=1,
2,.....20. the charges are kept at positions n/20 in meter*)

f = Table[9 * 10 ^ 9 * 0.0003 n ^ 2 * (-1) ^ (n + 1) * 0.0003 m ^ 2 * (-1) ^ (m + 1) / Abs[n / 20 - m / 20],
  {m, 1, 19}, {n, m + 1, 20}]; Max[f]
Min[
 f]
```

1.04976×10^9

-2.33928×10^9

```
(*Q1:find the largest and smallest number 2.3,Pi,1,1/3,-0.01;*)

Max[2.3, Pi, 1, 1 / 3, -0.01]
```

π

```
Min[2.3, Pi, 1, 1 / 3, -0.01]
```

-0.01

```
? Maximize
```

Maximize[f, {x, y, ...}] maximizes f with respect to x, y, ...
Maximize[{f, $cons$}, {x, y, ...}] maximizes f subject to the constraints $cons$.
Maximize[{f, $cons$}, {x, y, ...}, dom] maximizes with variables over the domain dom, typically Reals or Integers. ≫

```
? Do
```

Do[$expr$, {i_{max}}] evaluates $expr$ i_{max} times.
Do[$expr$, {i, i_{max}}] evaluates $expr$ with the variable i successively taking on the values 1 through i_{max} (in steps of 1).
Do[$expr$, {i, i_{min}, i_{max}}] starts with $i = i_{min}$.
Do[$expr$, {i, i_{min}, i_{max}, di}] uses steps di.
Do[$expr$, {i, {i_1, i_2, …}}] uses the successive values i_1, i_2, ….
Do[$expr$, {i, i_{min}, i_{max}}, {j, j_{min}, j_{max}}, …] evaluates $expr$ looping over different values of j, etc. for each i. ≫

```
? Table
```

Table[$expr$, {i_{max}}] generates a list of i_{max} copies of $expr$.
Table[$expr$, {i, i_{max}}] generates a list of the values of $expr$ when i runs from 1 to i_{max}.
Table[$expr$, {i, i_{min}, i_{max}}] starts with $i = i_{min}$.
Table[$expr$, {i, i_{min}, i_{max}, di}] uses steps di.
Table[$expr$, {i, {i_1, i_2, …}}] uses the successive values i_1, i_2, ….
Table[$expr$, {i, i_{min}, i_{max}}, {j, j_{min}, j_{max}}, …] gives a nested list. The list associated with i is outermost. ≫

```
? Minimize
```

Minimize[f, {x, y, …}] minimizes f with respect to x, y, ...
Minimize[{f, $cons$}, {x, y, …}] minimizes f subject to the constraints $cons$.
Minimize[{f, $cons$}, {x, y, …}, dom] minimizes with variables over the domain dom, typically Reals or Integers. ≫

```
? NMinimize
```

NMinimize[f, {x, y, …}] minimizes f numerically with respect to x, y, ….
NMinimize[{f, $cons$}, {x, y, …}] minimizes f numerically subject to the constraints $cons$. ≫

```
? FindMinimum
```

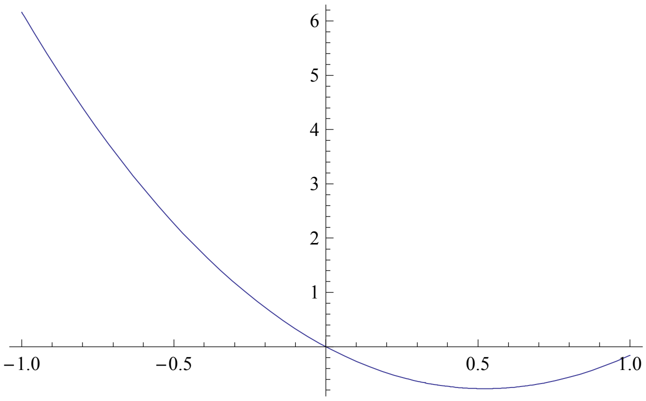
FindMinimum[f, {x, x_0}] searches for a local minimum in f, starting from the point $x = x_0$.
FindMinimum[f, {{x, x_0}, {y, y_0}, …}] searches for a local minimum in a function of several variables.
FindMinimum[{f, $cons$}, {{x, x_0}, {y, y_0}, …}] searches for a local minimum subject to the constraints $cons$.
FindMinimum[{f, $cons$}, {x, y, …}] starts from a point within the region defined by the constraints. ≫

```
(*Q3:find the local minimum of 3x^2-4x+
  Sin[x] near 1 and verify it satisfies the first and second derivative test*)
```

