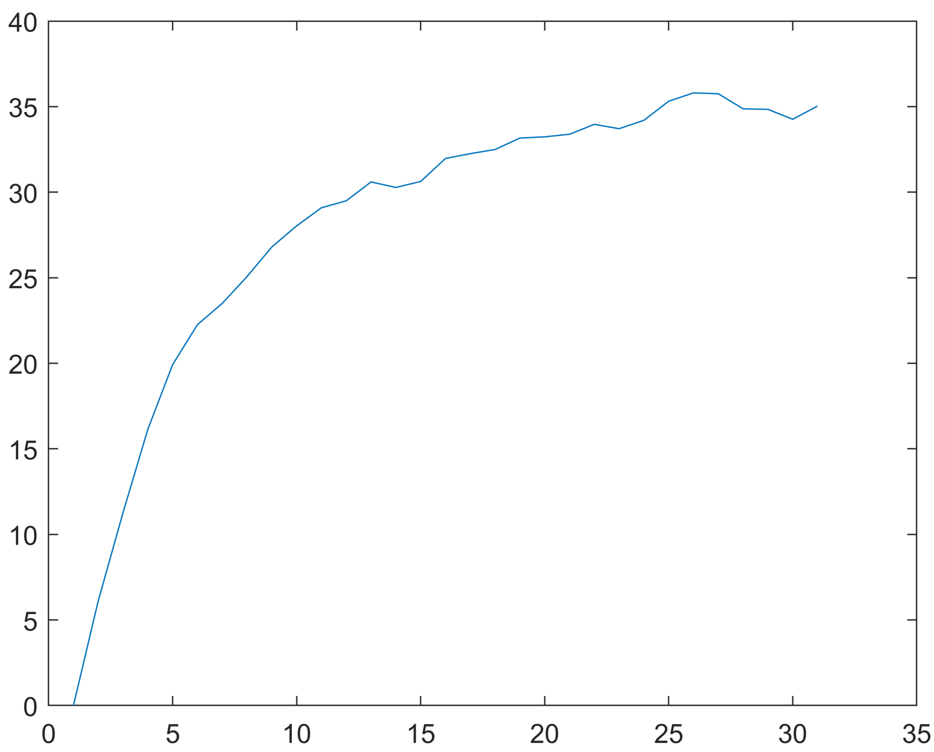
```
% Group 6, Michael Brian Damköhler, Michal Szabo
clear all
close all
clc
```

Exercise 1:

a)

```
N = 30;
k = 1;
y(1) = 0;
e = sqrt(0.4) * randn(1,N+1);
u = 10 * ones(1,N);
t = [0: N-1];

for k = 1:N
    y(k+1) = 0.8 * y(k) + 0.7 * u(k) + e(k+1) + 0.5 * e(k);
end

figure
plot([1:k+1], y);
```

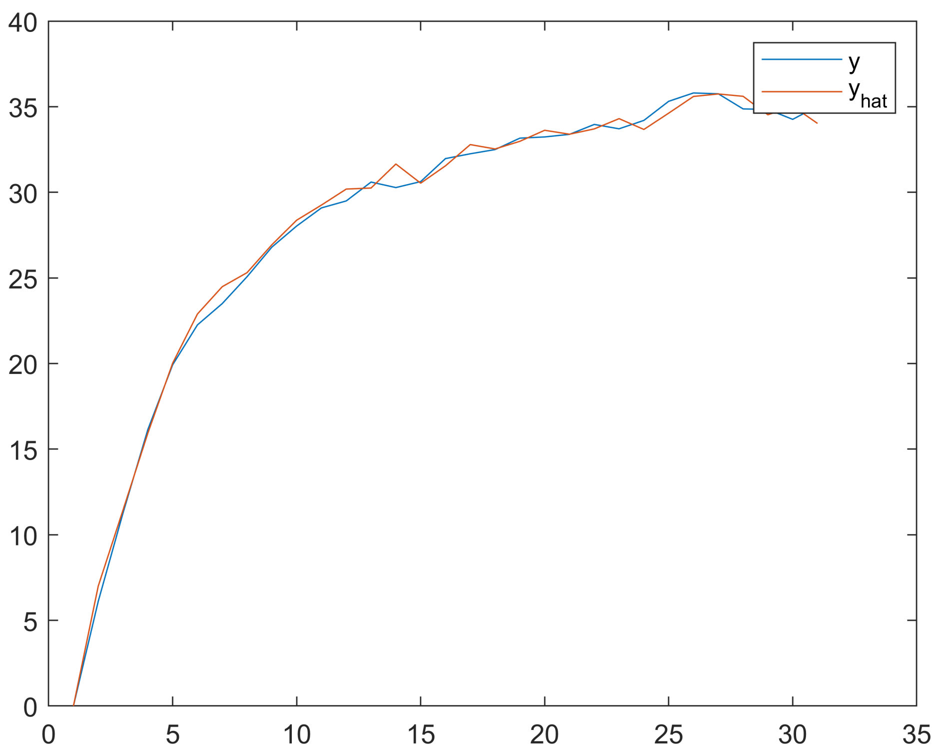



```
save('data_set.mat', 'y');
```

b)

