

```

/* Annuity Coding Example */                                line 1

# include <stdio.h>                                          line 2
# include <math.h>                                           line 3

double Annuity_Amount (double, double, int);                line 4

int main ()                                                  line 5
{
    double Future_Value = 500000.;                          line 6
    double Interest_Rate = 0.10/12;                         line 7
    int Period = 20*12;                                     line 8
                                                            line 9

    double A_A = Annuity_Amount (Future_Value, Interest_Rate, Period);line10

    printf ("Monthly Payment or Annuity Amount = %7.3lf\n", A_A); line11

    return 0;                                                line12
}                                                            line13
double Annuity_Amount (double F_V, double I_R, int Per)    line14
{                                                            line15
    return F_V*I_R/(pow((1+I_R),Per)-1);                    line16
}                                                            line17

```

**FIGURE A-T.2**

```

/* Future Value of Monthly Annuity Payments*/                                line 1

# include <stdio.h>                                                            line 2
# include <math.h>                                                            line 3

void Annuity_Payout(double, int, double, double[], double[]);                line 4

int main ()                                                                    line 5
{
    int    i_counter;                                                         line 6
    double Payment = 658.44;                                                  line 7
    double Interest_Rate = 0.10/12;                                          line 8
    int Period = 20*12;                                                       line 9
                                         line10

    double Future_Value_Interest[1000];                                       line11
    double Future_Value_Annuity_Payout[1000];                                line12

    Annuity_Payout(Interest_Rate, Period, Payment, Future_Value_Interest,
                   Future_Value_Annuity_Payout); line13

    printf("Month      End of Month      Interest Earned      Total \n"); line14
    printf("          Payment              for the month      Payment \n"); line15

    for (i_counter = 1; i_counter <=Period; i_counter++)                    line16
    {
        printf ("   %3d          %6.2f      %6.2f          %8.2f\n",          line17
            i_counter, Payment, Future_Value_Interest[i_counter] ,
            Future_Value_Annuity_Payout[i_counter]);                          line18
    }                                                                           line19

    return 0;                                                                  line20
}                                                                               line21


void Annuity_Payout(double I_R, int Per, double Pay, double FVI[1000]
, double FVAP[1000])                                                         line22
{
    int i;                                                                     line23
                                         line24

    FVAP[0] = 0.;                                                             line25
    for (i = 1; i <=Per; i++)                                                  line26
    {
        FVI[i]  = FVAP[i-1]*I_R;                                              line27
        FVAP[i] = FVAP[i-1] + Pay + FVI[i];                                  line28
    }                                                                           line29
                                         line30
}                                                                               line31

```

**FIGURE A-T.3**

```

/* Annuity Coding Examples */                                line 1

# include <stdio.h>                                           line 2
#include <math.h>                                              line 3

double Annuity_Amount (double, double, int);                 line 4
void Annuity_Payout(double, int, double, double[], double[]); line 5

int main ()                                                  line 6
{
    int i_counter;                                           line 7
    double Future_Value = 500000.;                          line 8
    double Interest_Rate = 0.10/12;                         line 9
    int Period = 20*12;                                       line10
    double Future_Value_Interest[1000];                     line11
    double Future_Value_Annuity_Payout[1000];               line12
    double A_A = Annuity_Amount (Future_Value, Interest_Rate, Period); line14

    printf ("Monthly Payment or Annuity Amount = %7.3lf\n\n", A_A); line15

    double Payment = A_A;                                     line16

    Annuity_Payout(Interest_Rate, Period, Payment, Future_Value_Interest,
                   Future_Value_Annuity_Payout); line17

    printf("Month      End of Month  Interest Earned    Total \n"); line18
    printf("Payment    for the month    Payment \n"); line19

        for (i_counter = 1; i_counter <=Period; i_counter++) line20
        {
            printf (" %3d      %6.2f      %6.2f      %8.2f\n", line21
                i_counter, Payment, Future_Value_Interest[i_counter] ,
                Future_Value_Annuity_Payout[i_counter]); line22
        } line23

    return 0;                                                 line24
}                                                             line25


double Annuity_Amount (double F_V, double I_R, int Per)     line26
{
    return F_V*I_R/(pow((1+I_R),Per)-1);                     line27
}                                                             line28
                                                             line29

void Annuity_Payout(double I_R, int Per, double Pay, double FVI[1000], double
FVAP[1000]) line30
{
    int i;                                                    line31
    FVAP[0] = 0.;                                             line32

    FVAP[0] = 0.;                                             line33
    for (i = 1; i <=Per; i++) line34
    {
        FVI[i] = FVAP[i-1]*I_R; line35
        FVAP[i] = FVAP[i-1] + Pay + FVI[i]; line36
    } line37
}                                                             line38
                                                             line39