```
? Plot
```

Plot[f, {x, x_{min}, x_{max}}] generates a plot of f as a function of x from x_{min} to x_{max}.
Plot[{f_1, f_2, …}, {x, x_{min}, x_{max}}] plots several functions f_i. ≫

```
f = Plot[3 x^2 - 4 x + Sin[x], {x, -1, 1}]
```




```
f =.
```

```
f = 3 x^2 - 4 x + Sin[x]; l = FindMinimum[f, {x, 1}];
z = x /. l[[2]]
"first derivative test"
D[f, x] /. l[[2]]
"second derivative test"
If[(D[f, {x, 2}] /. l[[2]]) > 0, "Pass", "Fail"]
```

```
0.522214
```

```
first derivative test
```

-4.54712×10^{-10}

```
second derivative test
```

```
Pass
```

```
f =.
```

```
Clear[x]
```

```
? Sign
```

Sign[x] gives -1, 0 or 1 depending on whether x is negative, zero, or positive. ≫

```
? If
```

If[$condition$, t, f] gives t if $condition$ evaluates to True, and f if it evaluates to False.
If[$condition$, t, f, u] gives u if $condition$ evaluates to neither True nor False. ≫

```
(*Q4:find the equilibrium position of a charge q=2.1*10^-6C placed between q1=
  q2=4.1*10^-6C kept 3m apart using energy considerations. display
    the resulting energy. make sure that q is between q1 and q2:*)
```

```
q1 = q2 = 4.1 * 10 ^ -6; q3 = 2.1 * 10 ^ -6; f = Minimize[
  {9 * 10 ^ 9 (q1 * q3 / r + q3 * q2 / (3 * 10 ^ -2 - r) + q1 * q2 / (3 * 10 ^ -2)), 3 * 10 ^ -2 > r > 0}, r]
k = r /. f[[2]]
"energy"
f[[1]]
```

```
{15.375, {r → 0.015}}
```

```
0.015
```

```
energy
```

```
15.375
```

```
(*Q5:find where a charge willbe in equilibrium if it starts motion from
  (1,1,1) in a potential field 0.5x^2*y^2+2.5y^2*z^2+5y^2(z-1)^2+2.5*)
```

```
f = FindMinimum[0.5 x ^ 2 y ^ 2 + 2.5 y ^ 2 z ^ 2 + 5 y ^ 2 (z - 1) ^ 2 + 2.5, {{x, 1}, {y, 1}, {z, 1}}];
Print["(", x /. f[[2]], ",", y /. f[[2]], ",", z /. f[[2]], ")"]
```

$(0.870266, 4.52366 \times 10^{-14}, 0.496847)$

```
calculus of variations;
```

```
? Minimize
```

Minimize[f, {x, y, …}] minimizes f with respect to x, y, ….
Minimize[{f, $cons$}, {x, y, …}] minimizes f subject to the constraints $cons$.
Minimize[{f, $cons$}, {x, y, …}, dom] minimizes with variables over the domain dom, typically Reals or Integers. ≫

```
? NMinimize
```

NMinimize[f, {x, y, …}] minimizes f numerically with respect to x, y, ….
NMinimize[{f, $cons$}, {x, y, …}] minimizes f numerically subject to the constraints $cons$. ≫

```
? FindMinimum
```

FindMinimum[f, {x, x_0}] searches for a local minimum in f, starting from the point $x = x_0$.
FindMinimum[f, {{x, x_0}, {y, y_0}, …}] searches for a local minimum in a function of several variables.
FindMinimum[{f, $cons$}, {{x, x_0}, {y, y_0}, …}] searches for a local minimum subject to the constraints $cons$.
FindMinimum[{f, $cons$}, {x, y, …}] starts from a point within the region defined by the constraints. ≫

```
? Integrate
```

Integrate[f, x] gives the indefinite integral $\int f\,dx$.

Integrate[f, {x, x_{min}, x_{max}}] gives the definite integral $\int_{x_{min}}^{x_{max}} f\,dx$.

Integrate[f, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}, …] gives the multiple integral $\int_{x_{min}}^{x_{max}} dx \int_{y_{min}}^{y_{max}} dy \ldots f$. ≫

```
(*use the variation ax(1-x) to y=
 x to show that the curve of minmal length connecting (0,0) to (1,1) is y=x itself*)
y = x + a * x (x - 1) ; f = NMinimize[Integrate[Sqrt[1 + D[y, x] ^ 2], {x, 0, 1}], a]
"variation parameter"
a /. f[[2]]
"minimal length"
f[[1]]
```

$\{1.41421, \{a \to 8.60412 \times 10^{-8}\}\}$

variation parameter

8.60412×10^{-8}

minimal length

1.41421

