Name : Sarmistha Sutar
Roll No. : MAT/19/69

CHAPTER -3

EXERCISE 3.2

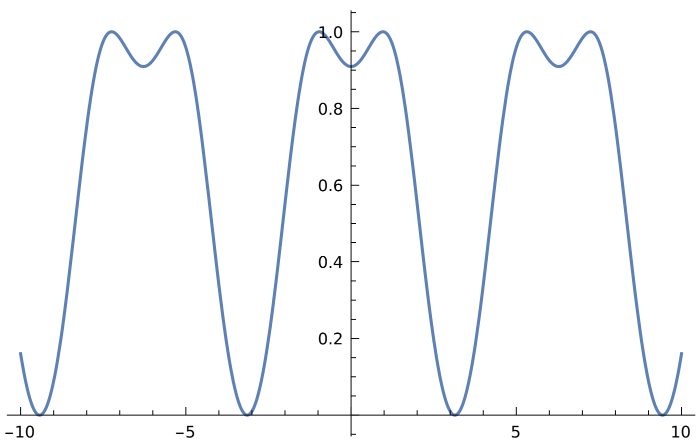
QUES 1. Plot the following functions on the domain $-10 \leq x \leq 10$.

(a) sin (1 + cos (x))

In[23]:=

```
Plot[Sin[1 + Cos[x]], {x, -10, 10}]
```

Out[23]=

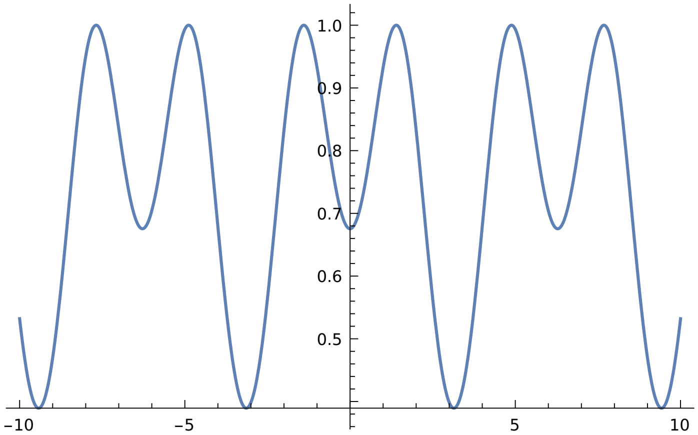


(b) sin (1.4 + cos (x))

In[24]:=

```
Plot[Sin[1.4 + Cos[x]], {x, -10, 10}]
```

Out[24]=

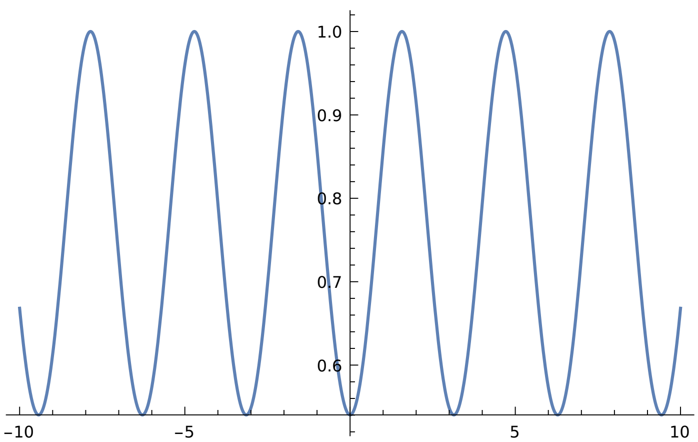


(c) sin (π / 2 + cos (x))

In[25]:=

```
Plot[Sin[ π / 2 + Cos[x]], {x, -10, 10}]
```

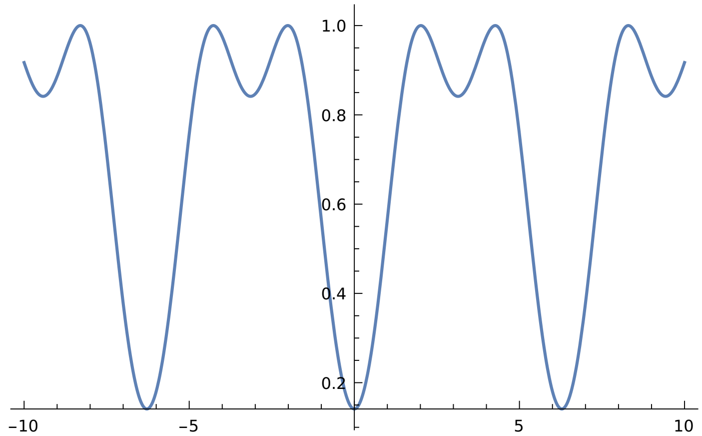
Out[25]=



(d) sin (2 + cos (x))