y^ = 0.8 * y(t-1) + 0.7 u(t-1) -0.5 v^(t-1) + 0.5 * v(t-1) c)

```
for j = 1:N
    v(j) = y(j+1) - 0.8 * y(j) - 0.7 * u(j);
end

v_hat(1) = 0;

for l = 1:N
    v_hat(l+1) = -0.5 * v_hat(l) + 0.5 * v(l);
end

y_hat(1) = 0;

for k = 1:N
    y_hat(k+1) = 0.8 * y(k) + 0.7 * u(k) + v_hat(k);
end

figure
plot([1:k+1], y, [1:k+1], y_hat);
legend('y', 'y_{hat}');
```

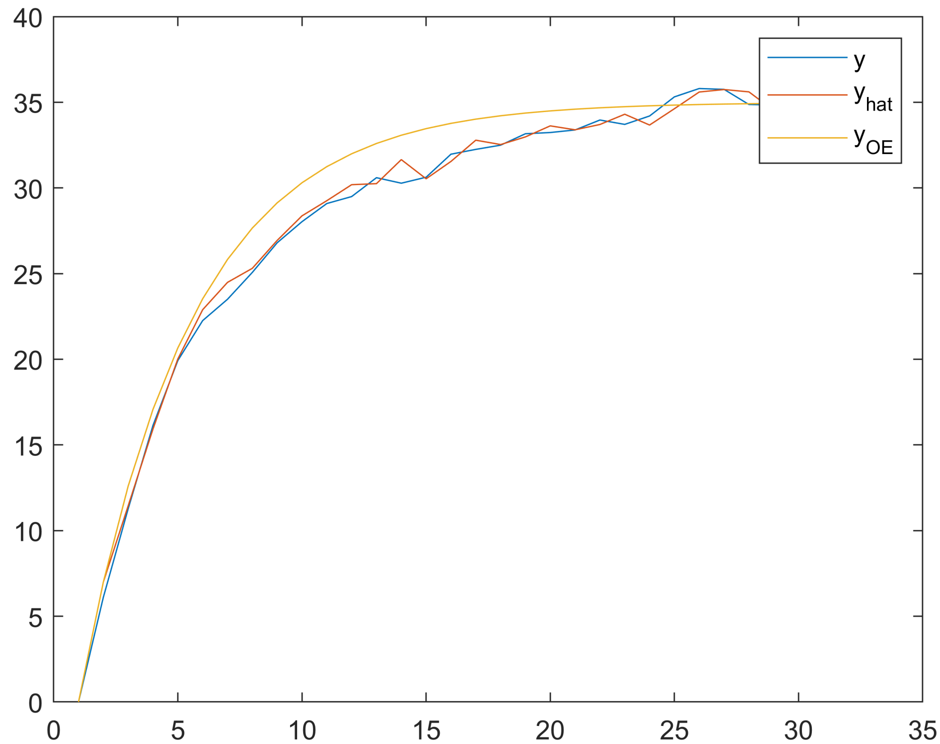


d)

```
V_n = sum((y - y_hat).^2) / N;
y_sum = sum(y_hat.^2);
prediction_error = sqrt(V_n / y_sum);
V_n_sqrt = sqrt(V_n);
```

e, f)

```
y_OE(1) = 0;

for g = 1:N
    y_OE(g+1) = 0.8 * y_OE(g) + 0.7 * u(g);
end

figure
plot([1:k+1], y, [1:k+1], y_hat, [1:k+1], y_OE);
legend('y', 'y_{hat}', 'y_{OE}');
```




```
V_n_OE = sum((y_OE - y_hat).^2) / N;
y_sum_OE = sum(y_hat.^2);
prediction_error_OE = sqrt(V_n_OE / y_sum_OE);
V_n_sqrt_OE = sqrt(V_n_OE);
```

Conclusio:

With this data point, the Armax prediction model should be prefered due to smaller prediction output error.