```
topic : interpolation;
```

```
(*find a collection of parabolas that passing through the {3.1,5.8},{2.3,4.4},
{1.4,6.1},{1.2,7.3}. evaluate the collection at independent variable=
 1.9 and -1. make a graph of the derivative *)
data = {{3.1, 5.8}, {2.3, 4.4}, {1.4, 6.1}, {1.2, 7.3}};
g1 = ListPlot[data, PlotRange → All];
f = Interpolation[data, InterpolationOrder → 2]
f[1.9]
"extrapolation"
f[-1]
g2 = Plot[f[x], {x, 1.2, 3.1}, PlotRange → All];
"the graphs"
Show[g1, g2]
"the derivative of collections"
Plot[f '[x], {x, 1.2, 3.1}, PlotRange → All]
"area under the curve"
Integrate[f[x], {x, 1.2, 3.1}]
```

InterpolatingFunction[{{1.2, 3.1}}, <>]

4.40808

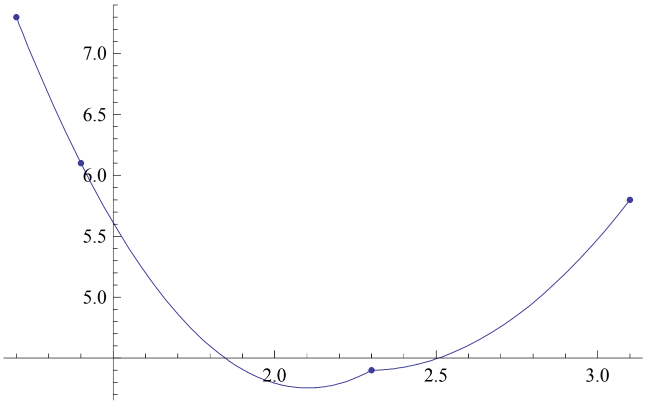
extrapolation

InterpolatingFunction::dmval :

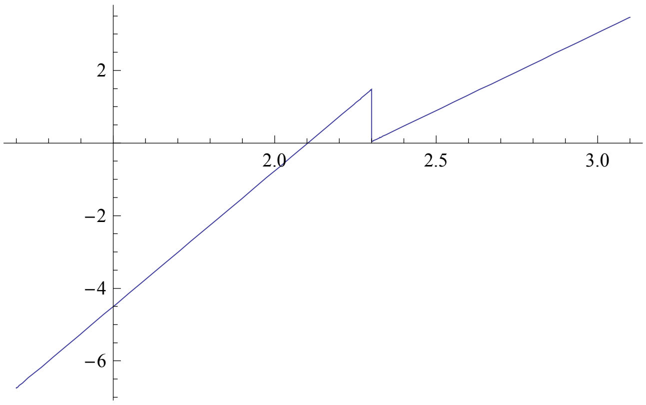
Input value {−1} lies outside the range of data in the interpolating function. Extrapolation will be used. ≫

40.2333

the graphs




```
the derivative of collections
```




```
area under the curve
```

```
9.50327
```

? D

D[f, x] gives the partial derivative $\partial f / \partial x$.
D[f, {x, n}] gives the multiple derivative $\partial^n f / \partial x^n$.
D[f, x, y, …] differentiates f successively with respect to x, y, ...
D[f, {{x_1, x_2, …}}] for a scalar f gives the vector derivative ($\partial f / \partial x_1$, $\partial f / \partial x_2$, ...). ≫

? Integrate

Integrate[f, x] gives the indefinite integral $\int f \, dx$.

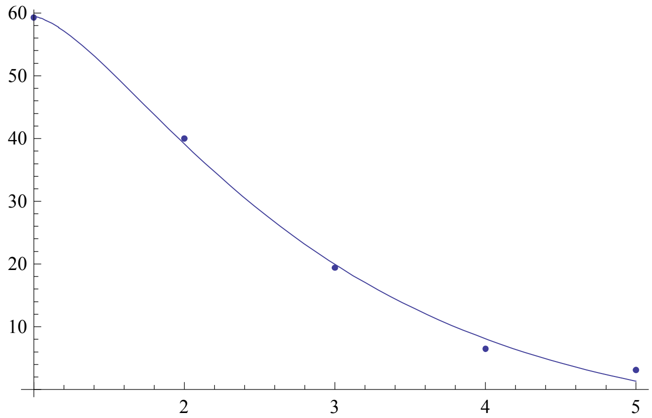
Integrate[f, {x, x_{min}, x_{max}}] gives the definite integral $\int_{x_{min}}^{x_{max}} f \, dx$.

Integrate[f, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}, …] gives the multiple integral $\int_{x_{min}}^{x_{max}} dx \int_{y_{min}}^{y_{max}} dy \ldots f$. ≫

