Listing 1.1 Main01.cpp
```
#include <iostream>
using namespace std;
int main()
{
cout << "Hi there" << endl;
char x; cin >> x;
return 0;
}
```

Listing 1.2 Main02.cpp
```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
double S0,U,D,R;
cout << "Enter S0: "; cin >> S0;
cout << "Enter U: "; cin >> U;
cout << "Enter D: "; cin >> D;
cout << "Enter R: "; cin >> R;
cout << endl;
if (S0<=0.0 || U<=-1.0 || D<=-1.0 || U<=D
|| R<=-1.0)
{
cout << "Illegal data ranges" << endl;
cout << "Terminating program" << endl;
return 1;
}
if (R>=U || R<=D)
{
cout << "Arbitrage exists" << endl;
cout << "Terminating program" << endl;
return 1;
}
cout << "Input data checked" << endl;
cout << "There is no arbitrage" << endl << endl;
cout << "q = " << (R-D)/(U-D) << endl;
int n=3; int i=2;
cout << "n = " << n << endl;
cout << "i = " << i << endl;
cout << "S(n,i) = " << S0*pow(1+U,i)*pow(1+D,n-i)
<< endl;
return 0;
}
```

Listing 1.3 Main03.cpp
```
#include <iostream>
#include <cmath>
using namespace std;
double RiskNeutProb(double U, double D, double R)
```

```
{
return (R-D)/(U-D);
}
double S(double S0, double U, double D, int n, int i)
{
return S0*pow(1+U,i)*pow(1+D,n-i);
}
int GetInputData(double& S0,
double& U, double& D, double& R)
{
cout << "Enter S0: "; cin >> S0;
cout << "Enter U: "; cin >> U;
cout << "Enter D: "; cin >> D;
cout << "Enter R: "; cin >> R;
cout << endl;
if (S0<=0.0 || U<=-1.0 || D<=-1.0 || U<=D
|| R<=-1.0)
{
cout << "Illegal data ranges" << endl;
cout << "Terminating program" << endl;
return 1;
}
if (R>=U || R<=D)
{
cout << "Arbitrage exists" << endl;
cout << "Terminating program" << endl;
return 1;
}
cout << "Input data checked" << endl;
cout << "There is no arbitrage" << endl << endl;
return 0;
}
int main()
{
double S0,U,D,R;
if (GetInputData(S0,U,D,R)==1) return 1;
cout << "q = " << RiskNeutProb(U,D,R) << endl;
int n=3; int i=2;
cout << "n = " << n << endl;
cout << "i = " << i << endl;
cout << "S(n,i) = " << S(S0,U,D,n,i) << endl;
return 0;
}
```

Listing 1.4 BinModel01.cpp

```
#include <iostream>
#include <cmath>
using namespace std;
double RiskNeutProb(double U, double D, double R)
{
return (R-D)/(U-D);
}
double S(double S0, double U, double D, int n, int i)
```

```
{
return S0*pow(1+U,i)*pow(1+D,n-i);
}
int GetInputData(double& S0,
double& U, double& D, double& R)
{
cout << "Enter S0: "; cin >> S0;
cout << "Enter U: "; cin >> U;
cout << "Enter D: "; cin >> D;
cout << "Enter R: "; cin >> R;
cout << endl;
if (S0<=0.0 || U<=-1.0 || D<=-1.0 || U<=D
|| R<=-1.0)
{
cout << "Illegal data ranges" << endl;
cout << "Terminating program" << endl;
return 1;
}
if (R>=U || R<=D)
{
cout << "Arbitrage exists" << endl;
cout << "Terminating program" << endl;
return 1;
}
cout << "Input data checked" << endl;
cout << "There is no arbitrage" << endl << endl;
return 0;
}
```

Listing 1.5 Main04.cpp

```
#include "BinModel01.h"
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
double S0,U,D,R;
if (GetInputData(S0,U,D,R)==1) return 1;
cout << "q = " << RiskNeutProb(U,D,R) << endl;
int n=3; int i=2;
cout << "n = " << n << endl;
cout << "i = " << i << endl;
cout << "S(n,i) = " << S(S0,U,D,n,i) << endl;
return 0;
}
```

Listing 1.6 BinModel01.h

```
#ifndef BinModel01_h
#define BinModel01_h
double RiskNeutProb(double U, double D, double R);
double S(double S0, double U, double D, int n, int i);
int GetInputData(double& S0,
double& U, double& D, double& R);
#endif
```

Listing 1.7 Options01.h

```
#ifndef Options01_h
#define Options01_h
int GetInputData(int& N, double& K);
double PriceByCRR(double S0, double U, double D,
double R, int N, double K);
double CallPayoff(double z, double K);
#endif
```