In[9]:= `Plot[Sin[2 + Cos[x]], {x, -10, 10}]`

Out[9]=

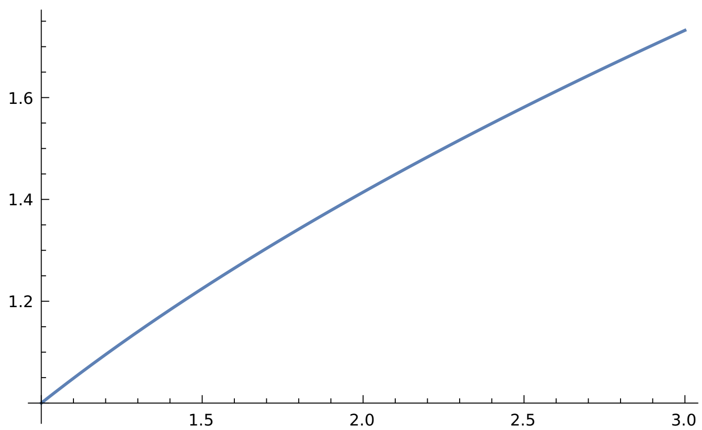


QUES2.

(a)

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

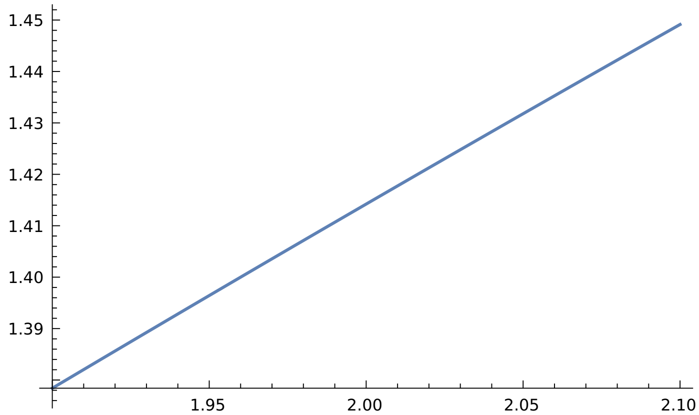
Out[10]=



(b)

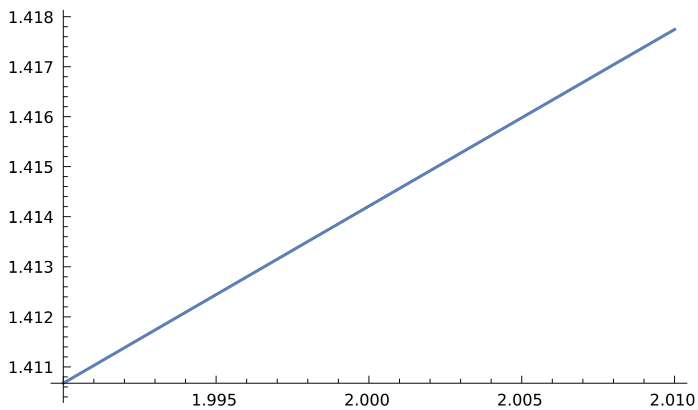
In[11]:= `With[{δ = 10^(-1)}, Plot[√x, {x, 2 - δ, 2 + δ}]]`

Out[11]=



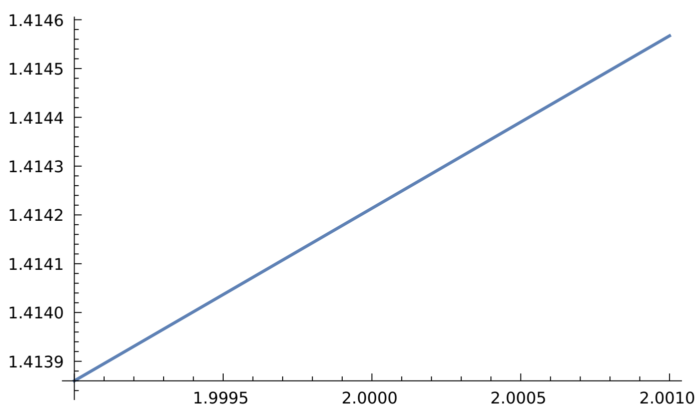
In[12]:= `With[{δ = 10^(-2)}, Plot[√x, {x, 2 - δ, 2 + δ}]]`

