Mahek Khurana
MAT/19/63

Chapter-3 Functions and Their Graphs

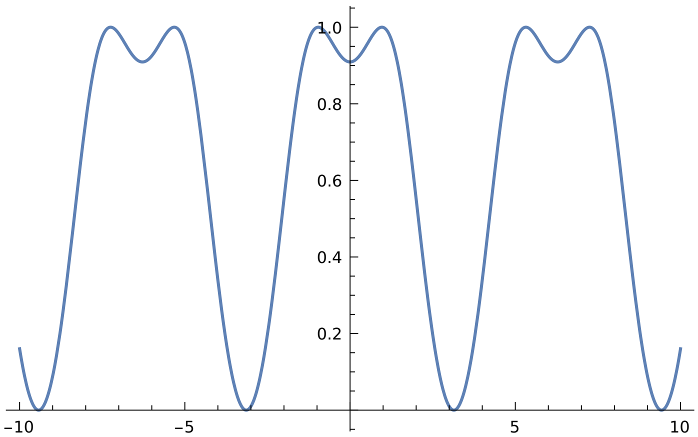
Exercise 3.2

Ques 1.Plot the following functions on the domain 10 ≤ x ≤ 10.

a. sin(1 + cos(x))

In[5]:=
```
Plot[Sin[1 + Cos[x]], {x, -10, 10}]
```

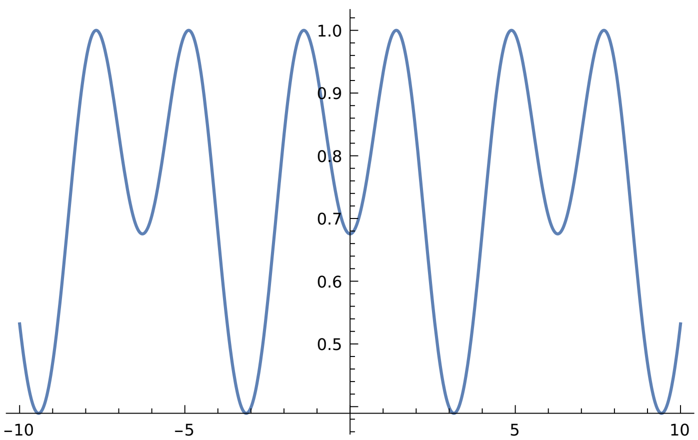
Out[5]=



b. sin(1.4 + cos(x))

In[6]:=
```
Plot[Sin[1.4 + Cos[x]], {x, -10, 10}]
```

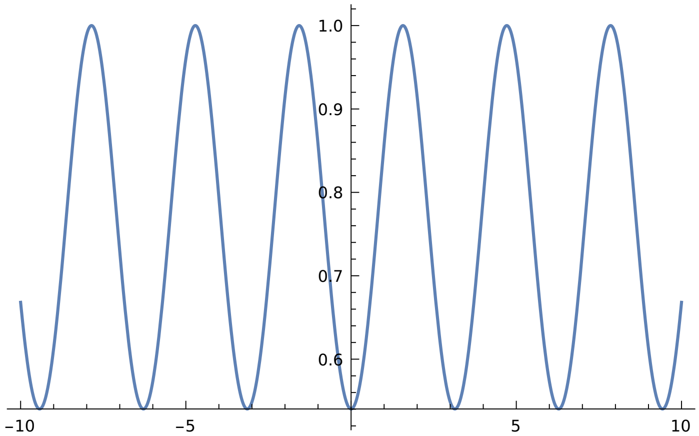
Out[6]=



c. sin(Pi/2 + cos(x))

