Demonstrate the “specials” or your implementation such as own research, re-coding of numeric methods, using the industrial-strength libraries of C++, Pyton or NAG

HJM

1. Get Historical Instantenuous Forward Rates (Bank Liability Curve – BLC; LIBOR linked instrument)
2. Get SONIA / OIS rates. OIS Spread over LIBOR?  
   read Bank of England paper (2001)  
   OIS discounting – general background. Sol Steinberg
3. Construct matrix of difference or return or log return
4. Generate the covariance matrix from this difference matrix
5. Verify that the matrix is positive definite (i.e. that we can find eigen values)
6. Apply Jacobi transformation to determine eigen value of the covariance matrix and eigen vectors
7. Sort eigen value and select the largest one (3 largest one). Identify which tenor they represent, these are our factors (PC1, PC2, PC3)
8. Perform curve fitting for PC1, PC2, PC3. i.e .determine the coefficients b0, b1, b2 and b3 in order to determine the functions which will proxy our principal factors  
   equivalent of LINEST in Excel
9. Use a numerical integral integration technique (instead of trapezium rule) to calculate m\_bar (drift)
10. Create a function that generation a matrix (for one simulation)   
    think of using quasi random number like Sobol
11. Bootstrap or obtain a discount curve (spot curve) like an OIS curve  
    Worse case, discount using the HJM matrix (not ideal)
12. Build pricing function for:  
    Bonds  
    Caps and Floors  
    Vanilla Swaption or Bermodan Swapption (optional)  
      
    Obtain the continuously compounded implied Forward LIBOR using numerical integration over a fitted curve instead of simple averaging
13. Back-out implied vol for Caps and Floors and Swaptions as these products are generally quoted in term of Black76 volatilities

LMM (very optional)

1. A revoir