Out[12]=



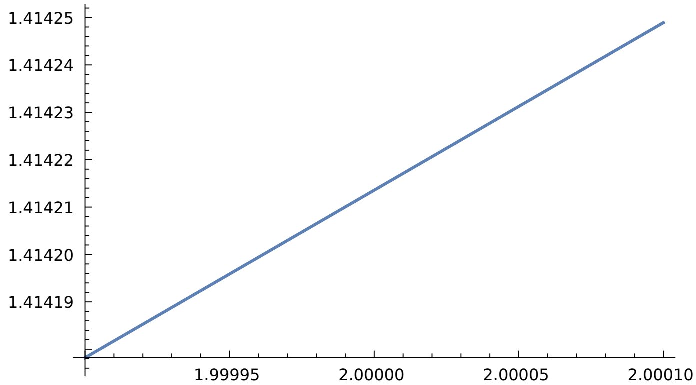
In[13]:= `With[{δ = 10^(-3)}, Plot[√x, {x, 2 - δ, 2 + δ}]]`

Out[13]=



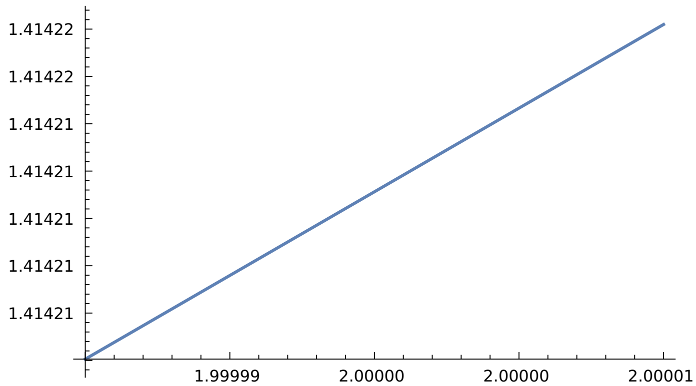
In[14]:= `With[{δ = 10^(-4)}, Plot[√x, {x, 2 - δ, 2 + δ}]]`

Out[14]=



In[15]:= `With[{δ = 10^(-5)}, Plot[√x, {x, 2 - δ, 2 + δ}]]`

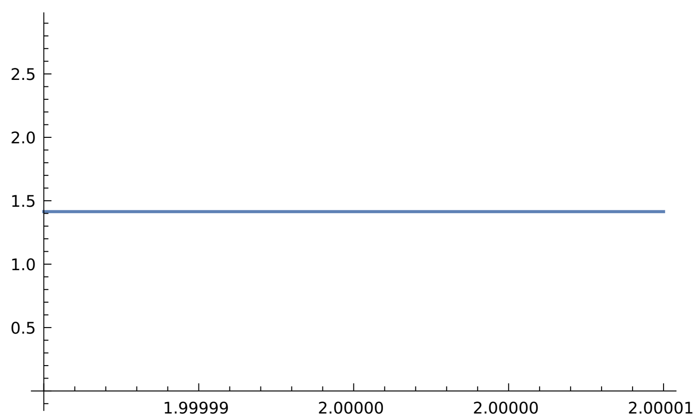
Out[15]=



(c)

In[6]:= `With[{δ = 10^(-5)}, Plot[√2, {x, 2 - δ, 2 + δ}]]`

Out[6]=



In[16]:= `f[x_] := N[Sqrt[x], 6]`

In[17]:= `f[2]`

Out[17]= 1.41421

(d)

In[26]:=
```
With[{δ = 10^(-20)}, Plot[√x, {x, 2 - δ, 2 + δ}]]
```

Plot : Endpoints for x in $\left\{x, \frac{199999999999999999999}{100000000000000000000}, \frac{200000000000000000001}{100000000000000000000}\right\}$ must have distinct machine -precision numerical values .

