Assignment guidlines:

- You should not hand in a first draft. Rewrite your solutions carefully and neatly. Use complete sentences. Make sure your arguments are clear.
- Unless otherwise specified MACF students should use an object-oriented programming language (*C++* or *Java*) for implementations/programming problems. Graduate and actuarial students may use *R* or *Matlab* if they wish but they are be required to implement everything from scratch without using any built-in mathematical functions.
- Your solutions should include the source code for all necessary functions, classes, etc. Your code should also be commented appropriately to indicate what your are doing in each step. Functions should be sufficiently general so they can be reused in other programming applications. You may be asked to demonstrate your program in class so that I can make sure it works. Please attach print-outs of you code with your written assignments.

Problems:

1. Consider the parameterization of the N-step binomial model with $\tilde{p} = \tilde{q} = 1/2$, $\Delta t = T/N$, continuously compounded interest rate R,

$$u = e^{R\Delta t}(1 + \sqrt{e^{\sigma^2 \Delta t} - 1}), \quad \text{and}$$
$$d = e^{R\Delta t}(1 - \sqrt{e^{\sigma^2 \Delta t} - 1})$$

(a) Show that

$$u = 1 + \sigma\sqrt{\Delta t} + R\Delta t + O(\Delta t^{3/2}) \quad \text{and}$$
$$d = 1 - \sigma\sqrt{\Delta t} + R\Delta t + O(\Delta t^{3/2})$$

as $\Delta t \to 0$.
[Hint: recall that $\sqrt{1+x} = 1 + \frac{1}{2}x + O(x^2)$.]

(b) **[MAST 729H/881 Only]** Consider the parameterization of the N-step binomial model with $\Delta t = T/N$, continuously compounded interest rate R,

$$v = e^{\sigma^2 \Delta t} \tag{1}$$
$$u = \frac{1}{2} v e^{R\Delta t}(v + 1 + \sqrt{v^2 + 2v - 3}) \tag{2}$$
$$d = \frac{1}{2} v e^{R\Delta t}(v + 1 - \sqrt{v^2 + 2v - 3}) \tag{3}$$

and risk-neutral probability of an up-step $\tilde{p} = \frac{e^{R\Delta t} - d}{u - d}$.
Show that the expected return of the stock price over an interval of length Δt in the above model is given by $r = 1 - e^{R\Delta t}$.
[Note that this matches the expected return, under the risk-neutral measure, of the stock price over the same time interval in the Black-Scholes model.]

2. **[Binomial Implied Volatility]** Consider the N-step binomial model for the asset price

$$S_n^{i,N} = u^n d^{N-n} S_0$$

where $0 \le n \le i$, $0 \le i \le N$, and n denotes the number of *heads* or *up-steps* in the binomial tree for S up to time-step i. Let $V_n^{i,N}$ denote the corresponding value of a European option on S at node (i, n) . Then we can show that

$$V_0^{0,N} = e^{-rT} \sum_{k=0}^{N} \binom{N}{k} \tilde{p}^k \tilde{q}^{N-k} V_k^{N,N} \tag{4}$$

where $\binom{N}{k}$ denotes the binomial coefficient $\binom{N}{k} = \frac{N!}{k!(N-k)!}$.

(a) With $\tilde{p} = \tilde{q} = 1/2$, $\Delta t = T/N$, and continuously compounded interest rate R where u and d are given by

$$u = e^{\sigma\sqrt{\Delta t} + (R - \frac{1}{2}\sigma^2)\Delta t}, \quad \text{and} \tag{5}$$

$$d = e^{-\sigma\sqrt{\Delta t} + (R - \frac{1}{2}\sigma^2)\Delta t}. \tag{6}$$

write a program which implements formula (4) as a function of the inputs $(S_0, R, \sigma, K, \tau, N)$ to find the price, C_0^N, at time zero of a European call option. Use small scale examples you can work out by hand to test your program. Create a corresponding program for put options. *[Hint: For large values of N you may run into computational problems calculating the binomial coefficients. Consider using log-binomial coefficients and Stirling's approximation:*

$$\ln(n!) = n \ln(n) - n + O(\ln(n)) \quad \textit{or} \quad n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n.$$

There are other simplifications that can be made to avoid computational difficulties.]

(b) Write a program that applies the bisection algorithm to the program from part (a), given starting values that satisfy the appropriate conditions, to solve

$$C_0^{0,N} = C_0^{obs}$$

for the **binomial implied volatility** where C_0^{obs} is an observed market price. Specify an error tolerance parameter ϵ such that the bisection algorithm stops when $|C_0^{0,N} - C_0^{obs}| < \epsilon$. Modify your code to create a program which calibrates to observed put option prices.

(c) Apply your programs from part (a)-(b) to the observed price data from July 29, 2002 given below to calculate the binomial implied volatilities for call and put option ask prices which expire on September 26, 2002. Assume $S_0 = 4.75$, and $R = 0.0492$ per annum (use a time scale in units of years with 365 days per year, you will need to calculate the number of years T to expiry). Assume $\epsilon = 10^{-6}$. Calculate each of the implied volatilities for increasing $N = 10^k$, $k = 1, 2, \ldots, 5$ (or more if your programs allow) and report your results in a table. Comment on convergence.

K	Call price-ask	Put price-ask
4.00	-	0.02
4.25	-	0.04
4.50	0.33	0.09
4.75	0.16	0.20
5.00	0.06	0.38
5.25	0.02	0.59
5.50	0.01	-
5.75	0.01	-

(d) Implied volatilities should be the same, all else equal, whether we calibrate to market call or put prices. Practitioners calibrate to out-of-the-money option prices to estimate implied volatility. Plot your results from part (b) as a function of strike-price for the out-of-the-money options using largest value N for you were able to calculate. Is there a 'volatility smile'?