1.

The exact price of geometric Asian call option is

>> AsianGeomCall

price =

4.0655

2.

	Crude MC	Control MC
	4.5564	4.3522
	4.1762	4.3374
	4.3601	4.3416
	4.2992	4.3493
	4.1039	4.3452
	4.3855	4.3376
	4.0470	4.3398
	4.0604	4.3360
	4.3605	4.3496
	4.1892	4.3351
mean	4.2538	4.3424
std	0.1655	0.0063

95% confidence interval is [mean ±1.96*std/sqrt(n)]

So 95% confidence interval for Crude **Monte Carlo is [4.1513, 4.3564]**;

For Control Monte Carlo is [4.3385, 4.3463]

	Crude MC	Control MC
MSE	0.02465	0.00004

Comparing Crude Monte Carlo Control Monte Carlo has a less standard deviation and a less mean square error. So Control Monte Carlo method is better and improve the mean square error from 0.025 to 4e-5.

Appendix:

Code for part 1,

```
function AsianGeomCall

r=0.05;
```

```
S0=50;
K=50;
sigma=0.3;
n=10;
t=[0.1:0.1:1];
i=[1:1:10];
expSG=S0*exp((r-sigma^2/2)*(sum(t)/n)+sigma^2/2/n^2*sum((2*n+1).*t-
2.*i.*t));
var=sigma^2/n^2*sum((2*n+1).*t-2*i.*t);
d1=(log(expSG/K)+var/2)/sqrt(var);
d2=(log(expSG/K)-var/2)/sqrt(var);

price=exp(-r*1)*(expSG*normcdf(d1)-K*normcdf(d2));
price
```

Code for Asian arithmetic call option using Crude Monte Carlo

```
function CrudeAsianAriCall
path = 1000;
num_est = 10;
dim = 10;
x=zeros(dim,path);

option = zeros(1, num_est);
for k = 1: num_est
    u=rand(dim,path);
    for i = 1:path
        for j = 1:2:dim
            tempp = BoxMuller(u(j,i),u(j+1,i));
            x(j,i)=tempp(1);
            x(j+1,i)=tempp(2);
        end
    end

    std = 0.3;
    sumtemp = 0.0;
    ssum=0.0;
    s0 = 50;
    r=0.05;
    strike = 50;
    for i = 1:path
        s = s0;
        for j = 1:dim
            s = s * exp( ( r - std * std / 2 ) * 0.1 + std * sqrt(0.1)
* x( j , i ) );
            ssum=ssum+s;
        end
        ssum=(ssum-50)/dim;
        sumtemp = sumtemp + max( ssum - strike,0 );
    end
    option(1,k) = exp(-r)*(sumtemp/path);
end
option'
end
```

Code for Asian arithmetic call option using Control Monte Carlo

```
function ControlAsianAriCall
dbstop if error

path = 1000;
num_est = 10;
dim = 10;
x=zeros(dim,path);
C=4.0655;

option = zeros(1, num_est);
for k = 1: num_est
    u=rand(dim,path);
    for i = 1:path
        for j = 1:2:dim
            tempp = BoxMuller(u(j,i),u(j+1,i));
            x(j,i)=tempp(1);
            x(j+1,i)=tempp(2);
        end
    end

    std = 0.3;
    asum=0.0;
    gsum=1.0;
    s0 = 50;
    r=0.05;
    strike = 50;
    a=[];
    g=[];
    for i = 1:path
        s = s0;
        for j = 1:dim
            s = s * exp( ( r - std * std / 2 ) * 0.1 + std * sqrt(0.1)
* x( j , i ) );
            asum=asum+s;
            gsum=gsum*s;
        end
        asum=(asum-50)/dim;
        gsum=(gsum/50)^(1/dim);
        a(i)= exp(-r)*max( asum - strike,0 );
        g(i)= exp(-r)*max( gsum - strike,0 );
    end
    ag=(a-mean(a)).*(g-mean(g));
    gg=(g-mean(g)).*(g-mean(g));
    betatemp=sum(ag)/sum(gg);
    option(1,k) = 1/path*sum(a-betatemp.*(g-C));
end
option'
end
```