In[7]:=
```
Plot[Sin[Pi / 2 + Cos[x]], {x, -10, 10}]
```

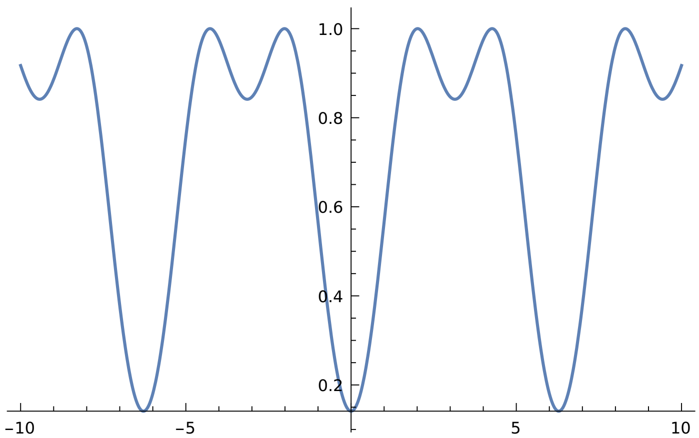
Out[7]=



d. sin(2 + cos(x))

In[8]:=
```
Plot[Sin[2 + Cos[x]], {x, -10, 10}]
```

Out[8]=



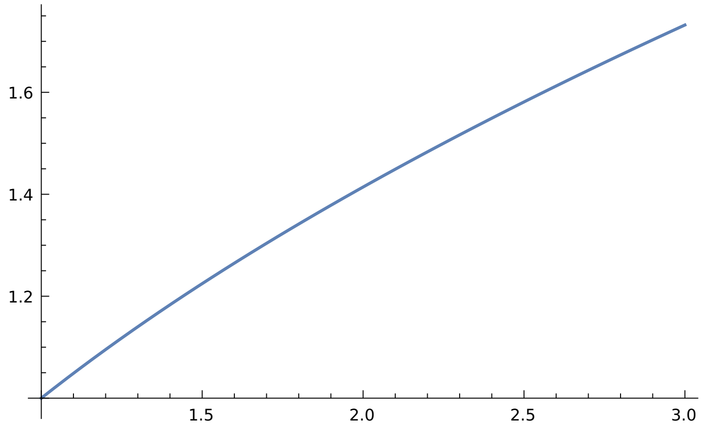
Ques 2.One can zoom in toward a particular point in the domain of a function and see how the graph
appears at different zoom levels. For instance, consider the square root function f(X) = sqrt(x) when
x is near 2.

a. Enter the input below to see the graph of f as x goes from 1 to 3.
With[{δ=10^0},Plot[f[x],{x,2-δ,2+δ}]]

In[38]:=
```
f[x_] := Sqrt[x]
```

In[39]:= `With[{δ = 10 ^ 0}, Plot[f[x], {x, 2 - δ, 2 + δ}]]`

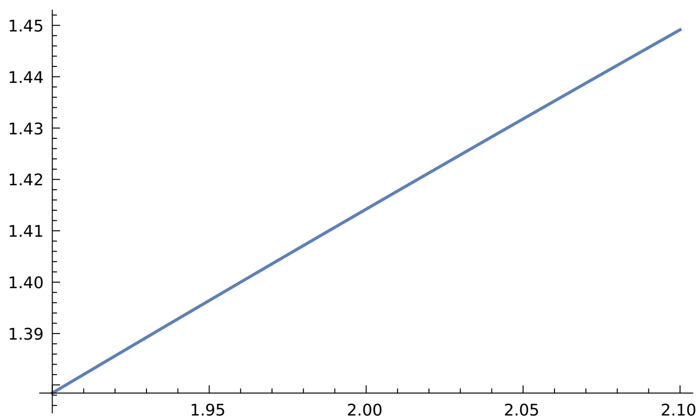
Out[39]=



b. Now zoom; change the value of δ to be 10^(-1) and re-enter the input above to see the graph of f as x goes from 1.9 to 2.1. Do this again for δ 10^(-2) , 10^(-3) , 10(-4) , and 10(-5).

