Appendix: Matlab code

```
%% Initialize Markowitz portfolio

clear;

n=8;
Corr=zeros(n,n);
for i=1:n
    for j=1:n
        Corr(i,j)=(-1)^abs(i-j)/(abs(i-j)+1);
    end
end
sigma=zeros(n,1);
mu=zeros(n,1);
sigma(1)=2;
mu(1)=3;
for i=1:n-1
    sigma(i+1)=sigma(i)+2*rand;
    mu(i+1)=mu(i)+1;
end
D=diag(sigma);
C2=D*Corr*D;
C=0.5*(C2+C2');

%% Exercise 1
%lb and ub are vectors of zero and 1, the boundaries of x as specified in the problem
lb = zeros(j,1);
ub = ones(j,1);
r = 3:0.25:9;
len = length(r);
result1_vec_sigma = zeros(1,len);
result1_vec_mu = zeros(1,len);
e = ones(1,j);
mu_plus_e = [mu'; e];
count = 1;

for count = 1:len

    [x,fval,exitflag,output] = quadprog(C,[],[],[],mu_plus_e,[r(count); 1],lb,ub);
    %we have no extra vector f and no inequalities, the equality constraint
    %is constructed by fusing the mu and e vectors and the results
    sigma_result = sqrt(x' * C * x);
    mu_result = (mu' * x);
    result1_vec_sigma(1,count) = sigma_result;
    result1_vec_mu(1,count) = mu_result;
    count = count+1;

end
```

```
plot(result1_vec_sigma, result1_vec_mu)
title('Mu(x) and sigma(x) plotted for all values of r')
xlabel('sigma(x)')
ylabel('mu(x)')

%% Exercise 2

lb = zeros(j,1);
ub = ones(j,1);
r = 3:0.25:9;
len = length(r);
result2_vec_sigma = zeros(1,len);
result2_vec_mu = zeros(1,len);
e = ones(1,j);
count = 1;

for count = 1:len

    [x,fval,exitflag,output] = quadprog(C,[],e,1,mu',r(count),lb,ub);
   %we keep mu as Aeq and each r as beq, and use e and 1 as inequality
   %constraints

    sigma_result = sqrt(x' * C * x);
    mu_result = (mu' * x);
    result2_vec_sigma(1,count) = sigma_result;
    result2_vec_mu(1,count) = mu_result;
    count = count+1;

end

plot(result2_vec_sigma, result2_vec_mu)
title('Mu(x) and sigma(x) plotted for all values of r')
xlabel('sigma(x)')
ylabel('mu(x)')

%% Exercise 3
lb = zeros(j,1);
ub = ones(j,1);
r = 3:0.25:9;
len = length(r);
result3_vec_sigma = zeros(1,len);
result3_vec_mu = zeros(1,len);
e = ones(1,j);
count = 1;

for count = 1:len

    [x,fval,exitflag,output] = quadprog(C,[],-mu',-r(count),e,1,lb,ub);
```

```
    % e and 1 is our equality constraints, while mu and each r is our
    % inequality constraints
     sigma_result = sqrt(x' * C * x);
     mu_result = (mu' * x);
     result3_vec_sigma(1,count) = sigma_result;
     result3_vec_mu(1,count) = mu_result;
     count = count+1;

end

plot(result3_vec_sigma, result3_vec_mu)
title('Mu(x) and sigma(x) plotted for all values of r')
xlabel('sigma(x)')
ylabel('mu(x)')

%mu_result equals r per definition

%% Exercise 4
%lb is set to empty, since removing the lower bound removes the constraint on x being large
lb = [];
ub = ones(j,1);
r = 3:0.25:9;
len = length(r);
result4_vec_sigma = zeros(1,len);
result4_vec_mu = zeros(1,len);
e = ones(1,j);
mu_plus_e = [mu'; e];
count = 1;

for count = 1:len

    [x,fval,exitflag,output] = quadprog(C,[],[],[],mu_plus_e,[r(count); 1],lb,ub);
    sigma_result = sqrt(x' * C * x);
    mu_result = (mu' * x);
    result4_vec_sigma(1,count) = sigma_result;
    result4_vec_mu(1,count) = mu_result;
    count = count+1;

end

plot(result4_vec_sigma, result4_vec_mu)
title('Mu(x) and sigma(x) plotted for all values of r')
xlabel('sigma(x)')
ylabel('mu(x)')

%% Exercise 5 initialize

%1 = GE, 2 = SSAAF, 3 = AMKBY, 4 = TTM, 5 = PFE, 6 = GS
j_5 = 6;
```

```
i_5 = 6;
%take the annualized data from the downloaded material in percentage
%points

mu_5 = [-5.51; -12.31; 3.87; -10.79; 10.43; 10.37]; %annualized return
sigma_5 = [27.25; 35.81; 29.01; 42.76; 15.73; 23.89]; %annualized standard deviation
D_5 = diag(sigma_5);

%take the correlations from the downloaded materials
Corr_5 = [1 -0.58 0.52 0.41 -0.50 0.73;
         -0.58 1 -0.61 -0.14 0.35 -0.64;
         0.52 -0.61 1 0.60 -0.09 0.92;
         0.41 -0.14 0.60 1 -0.45 0.76;
         -0.50 0.35 -0.09 -0.45 1 -0.40;
         0.73 -0.64 0.92 0.76 -0.40 1];

