In [ ]:
```
import numpy as np
```

In [ ]:
```
# check second-order
def L2_positive(L2, y):
    return (y.T @ L2 @ y).reshape(1,)[0] > 0
```

solution - 1

In [ ]:
```
L2 = np.array([  [-6, 0],
                 [0, 2]])
y = np.array([5/2, 1]).reshape(2,1)
L2_positive(L2, y)
```

Out[ ]:
```
False
```

solution - 2

In [ ]:
```
P = np.array([  [0, 0],
                [0, 2]])
y = np.array([-0.49, 1]).reshape(2,1)
L2_positive(L2, y)
```

Out[ ]:
```
True
```

In [ ]:
```
P = np.array([  [0, 0],
                [0, 2]])
y = np.array([-0.53, 1]).reshape(2,1)
L2_positive(L2, y)
```

Out[ ]:
```
True
```

find optimal value

In [ ]:
```
def f(x1, x2):
    return (x1 - 1)**2 + (x2 - 10)**2
```

In [ ]:
```
# point 1
x1 = -5.12
x2 = 7.5
print('Optimal value:',f(x1,x2))
```

```
Optimal value: 43.7044
```

In [ ]:
```
# point 2
x1 = 7.12
x2 = 7.5
print('Optimal value:',f(x1,x2))
```

```
Optimal value: 43.7044
```