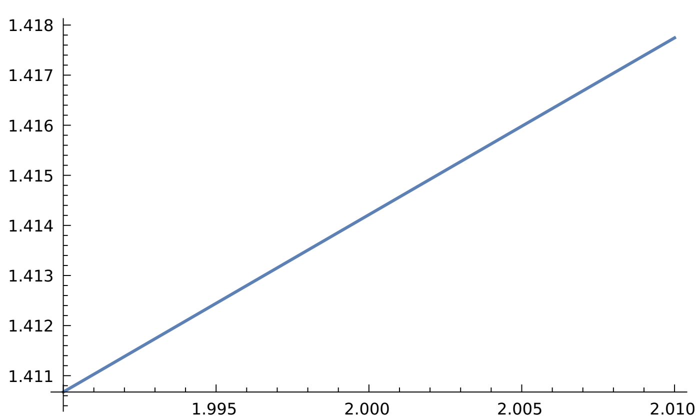
In[40]:= `With[{δ = 10 ^ (- 1)}, Plot[f[x], {x, 2 - δ, 2 + δ}]]`

Out[40]=

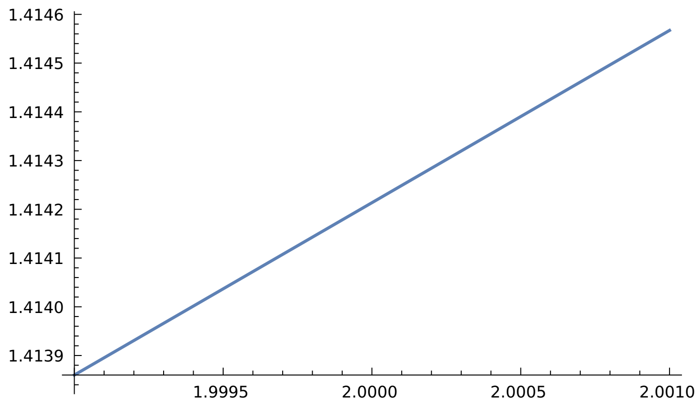


In[42]:= `With[{δ = 10 ^ (- 2)}, Plot[f[x], {x, 2 - δ, 2 + δ}]]`

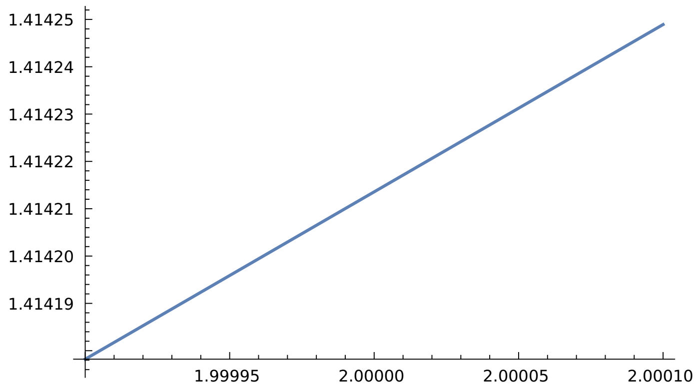
Out[42]=



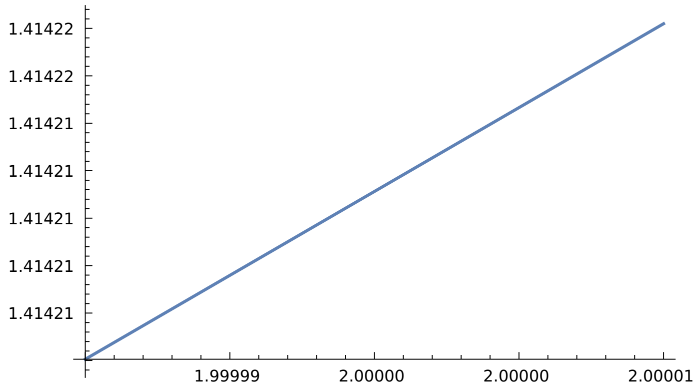
In[43]:= With[{δ = 10 ^ (-3)}, Plot[f[x], {x, 2 - δ, 2 + δ}]]

Out[43]=


In[44]:= With[{δ = 10 ^ (-4)}, Plot[f[x], {x, 2 - δ, 2 + δ}]]

Out[44]=


In[45]:= With[{δ = 10 ^ (-5)}, Plot[f[x], {x, 2 - δ, 2 + δ}]]

Out[45]=


c. Use the last plot to approximate sqrt(2) to six significant digits. Check your answer using N.