Plot : Endpoints for x in $\left\{x, \frac{199999999999999999999}{100000000000000000000}, \frac{200000000000000000001}{100000000000000000000}\right\}$ must have distinct machine -precision numerical values .

Plot : Endpoints for x in $\left\{x, \frac{199999999999999999999}{100000000000000000000}, \frac{200000000000000000001}{100000000000000000000}\right\}$ must have distinct machine -precision numerical values .

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

Plot : Endpoints for x in $\left\{x, \frac{199999999999999999999}{100000000000000000000}, \frac{200000000000000000001}{100000000000000000000}\right\}$ must have distinct machine -precision numerical values .

Plot : Endpoints for x in $\left\{x, \frac{199999999999999999999}{100000000000000000000}, \frac{200000000000000000001}{100000000000000000000}\right\}$ must have distinct machine -precision numerical values .

Plot : Endpoints for x in $\left\{x, \frac{199999999999999999999}{100000000000000000000}, \frac{200000000000000000001}{100000000000000000000}\right\}$ must have distinct machine -precision numerical values .

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

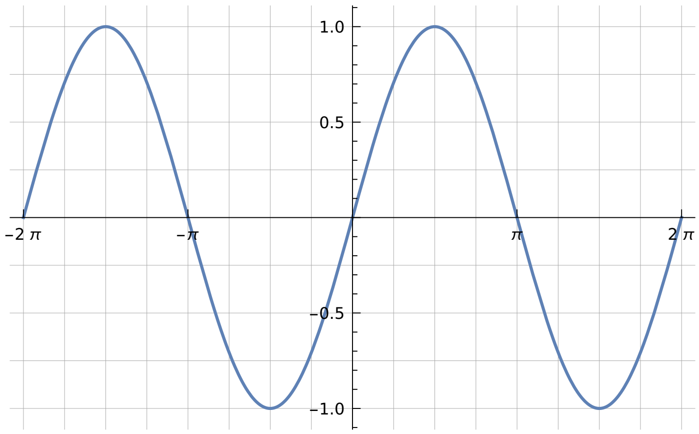
Out[26]= $\text{Plot}\left[\sqrt{x}, \left\{x, 2 - \frac{1}{100\,000\,000\,000\,000\,000\,000}, 2 + \frac{1}{100\,000\,000\,000\,000\,000\,000}\right\}\right]$

The two values ie. (2 + δ) and (2 - δ) and hence their difference is so small that it cannot be read by the computer thus the mathematica is showing error.

EXERCISE 3.3

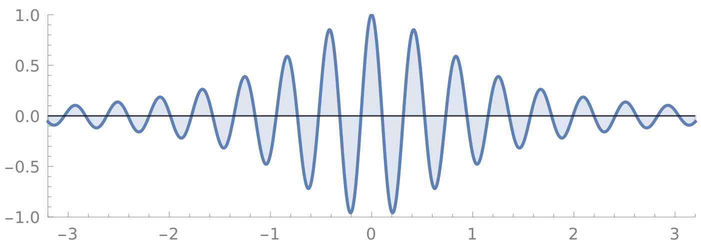
QUES1. Use the GridLines and Ticks options,
as well as the setting GridLinesStyle → Lighter[Gray],
to produce the following Plot of the sine function :

In[22]:=
```
Plot[Sin[x], {x, -2 Pi, 2 Pi}, GridLinesStyle → Lighter[Gray],
 GridLines → {Range[-2 Pi, 2 Pi, Pi / 4], Range[-1, 1, 0.25]},
 Ticks → {Range[-2 Pi, 2 Pi, Pi], Automatic}]
```

Out[22]=


QUES2.

In[7]:=
```
Plot[Cos[15 * x] / (1 + x ^ 2), {x, -3.2, 3.2}, AxesOrigin → {-3, 0},
 Axes → {x, y}, Frame → {{True, False}, {True, False}}, FrameStyle → Gray,
 PlotRange → {{-3.2, 3.2}, {-1, 1}}, AspectRatio → Automatic, Filling → Axis]
```