Listing 1.8 Options01.cpp

```
#include "Options01.h"
#include "BinModel01.h"
#include <iostream>
#include <cmath>
using namespace std;
int GetInputData(int& N, double& K)
{
cout << "Enter steps to expiry N: "; cin >> N;
cout << "Enter strike price K: "; cin >> K;
cout << endl;
return 0;
}
double PriceByCRR(double S0, double U, double D,
double R, int N, double K)
{
double q=RiskNeutProb(U,D,R);
double Price[N+1];
for (int i=0; i<=N; i++)
{
Price[i]=CallPayoff(S(S0,U,D,N,i),K);
}
for (int n=N-1; n>=0; n--)
{
for (int i=0; i<=n; i++)
{
Price[i]=(q*Price[i+1]+(1-q)*Price[i])/(1+R);
}
}
return Price[0];
}
double CallPayoff(double z, double K)
{
```

```
if (z>K) return z-K;
return 0.0;
}
```

Listing 1.9 Main05.cpp

```
#include "BinModel01.h"
#include "Options01.h"
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
double S0,U,D,R;
if (GetInputData(S0,U,D,R)==1) return 1;
int N;
cout << "Enter call option data:" << endl;
GetInputData(N,K);
cout << "European call option price = "
<< PriceByCRR(S0,U,D,R,N,K)
<< endl << endl;
return 0;
}
```

Listing 1.10 Options02.h

```
#ifndef Options02_h
#define Options02_h
int GetInputData(int* PtrN, double* PtrK);
double PriceByCRR(double S0, double U, double D,
double R, int N, double K);
double CallPayoff(double z, double K);
#endif
```

Listing 1.11 Options02.cpp

```
#include "Options02.h"
#include "BinModel01.h"
#include <iostream>
#include <cmath>
using namespace std;
int GetInputData(int* PtrN, double* PtrK)
{
cout << "Enter steps to expiry N: "; cin >> *PtrN;
cout << "Enter strike price K: "; cin >> *PtrK;
cout << endl;
return 0;
}
double PriceByCRR(double S0, double U, double D,
double R, int N, double K)
{
```

```
double q=RiskNeutProb(U,D,R);
double Price[N+1];
for (int i=0; i<=N; i++)
{
*(Price+i)=CallPayoff(S(S0,U,D,N,i),K);
}
for (int n=N-1; n>=0; n--)
{
for (int i=0; i<=n; i++)
{
*(Price+i)=(q*(*(Price+i+1))+(1-q)*(*(Price+i)))/(1+R);
}
}
return *Price;
}
double CallPayoff(double z, double K)
{
if (z>K) return z-K;
return 0.0;
}
```

Listing 1.12 Main06.cpp

```
#include "BinModel01.h"
#include "Options02.h"
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
double S0,U,D,R;
if (GetInputData(S0,U,D,R)==1) return 1;
double K;
int N;
cout << "Enter call option data:" << endl;
GetInputData(&N,&K);
cout << "European call option price = "
<< PriceByCRR(S0,U,D,R,N,K)
<< endl << endl;
return 0;
}
```

Listing 1.13 Options03.h

```
#ifndef Options03_h
#define Options03_h
int GetInputData(int& N, double& K);
double PriceByCRR(double S0, double U, double D,
double R, int N, double K,
double (*Payoff)(double z, double K));
double CallPayoff(double z, double K);
double PutPayoff(double z, double K);
```

```
#endif
```

Listing 1.14 Options03.cpp

```
#include "Options03.h"
#include "BinModel01.h"
#include <iostream>
#include <cmath>
using namespace std;
int GetInputData(int& N, double& K)
{
cout << "Enter steps to expiry N: "; cin >> N;
cout << "Enter strike price K: "; cin >> K;
cout << endl;
return 0;
}
double PriceByCRR(double S0, double U, double D,
double R, int N, double K,
double (*Payoff)(double z, double K))
{
double q=RiskNeutProb(U,D,R);
double Price[N+1];
for (int i=0; i<=N; i++)
{
Price[i]=Payoff(S(S0,U,D,N,i),K);
}
for (int n=N-1; n>=0; n--)
{
for (int i=0; i<=n; i++)
{
Price[i]=(q*Price[i+1]+(1-q)*Price[i])/(1+R);
}
}
return Price[0];
}
double CallPayoff(double z, double K)
{
if (z>K) return z-K;
return 0.0;
}
double PutPayoff(double z, double K)
{
if (z<K) return K-z;
return 0.0;
}
```

Listing 1.15 Main07.cpp

```
#include "BinModel01.h"
#include "Options03.h"
#include <iostream>
#include <cmath>
```

```
using namespace std;
int main()
{
double S0,U,D,R;
if (GetInputData(S0,U,D,R)==1) return 1;
double K;
int N;
cout << "Enter call option data:" << endl;
GetInputData(N,K);
cout << "European call option price = "
<< PriceByCRR(S0,U,D,R,N,K,CallPayoff)
<< endl << endl;
cout << "Enter put option data:" << endl;
GetInputData(N,K);
cout << "European put option price = "
<< PriceByCRR(S0,U,D,R,N,K,PutPayoff)
<< endl << endl;
return 0;
}
```