In[49]:= With[{δ = 10 ^ (-10)}, Plot[f[x], {x, 2 - δ, 2 + δ}]]

From the above plot we can clearly observe that the approximate value of sqrt(2) is 1.41421

In[50]:= N[Sqrt[2]]

Out[50]= 1.41421

d. When making a Plot, the lower and upper bounds on the iterator must be distinct when rounded to machine precision. Enter the previous Plot command with δ = 10(-20). An error message results. Read the error message and speculate as to what is happening. The bottom line is that zooming has its limits.

In[51]:=
```
With[{δ = 10^(-20)}, Plot[f[x], {x, 2 - δ, 2 + δ}]]
```

Plot : Endpoints for x in $\left\{x, \frac{1999999999999999999999}{1000000000000000000000}, \frac{2000000000000000000001}{1000000000000000000000}\right\}$ must have distinct machine −precision numerical values .

Plot : Endpoints for x in $\left\{x, \frac{1999999999999999999999}{1000000000000000000000}, \frac{2000000000000000000001}{1000000000000000000000}\right\}$ must have distinct machine −precision numerical values .

Plot : Endpoints for x in $\left\{x, \frac{1999999999999999999999}{1000000000000000000000}, \frac{2000000000000000000001}{1000000000000000000000}\right\}$ must have distinct machine −precision numerical values .

General : Further output of Plot::plld will be suppressed during this calculation .

Plot : Endpoints for x in $\left\{x, \frac{1999999999999999999999}{1000000000000000000000}, \frac{2000000000000000000001}{1000000000000000000000}\right\}$ must have distinct machine −precision numerical values .

Plot : Endpoints for x in $\left\{x, \frac{1999999999999999999999}{1000000000000000000000}, \frac{2000000000000000000001}{1000000000000000000000}\right\}$ must have distinct machine −precision numerical values .

Plot : Endpoints for x in $\left\{x, \frac{1999999999999999999999}{1000000000000000000000}, \frac{2000000000000000000001}{1000000000000000000000}\right\}$ must have distinct machine −precision numerical values .

General : Further output of Plot::plld will be suppressed during this calculation .

Out[51]= $\text{Plot}\left[f[x], \left\{x, 2 - \frac{1}{100\,000\,000\,000\,000\,000\,000}, 2 + \frac{1}{100\,000\,000\,000\,000\,000\,000}\right\}\right]$

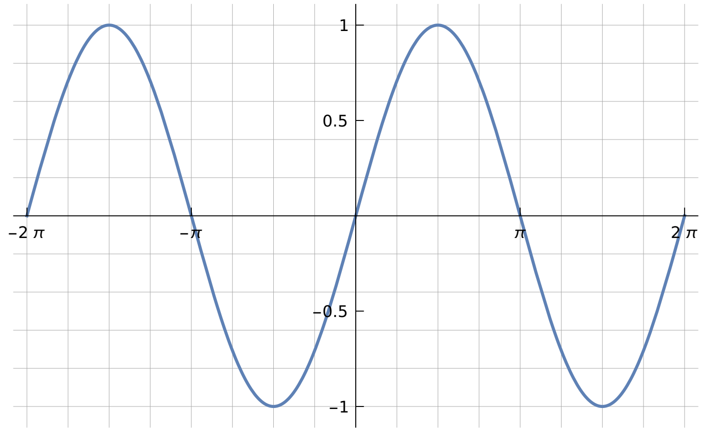
Clearly the lower and upper bound are so small that they tends to become equal and hence the iterator has the same value i.e. the iterator does'nt have distinct when value when rounded to machine-precision numerical value.

Exercise 3.3

Ques 1. Use the GridLines and Ticks options, as well as the setting GridLinesStyle ->Lighter[Gray],to produce the Plot of the sine function.