```
? FindFit
```

FindFit[*data*, *expr*, *pars*, *vars*] finds numerical values of the parameters *pars* that make *expr* give a best fit to *data* as a function of *vars*. The data can have the form $\{\{x_1, y_1, \ldots, f_1\}, \{x_2, y_2, \ldots, f_2\}, \ldots\}$, where the number of coordinates *x*, *y*, ... is equal to the number of variables in the list *vars*. The data can also be of the form $\{f_1, f_2, \ldots\}$, with a single coordinate assumed to take values 1, 2, ….
FindFit[*data*, {*expr*, *cons*}, *pars*, *vars*] finds a best fit subject to the parameter constraints *cons*. ≫

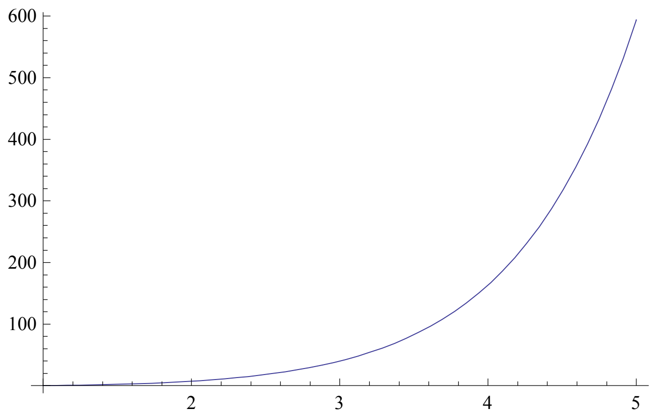
```
(*Q1:model the data {1,59.3},{2,40},{3,19.4},{4,6.5},
{5,3.1} as a function of x of form a/x+b*Exp[-c*x] with least chi-
 square deviation. graph the data and the model together*)

data = {{1, 59.3}, {2, 40}, {3, 19.4}, {4, 6.5}, {5, 3.1}};
g1 = ListPlot[data];
l = FindFit[data, a / x + b * Exp[-c * x], {a, b, c}, x];
g2 = Plot[a / x + b * Exp[-c * x] /. l, {x, 1, 5}];
Show[g1, g2]
```




```
Clear[g1, g2, l]

? Pochhammer
```

Pochhammer[*a*, *n*] gives the Pochhammer symbol $(a)_n$. ≫

```
Plot[(x - 1) * Exp[x], {x, 1, 5}]
```



```
? Sum
```

Sum[f, {i, i_{max}}] evaluates the sum $\sum_{i=1}^{i_{max}} f$.

Sum[f, {i, i_{min}, i_{max}}] starts with $i = i_{min}$.
Sum[f, {i, i_{min}, i_{max}, di}] uses steps di.
Sum[$expr$, {i, {i_1, i_2, …}}] uses successive values i_1, i_2, ….

Sum[f, {i, i_{min}, i_{max}}, {j, j_{min}, j_{max}}, …] evaluates the multiple sum $\sum_{i=i_{min}}^{i_{max}} \sum_{j=j_{min}}^{j_{max}} \ldots f$. ≫

```
data = {{1, 2}, {3, 4}}
data[[1, 2]]
data[[1, 1]]
```

```
{{1, 2}, {3, 4}}

2

1
```

```
(*generate the chi square difference between the data {1.1,4.2},{2.3,5.7},
{1.8,6.2},{0.9,1.7} and the model a/x+bx. find the values of a and b for
  which the chi square is minimum. compare with the direct output of FindFit*)
```

```
data = {{1.1, 4.2}, {2.3, 5.7}, {1.8, 6.2}, {0.9, 1.7}};
model = a / x + b * x;
chisq = Sum[(data[[i, 2]] - model /. x → data[[i, 1]])^2, {i, 1, Length[data]}];
f = Minimize[chisq, {a, b}]
"a="
a /. f[[2]]
"b="
b /. f[[2]]
"compare"
FindFit[data, model, {a, b}, x]
```

```
{3.73773, {a → 0.0922429, b → 2.84844}}

a=

0.0922429

b=

2.84844

compare

{a → 0.0922429, b → 2.84844}
```

```
? Length
```

Length[$expr$] gives the number of elements in $expr$. ≫