Just Do It!

Problem 1:

a) Execute the following three commands:

```
x = 2
```

```
2
```

```
Factor[x^2 - 1]
```

```
3
```

$\text{Sum}\left[i^2 - \text{Sin}[i], \{i, 0, x\}\right]$

```
5 - Sin[1] - Sin[2]
```

and briefly explain the output that you get.

For an output, we get 5 - Sin[1] - Sin[2]. We assigned x as the number 2, so in input 12, 2 squared is 4. minus one is 3.

b) Execute the following three commands:

```
Clear[x]
```

```
Factor[x^2 - 1]
```

```
(-1 + x) (1 + x)
```

$\text{Sum}\left[i^2 - \text{Sin}[i], \{i, 0, x\}\right]$

$$\frac{1}{6} x (1 + x) (1 + 2 x) + \frac{1}{2} \left(-\text{Cos}\left[\frac{1}{2}\right] + \text{Cos}\left[\frac{1}{2} + x\right]\right) \text{Csc}\left[\frac{1}{2}\right]$$

The out puts of the last two commands are different that tbhey were in part a. THis is because in part a, x was given a value, here there is no value given to x.

and briefly explain why the outputs of the last two commands are different than they were in Part a)

Just Do It!

Problem 2: A local package delivery service has the following rate structure:

a) Any package weighing less than a pound costs $5.00.

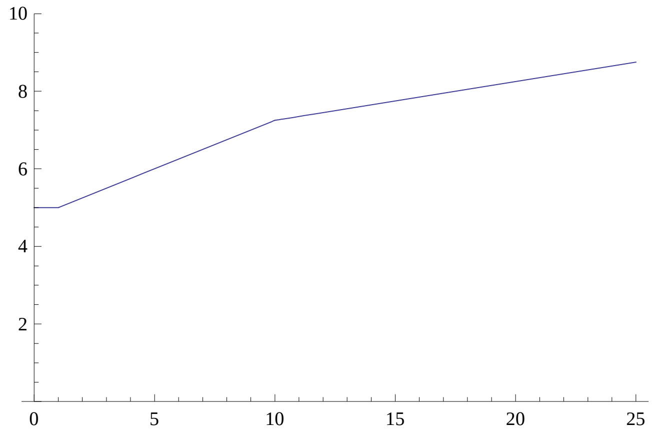
b) A package weighing at least one pound but less than 10 pounds costs 25 cents a pound for each additional pound or part of a pound over 1 pound.

c) A package weighing at least 10 pounds but less than 25 pounds costs 10 cents a pound for each additional pound or part of a pound over 10 pounds.

No packages weighing 25 pounds or more are accepted. Define function cost[x] that gives the cost of delivering a package weighing x pounds for $0 < x < 25$. Graph the function on the interval $0 < x < 25$.

```
cost[x_] := 5 /; x < 1
cost[x_] := 5 + .25 (x - 1) /; 10 > x ≥ 1
cost[x_] := 7.25 + .10 (x - 10) /; 25 ≥ x > 10
```

```
Plot[cost[x], {x, 0, 25}, PlotRange → {0, 10}]
```



Problem 3: (Defining Periodic Functions)

Recall that a function f[x] is periodic if there is a positive constant p such that

(*) f[x + p] = f[x] for all x.

The six trigonometric functions are periodic, and so is any constant function. If there is a smallest positive number p with the property (*), it is called the period of f.

The built-in Mathematica function Mod[m,n] can, among many other things, help us to define periodic functions different from the standard trigonometric functions.

For two real numbers m and n, Mod[m,n] gives the reainder for the division of the number m by the number n; that is, if

m = q n + r where r is between 0 and n but not equal to n

then Mod[m,n] = r. Run the following to check out a few values:

```
Mod[3.5, 2]

1.5

Mod[3.5, 2.5]

1.

Mod[1.5, 2]

1.5

Mod[-3.5, 2]

0.5

Mod[3.5, -2]

-0.5
```

Notice that if n is a positive number, then the remainder r is nonnegative and less than n; that is,

$n > 0 \implies 0 \leq r < n$.