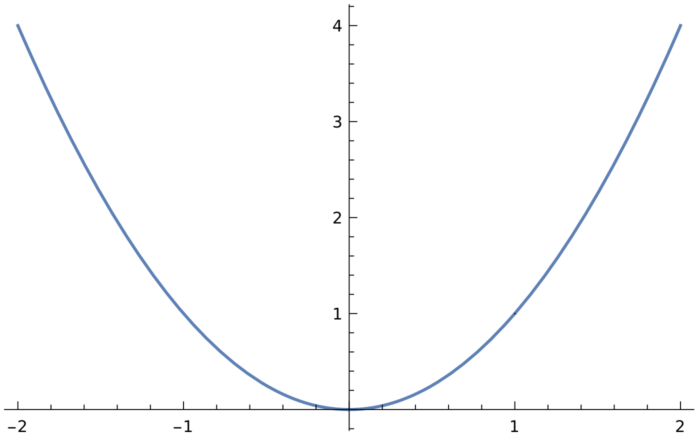
Out[7]=


QUES4.

In[2]:=
```
g[x_] := x ^ 2
```

In[3]:=
```
Plot[g[x], {x, -2, 2}, Exclusions → {x == 1}, ExclusionsStyle → Dashed]
```

Out[3]=



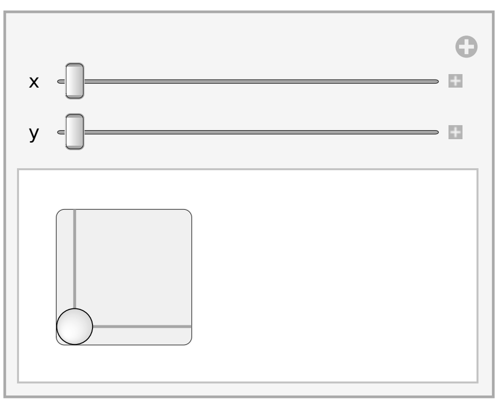
→ x = 1 is excluded which enables the vertical asymptote at x = 1.

EXERCISE 3.4

QUES1. 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[2]:=
```
Manipulate[Slider2D[{x, y}], {x, 0, 1}, {y, 0, 1}]
```

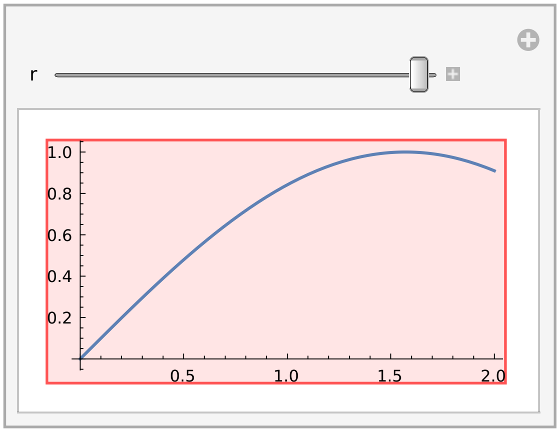
Out[2]=



QUES2. 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[4]:=
```
Manipulate [Plot[Sin[x], {x, 0, r},
  AspectRatio → k {k, 1 / 5, 5}, ImageSize → {Automatic , 128}], {r, 1, 2}]
```

Out[4]=



EXERCISE 3.5

QUES1 . 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[1]:=
```
Range[100]
```

Out[1]=
```
{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[2]:=
```
Partition [Range[100], 10]
```

Out[2]=
```
{{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}}
```

(b). Format a table of the first 100 integers,
with twenty digits per row. The first two rows, for example, should look like this:

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 × 38 × 40

In[3]:=
```
data = Partition[Range[100], 20]
```

Out[3]=
```
{{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[4]:=
```
Grid[data]
```

Out[4]=

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[8]:=
```
Grid[Table[Range[x, x + 19], {x, {1, 21, 41, 61, 81}}]]
```

Out[8]=

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[9]:=
```
f[x_] := x
```

In[10]:=
```
Grid[Table[Table[f[x], {x, x, x + 19}], {x, {1, 21, 41, 61, 81}}]]
```

Out[10]=

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

Ques2. The Style command is used to apply a particular style to an expression.

(a) Enter the following inputs and discuss the outputs.

In[20]:=
```
Style[4, Red]
```
Out[20]= 4

In[21]:=
```
Style[4, 72]
```
Out[21]= 4

In[22]:=
```
Style[4, "Section"]
```
Out[22]= 4

In[23]:=
```
Style[4, FontFamily → "Helvetica", FontWeight → "Bold"]
```
Out[23]= **4**

(b) One can apply a particular style to every item in a Grid by using the entire Grid as the first argument to Style. Create an output that matches that below. The font is Comic Sans MS, and the text should be blue.

