```
% Define function
function R = CPPI(F,M)

% Set up values
T = 2; % Investment horizon
sigma = 0.3; % Volatility
mu = 0.1; % Real world drift
P0 = 100; % Initial wealth
r = 0.05; % Risk-free rate
R0 = 1/250; % Rebalancing interval
B0 = 100; % Initial cash
alpha0 = 0; % Initial risky asset position
S0 = 100; % Initial risky asset price
Sim = 80000; % Number of simulations
N = T/R0;


% Set up initial vectors
% Stock price
S_old = zeros(Sim,1);
S_new = zeros(Sim,1);
S_old(1:Sim,1) = S0;
% Risk-free account (Bank account)
B_old = zeros(Sim,1);
B_new = zeros(Sim,1);
B_old(1:Sim,1) = B0;
% Numer of asset share
alpha_old = zeros(Sim,1);
alpha_new = zeros(Sim,1);
alpha_old(1:Sim,1) = alpha0;
% Portfolio value
P_old = zeros(Sim,1);
P_new = zeros(Sim,1);
P_old(1:Sim,1) = P0;
% Calculate portfolio value at time 0
%Pzero = P0*ones(Sim,1);
% Return
R = zeros(Sim,1);

% Timestep loop
for i = 1:N
    % Stock price
    S_new(:,1) = S_old(:,1).*exp((mu-sigma^2/2)*R0+sigma*randn(Sim,1)*sqrt(R0));
    S_new(:,1) = max(S_new(:,1),0);
    % Number of shares
    alpha_new(:,1) = M*(max(0,B_old(:,1).*exp(r*R0)+alpha_old(:,1).*S_new(:,1)-F)).↙
/S_new(:,1);
    % Risk-free account (Bank account)
    B_new(:,1) = B_old(:,1).*exp(r*R0)-(alpha_new(:,1)-alpha_old(:,1)).*S_new(:,1);
    % Portfolio value
    P_new(:,1) = B_new(:,1)+alpha_new(:,1).*S_new(:,1);
    % Update
    S_old(:,1) = S_new(:,1);
    alpha_old(:,1) = alpha_new(:,1);
    B_old(:,1) = B_new(:,1);
end
R(:,1) = log(P_new(:,1)./P0);
end
```