Also, if f[x] is a periodic function of period p, then the value of f at any x is the same as its value at Mod[x,p]; that is,

f[x] = f[Mod[x,p]]

and 0 ≤ Mod[x,p] < p.

a) Enter the non-periodic function g[x] = x^2- 2 x.

b) Use the Mod[m,n] command to define a function f[x] of period 3 that is identical to g[x] on the interval 0 < x < 3. Graph the fuction f[x] on the interval

-6 ≤ x ≤ 6.

c) Did you need to use := to define the function f[x]? Why?

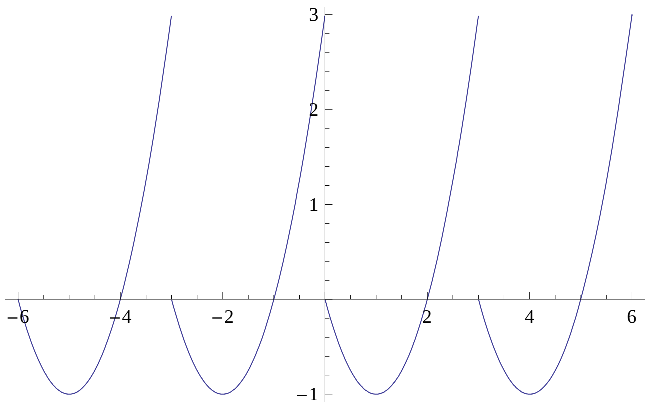
```
yes I needed to use := to define function f[x] because if I didn't use it,
it wouldn't understand the restriction that 0 ≤ Mod[x, 3] < 3
```

```
Clear[f, g]

g[x_] := x^2 - 2 x

f[x_] := g[Mod[x, 3]] /;
  0 ≤ Mod[x, 3] < 3

Plot[f[x], {x, -6, 6}]
```



Just Do It!

Problem 4: Newton's Law of Cooling states that, under suitable conditions, the temperature of a cooling body decreases at a rate proportional to the difference between the body's temperature and the (assumed constant) temperature of its surroundings. Suppose that a cake is taken from the oven at 350 degrees F. and placed in a large room that is maintained at 70 degrees to cool. Let t[n] represent the temperature of the cake n minutes after it is removed from the oven. Then application of Newton's Law of Cooling would lead to the following sort of mathematical model for t[n]:

t[0] = 350

t[n] - t[n-1] = -c (t[n-1] - 70)

where c is a positive constant that measure the rate at which this cake dissipates heat.

a) Define a function t[n] that represents the temperature of the cake n minutes after it is removed from the oven if the value of c for this particular cake is c = .1.

What is the temperature of the cake after 30 minutes?

b) Make a table of values of t[n] for n values between 0 and 60 minutes.

```
Clear[c, t, n]
c = .1
0.1
t[0] = 350
350
t[n_] := t[n] = -c (t[n - 1] - 70) + t[n - 1]
t[30]
81.8695
Table[t[n], {n, 0, 60}]
{350, 322., 296.8, 274.12, 253.708, 235.337, 218.803, 203.923,
 190.531, 178.478, 167.63, 157.867, 149.08, 141.172, 134.055, 127.65,
 121.885, 116.696, 112.026, 107.824, 104.041, 100.637, 97.5736,
 94.8162, 92.3346, 90.1011, 88.091, 86.2819, 84.6537, 83.1884, 81.8695,
 80.6826, 79.6143, 78.6529, 77.7876, 77.0088, 76.308, 75.6772, 75.1094,
 74.5985, 74.1386, 73.7248, 73.3523, 73.0171, 72.7154, 72.4438,
 72.1994, 71.9795, 71.7816, 71.6034, 71.4431, 71.2988, 71.1689,
 71.052, 70.9468, 70.8521, 70.7669, 70.6902, 70.6212, 70.5591, 70.5032}
```