1	1	× 1	× 1
2	× 4	× 8	× 16
3	× 9	× 27	× 81
4	× 16	× 64	× 256
5	× 25	× 125	× 625

In[24]:=
```
data = Table[{x, x^2, x^3, x^4}, {x, 1, 5}]
```
Out[24]= {{1, 1, 1, 1}, {2, 4, 8, 16}, {3, 9, 27, 81}, {4, 16, 64, 256}, {5, 25, 125, 625}}

In[25]:=
```
Style[Grid[data], FontFamily → "Conic Sans HS", Blue]
```
Out[25]=

1	1	1	1
2	4	8	16
3	9	27	81
4	16	64	256
5	25	125	625

(c) Alternately, one can apply style elements to an entire grid by selecting the cell bracket of the cell containing the grid, and visiting the Format menu. For instance, Format > Text Color > Blue will make all the text blue. Reproduce the Grid above, this time using the menu items to change the style.

In[26]:=
```
Style[Grid[data], Red, 20]
```

Out[26]=

1	1	1	1
2	4	8	16
3	9	27	81
4	16	64	256
5	25	125	625

In[27]:=
```
Clear[data];
```

Ques3 . A statement that is either true or false is called a predicate ; in Mathematica a predicate is any expression that evaluates to True or False . In this exercise you will learn how to use predicates to apply Styles selectively .

(a) A statement that is either true or false is called a predicate ; in Mathematica a predicate is any expression that evaluates to True or False . In this exercise you will learn how to use predicates to apply Styles selectively .

In[28]:=
```
? PrimeQ
```

Out[28]=

Symbol

PrimeQ [n] yields True if n is a prime number , and yields False otherwise .

Options	GaussianIntegers → False
Attributes	{Listable , Protected }
Full Name	System`PrimeQ

In[30]:=
```
? *Q
```

System`

AcyclicGraphQ	CoprimeQ	GroupElementQ	MandelbrotSet⋱ MemberQ	PermutationCy⋱ clesQ	StringEndsQ
AlgebraicInteg⋱ erQ	DataStructureQ	HamiltonianGr⋱ aphQ	MarcumQ	PermutationLis⋱ tQ	StringFormatQ
AlgebraicUnitQ	DateObjectQ	HermitianMatri⋱ xQ	MatchLocalNa⋱ meQ	PlanarGraphQ	StringFreeQ
Antihermitian⋱ MatrixQ	DateOverlapsQ	Hypergeometri⋱ cPFQ	MatchQ	PointProcessP⋱ arameterQ	StringMatchQ
Antisymmetric⋱ MatrixQ	DateWithinQ	ImageContainsQ	MatrixQ	PolynomialExp⋱ ressionQ	StringQ
ArgumentCoun⋱ tQ	DaylightQ	ImageInstanceQ	MemberQ	PolynomialQ	StringStartsQ