In[17]:=
```
Plot[Sin[x], {x, -2 Pi, 2 Pi}, GridLines → {Range[-2 Pi, 2 Pi, Pi / 4], Range[-1, 1, 0.2]},
 Ticks → {{-2 Pi, -Pi, Pi, 2 Pi}, {-1, -0.5, 0.5, 1}}, GridLinesStyle → Lighter[Gray]]
```

Out[17]=

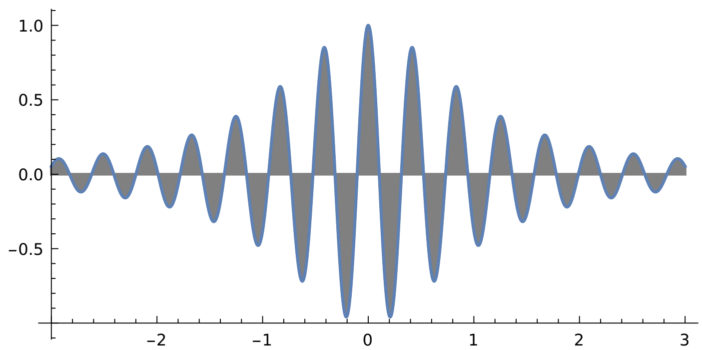


Ques2. Use the Axes, Frame, Filling, FrameStyle, PlotRange, and AspectRatio options to produce the following plot of the function y = cos(15x)/(1+x^2).

In[73]:=
```
F[x_] := Cos[15 x] / (1 + x ^ 2)
```

In[74]:=
```
Plot[F[x], {x, -3, 3}, Axes → True, AxesOrigin → {-3, -1}, PlotRange → Full, Filling → 0.0,
 FillingStyle → {Gray}, Frame → False, FrameStyle → False, AspectRatio → 0.5]
```

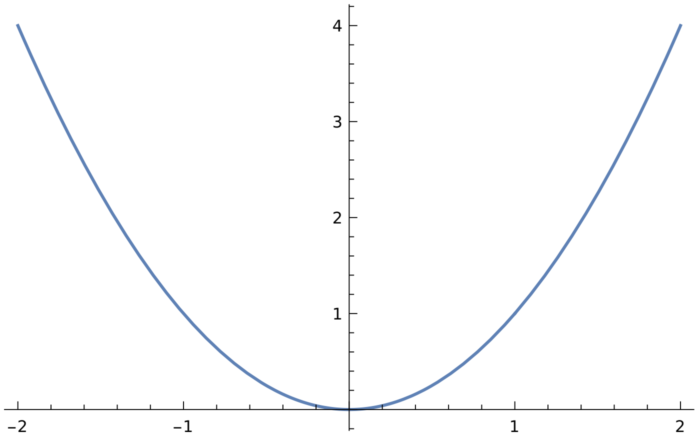
Out[74]=



Ques 3 . Plot the function f (x) = x^2 on the domain -2 ≤ x ≤ 2, and set Exclusions to [x == 1]. Note that f has no vertical asymptote at x 1. What happens?

In[59]:=
```
Plot[x^2, {x, -2, 2}, Exclusions → {x == 1}]
```

Out[59]=



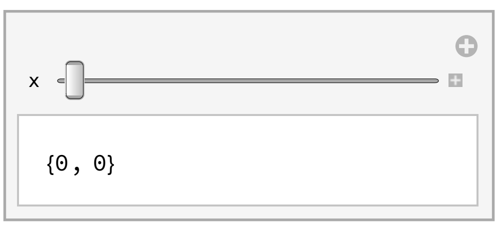
Since the graph is continuous at x=1.Therefore it has no vertical asymptotes.

Exercise 3.4

Ques 1. Manipulate[{x,y},{x,0,1},{y,0,1}]
The following simple Manipulate has two sliders: one for x and one for y. Make a Manipulate that also has output {x,y}, but that has a single Slider2D controller.

In[58]:=
```
Manipulate[{x, 2 x}, {x, 0, 1}]
```

Out[58]=

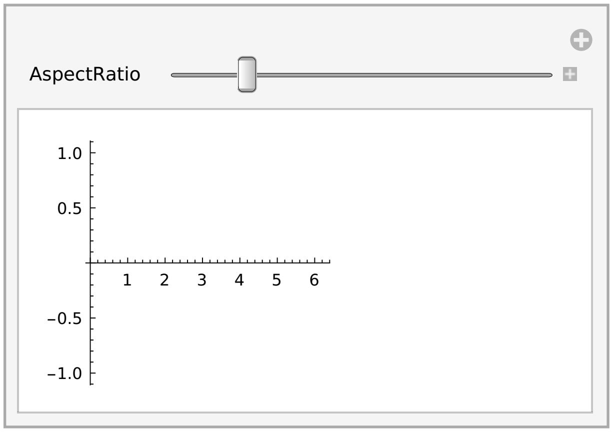


Ques 2. Make a Manipulate of a Plot where the user can adjust the AspectRatio in real time, from a starting value of 1/5 (five times as wide as it is tall) to an ending value of 5 (five times as tall as it is wide). Set ImageSize to Automatic, 128 so the height remains constant as the slider is moved.