C2_5 = D_5*Corr_5*D_5;
C_5 = 0.5*(C2_5+C2_5');

%% Exercise 5 run and plot

%lb and ub are vectors of zero and 1, the boundaries of x as specified in the problem
lb_5 = zeros(j_5,1);
ub_5 = ones(j_5,1);
r = 3:0.25:9;
len = length(r);
result5_vec_sigma = zeros(1,len);
result5_vec_mu = zeros(1,len);
e = ones(1,j_5);
mu_plus_e_5 = [mu_5'; e];
count = 1;

for count = 1:len

    [x,fval,exitflag,output] = quadprog(C_5,[],[],[],mu_plus_e_5,[r(count); 1],lb_5,ub_5);
    %we have no extra vector f and no inequalities, the equality constraint
    %is constructed by fusing the mu and e vectors and the results
    sigma_result = sqrt(x' * C_5 * x);
    mu_result = (mu_5' * x);
    result5_vec_sigma(1,count) = sigma_result;
    result5_vec_mu(1,count) = mu_result;
    count = count+1;

end

plot(result5_vec_sigma, result5_vec_mu)
title('Mu(x) and sigma(x) plotted for all values of r')
xlabel('sigma(x)')
ylabel('mu(x)')
```

```
%% exercise 5 conditions

%cash condition
lb_5 = zeros(j_5,1);
ub_5 = ones(j_5,1);
r = 3:0.25:9;
len = length(r);
result6_vec_sigma = zeros(1,len);
result6_vec_mu = zeros(1,len);
e = ones(1,j_5);
mu_plus_e_5 = [mu_5'; e];
count = 1;

for count = 1:len

    [x,fval,exitflag,output] = quadprog(C_5,[],e,1,mu_5',r(count),lb_5,ub_5);
    sigma_result = sqrt(x' * C_5 * x);
    mu_result = (mu_5' * x);
    result6_vec_sigma(1,count) = sigma_result;
    result6_vec_mu(1,count) = mu_result;
    count = count+1;

end

%higher return condition

lb_5 = zeros(j_5,1);
ub_5 = ones(j_5,1);
r = 3:0.25:9;
len = length(r);
result7_vec_sigma = zeros(1,len);
result7_vec_mu = zeros(1,len);
e = ones(1,j_5);
mu_plus_e_5 = [mu_5'; e];
count = 1;

for count = 1:len

    [x,fval,exitflag,output] = quadprog(C_5,[],-mu_5',-r(count),e,1,lb_5,ub_5);
    sigma_result = sqrt(x' * C_5 * x);
    mu_result = (mu_5' * x);
    result7_vec_sigma(1,count) = sigma_result;
    result7_vec_mu(1,count) = mu_result;
    count = count+1;

end

%short selling condition
```

```
lb_5 = [];
ub_5 = ones(j_5,1);
r = 3:0.25:9;
len = length(r);
result8_vec_sigma = zeros(1,len);
result8_vec_mu = zeros(1,len);
e = ones(1,j_5);
mu_plus_e_5 = [mu_5'; e];
count = 1;

for count = 1:len

    [x,fval,exitflag,output] = quadprog(C_5,[],[],[],mu_plus_e_5,[r(count); 1],lb_5,ub_5);
    sigma_result = sqrt(x' * C_5 * x);
    mu_result = (mu_5' * x);
    result8_vec_sigma(1,count) = sigma_result;
    result8_vec_mu(1,count) = mu_result;
    count = count+1;

end

plot(result5_vec_sigma, result5_vec_mu, result6_vec_sigma, result6_vec_mu, result7_vec_sigm
title('Mu(x) and sigma(x) plotted for all values of r and conditions for exercise 5')
legend('vanilla', 'cash', 'higher return','short selling', 'location', 'northwest')
xlabel('sigma(x)')
ylabel('mu(x)')


%% Plots

hold on

% Figure 1: The two pairs of vectors corresponding to Exercises 1 and 2.
figure
plot(result1_vec_sigma, result1_vec_mu, result2_vec_sigma, result2_vec_mu)
title('Mu(x) and sigma(x) plotted for exercise 1 and 2')
legend('Exercise 1', 'Exercise 2')
xlabel('sigma(x)')
ylabel('mu(x)')

% Figure 2: The two pairs of vectors corresponding to Exercises 1 and 3.
figure
plot(result1_vec_sigma, result1_vec_mu, result3_vec_sigma, result3_vec_mu)
title('Mu(x) and sigma(x) plotted for exercise 1 and 3')
legend('Exercise 1', 'Exercise 3')
xlabel('sigma(x)')
ylabel('mu(x)')
```

```
% Figure 3: The two pairs of vectors corresponding to Exercises 1 and 4.
figure
plot(result1_vec_sigma, result1_vec_mu, result4_vec_sigma, result4_vec_mu)
title('Mu(x) and sigma(x) plotted for exercise 1 and 4')
legend('Exercise 1', 'Exercise 4')
xlabel('sigma(x)')
ylabel('mu(x)')

% Figure 4: The two pairs of vectors corresponding to Exercises 1 and 5.
figure
plot(result1_vec_sigma, result1_vec_mu, result5_vec_sigma, result5_vec_mu)
title('Mu(x) and sigma(x) plotted for exercise 1 and 5')
legend('Exercise 1', 'Exercise 5')
xlabel('sigma(x)')
ylabel('mu(x)')
```