Matrices and Array HW-7

1) Develop a VB program to compute the areas of “n” number of squares and circles, given their characteristic dimensions (length of the side for squares or radius for circles). Use matrices to store the results (details in lab problem-1).

2) Modify the above code and use two subroutines square and circle to find the areas (details in lab problem 2)

3) Develop a VB to multiply a n x n square matrix with a n x 1 column matrix (lab problem 3) test it with an example and give print out of the code (details in lab problem 3).

4) Sort the following 12 numbers from low to high: 55, 12, 33, 90, 17, 88, 97, 23, 49, 32, 78, and 123 (modify the lab problem 4 to solve this).

5) Develop a general VB code that can implement the backward substitution method to solve a general (n x n) upper triangular matrix. Use the problem below to demonstrate how the inputs will be defined and solve the following LAE problem. Note the code should be general enough to solve any (n x n) system.

\[
\begin{align*}
4x_1 + 2x_2 + x_3 - 3x_4 + x_5 &= 4 \\
5x_2 + x_3 + 3x_4 + 1x_5 &= 30 \\
-3x_3 + x_4 + 2x_5 &= 5 \\
5x_4 + 2x_5 &= 30 \\
-2x_5 &= -10
\end{align*}
\]

Adapt the following algorithm to develop the VB code

\[
\begin{align*}
x(n) &= b(n) / a(n, n) \\
For \ i = n - 1 \ To \ 1 \ Step \ -1 \\
sum &= 0# \\
For \ j = i + 1 \ To \ n \\
\sum &= \sum + a(i, j) * x(j) \\
Next \ j \\
x(i) &= (b(i) - \sum) / a(i, i) \\
Next \ i
\end{align*}
\]

6) Solve the above problem using EXCEL and give screen prints of your EXCEL setup.

7) Develop a general VB code that can compute the determinant of any arbitrary 3 x 3 matrix (the method was discussed in class). Test the code by computing the determinant of the matrix A in the following LAE problem. Ans: -69

\[
\begin{align*}
-3y + 7z &= 2 \\
x + 2y - z &= 3 \\
5x - 2y &= 2
\end{align*}
\]

8) Hand solve the above LAE problem using the Cramers rule. Cross check your answer using EXCEL.

9) Hand solve the following problem (show steps) and give the output of the code. Run the code and verify your hand calculated results

```vbnet
Private Sub CommandButton1_Click()
Dim i As Integer, a(3) As Single, b(3) As Single, c(3) As Single
Dim ff As Single, d(3) As Single, x(3) As Single
a(1) = 1
a(2) = 9
a(3) = 12
b(1) = 2
b(2) = 4
b(3) = 7
c(1) = 1
c(2) = 2
```
c(3) = 3
d(1) = 8
d(2) = 0
d(3) = 0
For i = 2 To 3
    ff = a(i) / b(i - 1)
    If (ff > 5#) Then
        b(i) = b(i) - (c(i - 1) * ff)
        d(i) = d(i) - (d(i - 1) * ff)
    Else
        b(i) = b(i) - (c(i - 1) / ff)
        d(i) = d(i) - (d(i - 1) / ff)
    End If
Next i
For i = 1 To 3
    x(i) = d(i) / b(i)
Next i
For i = 1 To 3
    Cells(i, 1) = a(i)
    Cells(i, 2) = b(i)
    Cells(i, 3) = c(i)
    Cells(i, 4) = d(i)
    Cells(i, 5) = x(i)
Next i
End Sub

10) Develop a general VB code that can solve the A x = b problem for an arbitrary 3 x 3 matrix system using the Cramer’s rule. Use the determinant code developed in previous HW as a subroutine and call this routine to implement the Cramer’s rule to solve the LAE problem discussed in Problem-8. Ans: 0.9855, 1.4638, 0.913
Program these Steps: Read matrix A
Copy it three times and make three matrices A1, A2, A3
Replace first column in A1 with b
Replace second column in A2 with b
Replace first column in A3 with b
Call determinant subroutine 4 times and use determinants to evaluate x1, x2, x3
sub mydeterminant (Amat() as single, determinantValue as single)
Term1 = a11*(a22*a33 – a32*a23). Similarly, define term2 and term3
determinantvalue = term1+term2+term3