In[108]:=
```
f[x_] := Sin[x];
```

In[110]:=
```
Manipulate[Plot[f[x], {x, 0, 2 Pi}, ImageSize → {Automatic , 128}, AspectRatio → k],
 {{k, 1, "AspectRatio "}, 0.2, 5}]
```

Out[110]=



Exercise 3.5

Ques 1.The Partition command is used to break a single list into sublists of equal length. It is useful for breaking up a list into rows for display within a Grid.

a. Enter the following inputs and discuss the outputs.

Range[100]

Partition[Range[100], 10]

In[90]:=
```
Range[100]
```

Out[90]=
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100}
```

In[91]:=
```
Partition[Range[100], 10]
```

Out[91]=
```
{{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, {11, 12, 13, 14, 15, 16, 17, 18, 19, 20},
 {21, 22, 23, 24, 25, 26, 27, 28, 29, 30}, {31, 32, 33, 34, 35, 36, 37, 38, 39, 40},
 {41, 42, 43, 44, 45, 46, 47, 48, 49, 50}, {51, 52, 53, 54, 55, 56, 57, 58, 59, 60},
 {61, 62, 63, 64, 65, 66, 67, 68, 69, 70}, {71, 72, 73, 74, 75, 76, 77, 78, 79, 80},
 {81, 82, 83, 84, 85, 86, 87, 88, 89, 90}, {91, 92, 93, 94, 95, 96, 97, 98, 99, 100}}
```

The given list of 100 elements breaks into sublists of 10 elements.

b. Format a table of the first 100 integers, with twenty digits per row.

In[103]:=
```
Data = Partition[Range[100], 20]
```

Out[103]=
```
{{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20},
 {21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40},
 {41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60},
 {61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80},
 {81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100}}
```

In[104]:=
```
Grid[Data]
```

Out[104]=

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

c. Make the same table as above, but use only the Table and Range commands. Do not use Partition.

In[165]:=
```
Data = Table[Range[20] + x, {x, 0, 90, 20}]
```

Out[165]=
```
{{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20},
 {21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40},
 {41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60},
 {61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80},
 {81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100}}
```

In[166]:=
```
Grid[Data]
```

Out[166]=

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

d. Make the same table as above, but use only the Table command (twice). Do not use Partition or Range.

In[169]:=
```
Table[x, {x, 1, 20}]
```

Out[169]=
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
```

In[170]:=
```
Data = Table[Table[x, {x, 1, 20}] + x, {x, 0, 90, 20}]
```

Out[170]=
```
{{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20},
 {21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40},
 {41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60},
 {61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80},
 {81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100}}
```

In[171]:=
```
Grid[Data]
```

Out[171]=

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

Ques 2 . The Sum command has a syntax similar to that of Table.

In[57]:=
```
Sum[x^3, {x, 1, 10}]
```

Out[57]= 3025

In[56]:=
```
Table[x^3, {x, 1, 10}]
```

Out[56]= {1, 8, 27, 64, 125, 216, 343, 512, 729, 1000}

a. Use the Sum command to evaluate the following expression:

1^3 + 2^3 + 3^3 + 4^3 + 5^3 + 6^3 + 7^3 + 8^3 + 9^3 + 10^3 +11^3 + 12^3 + 13^3 + 14^3 + 15^3 +16^3 + 17^3 + 18^3 + 19^3 + 20^3

In[103]:=
```
Sum[y^x, {y, 1, 20}]
```

Out[103]= $1 + 2^x + 3^x + 4^x + 5^x + 6^x + 7^x + 8^x + 9^x + 10^x + 11^x + 12^x + 13^x + 14^x + 15^x + 16^x + 17^x + 18^x + 19^x + 20^x$

b. Make a table of values for x =1, 2, ... ,10 for the function.