Out[30]=

ArrayQ	DayMatchQ	ImageQ	MersennePrim⋱ eExponentQ	PositiveDefinit⋱ eMatrixQ	StructuredArra⋱ yHeadQ
AskedQ	DeviceOpenQ	IndefiniteMatri⋱ xQ	MeshRegionQ	PositiveSemide⋱ finiteMatrixQ	SubsetQ
AssociationQ	Diagonalizable⋱ MatrixQ	IndependentEd⋱ geSetQ	MissingQ	PossibleZeroQ	SymmetricMatr⋱ ixQ
AtomQ	DiagonalMatrixQ	IndependentVe⋱ rtexSetQ	MixedGraphQ	PrimePowerQ	SyntaxQ
AudioInstanceQ	DictionaryWor⋱ dQ	InexactNumberQ	MoleculeContai⋱ nsQ	PrimeQ	TautologyQ
AudioQ	DigitQ	IntegerQ	MoleculeEquiv⋱ alentQ	PrimitivePolyn⋱ omialQ	TensorQ
BinaryImageQ	DirectedGraphQ	IntersectingQ	MoleculeQ	PrintableASCIIQ	TimeObjectQ
BioSequenceQ	DirectoryQ	IntervalMembe⋱ rQ	MultigraphQ	ProcessParame⋱ terQ	TreeGraphQ
BipartiteGraphQ	DiscreteTimeM⋱ odelQ	InverseElliptic⋱ NomeQ	NameQ	QHypergeomet⋱ ricPFQ	TrueQ
BondQ	DisjointQ	IrreduciblePoly⋱ nomialQ	NegativeDefini⋱ teMatrixQ	QuadraticIrrati⋱ onalQ	UnateQ
BooleanQ	DispatchQ	IsomorphicGra⋱ phQ	NegativeSemid⋱ efiniteMatrix⋱ Q	QuantityQ	UndirectedGra⋱ phQ
BoundaryMesh⋱ RegionQ	DistributionPar⋱ ameterQ	KEdgeConnect⋱ edGraphQ	NormalMatrixQ	RationalExpres⋱ sionQ	UnitaryMatrixQ
BoundedRegio⋱ nQ	DuplicateFreeQ	KeyExistsQ	NumberQ	RegionQ	UnsameQ
BusinessDayQ	EdgeCoverQ	KeyFreeQ	NumericArrayQ	RegularlySamp⋱ ledQ	UpperCaseQ
ByteArrayForm⋱ atQ	EdgeQ	KeyMemberQ	NumericQ	RootOfUnityQ	UpperTriangul⋱ arMatrixQ
ByteArrayQ	EdgeTaggedGr⋱ aphQ	KnownUnitQ	ObservableMo⋱ delQ	SameQ	ValueQ
ColorQ	EdgeWeighted⋱ GraphQ	KVertexConnec⋱ tedGraphQ	OddQ	SatisfiableQ	VectorQ
CompatibleUni⋱ tQ	EllipticNomeQ	LeapYearQ	OptionQ	ScheduledTask⋱ ActiveQ	VertexCoverQ
CompleteGrap⋱ hQ	EmptyGraphQ	LegendreQ	OrderedQ	SimpleGraphQ	VertexQ
CompositeQ	EulerianGraphQ	LetterQ	OrthogonalMat⋱ rixQ	SimplePolygonQ	VertexWeighte⋱ dGraphQ
ConnectedGra⋱ phQ	EvenQ	LinkConnectedQ	OutputControll⋱ ableModelQ	SimplePolyhed⋱ ronQ	VideoQ
ConnectedMol⋱ eculeQ	ExactNumberQ	LinkReadyQ	PacletNewerQ	SocketReadyQ	WeaklyConnect⋱ edGraphQ

ConstantRegionQ	FailureQ	ListQ	PacletObjectQ	SolidRegionQ	WeightedGraphQ
ContinuousTimeModelQ	FileExistsQ	LoopFreeGraphQ	PalindromeQ	SpatialObservationRegionQ	
ControllableModelQ	FileFormatQ	LowerCaseQ	PartitionsQ	SpeakerMatchQ	
ConvexPolygonQ	FreeQ	LowerTriangularMatrixQ	PathGraphQ	SquareFreeQ	
ConvexPolyhedronQ	GeoWithinQ	MachineNumberQ	PerfectNumberQ	SquareMatrixQ	
ConvexRegionQ	GraphQ	ManagedLibraryExpressionQ	PermissionsGroupMemberQ	StringContainsQ	

CloudObject`

CloudDeployActiveQ UUIDQ

CloudSystem`

FeatureEnabledQ $CloudContinuousActionQ

JLink`

FrontEndSharedQ JavaObjectQ KernelSharedQ SameObjectQ

MSP`

MSPValueQ

Security`

InsecureExprQ

(b) The If command is used to generate one output if a specified condition (i.e., a predicate) is true, and another if that condition is false. The predicate is the first argument to If. The next argument is what is returned if the predicate is true (If is discussed in Section 8.5). A third argument specifies the expression to be returned if the predicate is false. Enter the following input and discuss the output.

In[31]:=
```
Table[If[PrimeQ[n], Style[n, Red], n], {n, 100}]
```

Out[31]=
```
{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) Format a table of the first 100 integers ,
with ten digits per row . In this table , make all prime numbers red .

In[32]:=
```
Partition[Table[If[PrimeQ[n], Style[n, Red], n], {n, 100}], 10]
```

Out[32]=
```
{{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) Format a table of the first 100 integers , with ten digits per row . In this table ,
make all squarefree numbers blue and underlined . Note :
An integer is squarefree if none of its divisors (other than 1) are perfect squares .