In[100]:=
```
f[x_] := Sum[y^x, {y, 1, 20}]
```

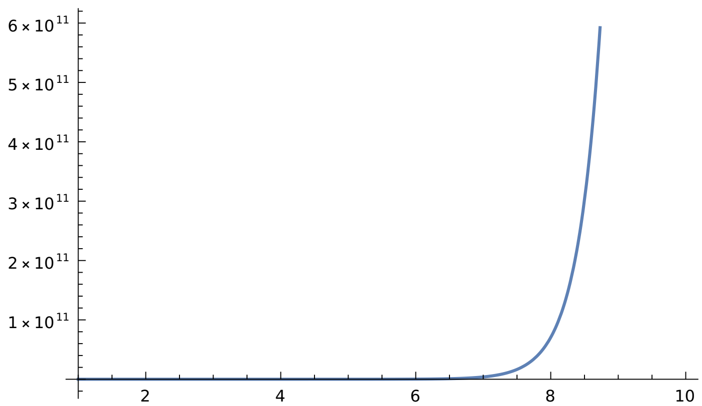
In[179]:=
```
Table[f[x], {x, 1, 10}]
```

Out[179]= {210, 2870, 44 100, 722 666, 12 333 300, 216 455 810, 3 877 286 700, 70 540 730 666, 1 299 155 279 940, 24 163 571 680 850}

c. Plot f (x) on the domain 1 ≤ x ≤ 10.

In[180]:=
```
Plot[f[x], {x, 1, 10}]
```

Out[180]=



Exercise 3.6

Ques 1. Make a plot of the piecewise function , and comment on its shape..

In[70]:=
```
f[x_] := Piecewise[{{0, x < 0}, {x ^ 2 / 2, 0 ≤ x < 1},
   {-x ^ 2 + 3 x - (3 / 2), 1 ≤ x < 2}, {(1 / 2) (3 - x) ^ 2, 2 ≤ x < 3}, {0, 3 ≤ x}}]
```

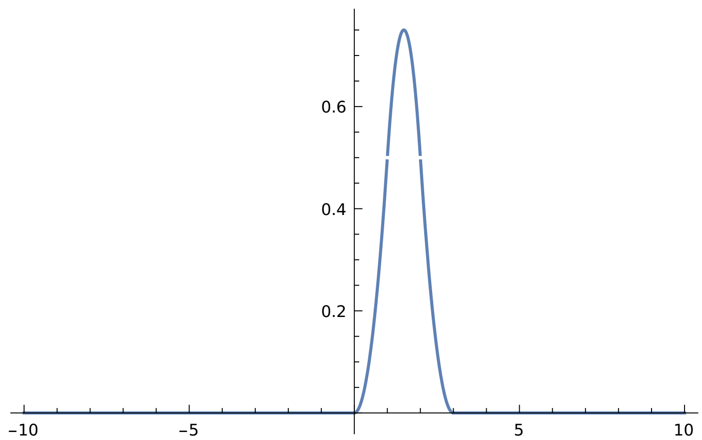
In[71]:=
```
f[x]
```

Out[71]=

$$\begin{cases} 0 & x < 0 \\ \frac{x^2}{2} & 0 \leq x < 1 \\ -\frac{3}{2} + 3x - x^2 & 1 \leq x < 2 \\ \frac{1}{2}(3-x)^2 & 2 \leq x < 3 \\ 0 & \text{True} \end{cases}$$

In[72]:=
```
Plot[f[x], {x, -10, 10}]
```

Out[72]=



The Piecewise Function is not continuous(breaks) at x=1 and x=2.

Ques 2. A step function assumes a constant value between consecutive integers n and n + 1. Make a plot of the step function f (x) whose value is n^2 when n ≤ x < n + 1. Use the domain 0 ≤ x ≤ 20.

In[50]:=

```
g[x_] := Table[Piecewise[{{n^2, n ≤ x < n + 1}}], {n, 0, 19}]
Plot[g[x], {x, 0, 20}]
```

Out[51]=