In[35]:=
```
Partition[Table[If[SquareFreeQ[n], Style[n, Blue, Underlined], n], {n, 100}], 10]
```

Out[35]=
```
{{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}}
```

(e) Format a table of the first 100 integers ,
with ten digits per row . In this table , make all prime powers orange
and italicized . Note : An integer is a prime power if it is equal to pn ,
where p is prime and n is a positive integer .

In[36]:=
```
Partition[Table[If[PrimePowerQ[n], Style[n, Orange], n], {n, 100}], 10]
```

Out[36]=
```
{{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}}
```

Ques4 . The Sum command has a syntax similar to that of Table .

(a)

In[37]:=
```
Sum[x ^ 3, {x, 1, 20}]
```
Out[37]= 44 100

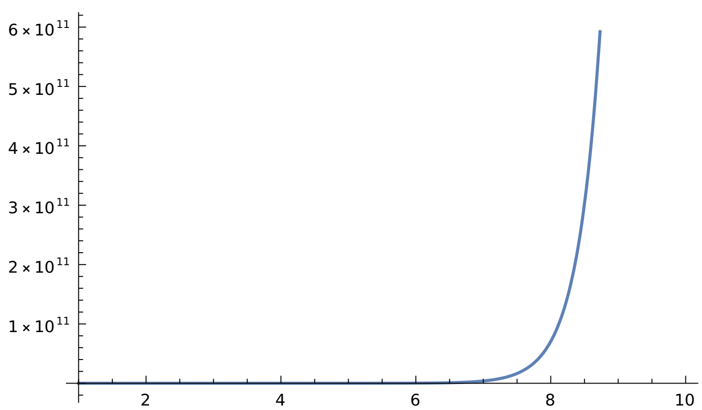
(b)

In[38]:=
```
f[x_] = 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
```
Out[38]= $1 + 2^x + 3^x + 4^x + 5^x + 6^x + 7^x + 8^x + 9^x + 10^x + 11^x + 12^x + 13^x + 14^x + 15^x + 16^x + 17^x + 18^x + 19^x + 20^x$

In[39]:=
```
Table[f[x], {x, 1, 10}]
```
Out[39]= {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)

In[40]:=
```
Plot[f[x], {x, 1, 10}]
```
,

Out[40]=


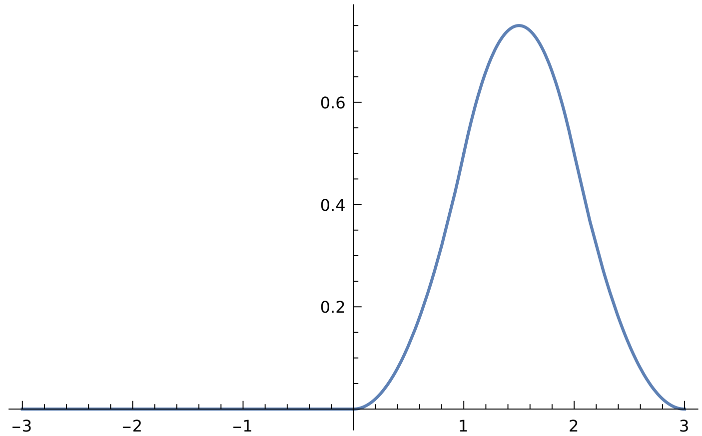
EXERCISE 3.6

QUES2 .

In[2]:=
```
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}}, 1]
```

In[3]:= `Plot[f[x], {x, -3, 3}]`

Out[3]=



QUES3 . 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[1]:=

```
h[x_] := Piecewise[{{0, 0 ≤ x < 1}, {1, 1 ≤ x < 2}, {4, 2 ≤ x < 3},
   {9, 3 ≤ x < 4}, {16, 4 ≤ x < 5}, {25, 5 ≤ x < 6}, {36, 6 ≤ x < 7}, {49, 7 ≤ x < 8},
   {64, 8 ≤ x < 9}, {81, 9 ≤ x < 10}, {100, 10 ≤ x < 11}, {121, 11 ≤ x < 12},
   {144, 12 ≤ x < 13}, {169, 13 ≤ x < 14}, {196, 14 ≤ x < 15}, {225, 15 ≤ x < 16},
   {256, 16 ≤ x < 17}, {289, 17 ≤ x < 18}, {324, 18 ≤ x < 19}, {361, 19 ≤ x < 20}}]
```

In[5]:= `Plot[h[x], {x, 0, 20}]`

Out[5]=