```

**FIGURE A-T.4**

```

#include "Annuity_Calcs_Lib1.h"                                line 1

/*System header files and function prototypes are moved to Annuity_Calcs_Lib.h
which will serve as the library header file                    line 2

double Annuity_Amount (double F_V, double I_R, int Per)        line 3
{                                                                line 4
    return F_V*I_R/(pow((1+I_R),Per)-1);                       line 5
}                                                                line 6

void Annuity_Payout(double I_R, int Per, double Pay, double FVI[1000], double
FVAP[1000])                                                      line 7
{                                                                line 8
    int i;                                                       line 9

    FVAP[0] = 0.;                                                line10
    for (i = 1; i <=Per; i++)                                    line11
    {                                                            line12
        FVI[i] = FVAP[i-1]*I_R;                                  line13
        FVAP[i] = FVAP[i-1] + Pay + FVI[i];                    line14
    }                                                            line15
}                                                                line16

```

**FIGURE A-T.5**

```
# include <stdio.h>                                line 1
#include <math.h>                                    line 2

double Annuity_Amount (double , double , int );     line 3
void Annuity_Payout(double I_R,int Per, double Pay,double* FVI, double *FVAP);
                                                    line 4
```

**FIGURE A-T.6**

```
Option Explicit line1

Public Declare Function Annuity_Amount Lib " C:\POEA\Bridging Excel and C
Codes\Problems\Time_Value_Money_Combined_Lib1\Debug\Time_Value_Money_Combined
_Lib1.dll" (ByVal Future_Value As Double, ByVal Interest_Rate As Double,
ByVal Period As Long) As Double line2

Public Declare Sub Annuity_Payout Lib " C:\POEA\Bridging Excel and C
Codes\Problems\Time_Value_Money_Combined_Lib1\Debug\Time_Value_Money_Combined
_Lib1.dll" (ByVal Interest_Rate As Double, ByVal Period As Long, ByVal
Payment As Double, ByRef Future_Value_Interest As Double, ByRef
Future_Value_Annuity_Payout As Double) line3
```

**FIGURE A-T.7**

```

Option Explicit                                                                    line 1

Public Declare Function Annuity_Amount Lib " C:\POEA\Bridging Excel and C
Codes\Problems
\Time_Value_Money_Combined_Lib1\Debug\Time_Value_Money_Combined_Lib1.dll"
(ByVal Future_Value As Double, ByVal Interest_Rate As Double, ByVal Period As
Long) As Double                                                                    line 2

Public Declare Sub Annuity_Payout Lib " C:\POEA\Bridging Excel and C
Codes\Problems
\Time_Value_Money_Combined_Lib1\Debug\Time_Value_Money_Combined_Lib1.dll"
(ByVal Interest_Rate As Double, ByVal Period As Long, ByVal Payment As Double,
ByRef Future_Value_Interest As Double, ByRef Future_Value_Annuity_Payout As
Double)                                                                    line 3

Public Sub Get_Annuity()                                                            line 4
    Dim Future_Value As Double                                                    line 5
    Dim Interest_Rate As Double                                                    line 6
    'In the Declare Function (above) Dim Period As Integer will not work bcz
allocation of size compatibility problems between C++ DLLs and VBA 7
    Dim Period As Long                                                            line 8
    Dim A_A As Double                                                            line 9

    Future_Value = 500000                                                         line10
    Interest_Rate = 0.10/12                                                       line11
    Period = 20 * 12                                                             line12

    A_A = Annuity_Amount(Future_Value, Interest_Rate, Period)                   line13
    Sheet1.Cells(1, 1) = A_A                                                     line14

    Dim Payment As Double                                                         line15
    Dim Future_Value_Interest(0 To 999) As Double                               line16
    Dim Future_Value_Annuity_Payout(0 To 999) As Double                         line17
    Payment = A_A                                                                line18
    Annuity_Payout Interest_Rate, Period, Payment, Future_Value_Interest(0),
Future_Value_Annuity_Payout(0)                                                  line19
    Future_Value_Interest(0)                                                     line20
    Future_Value_Annuity_Payout(0)                                               line21

    Dim i As Integer                                                             line22
    For i = 1 To 240                                                             line23
        Sheet1.Cells(i, 3) = Future_Value_Interest(i)                         line24
        Sheet1.Cells(i, 5) = Future_Value_Annuity_Payout(i)                   line25
    Next i                                                                       line26
End Sub                                                                        line27

```

**FIGURE A-T.8**

```

Public Sub ADD_Vector_Macro()
    Dim C1() As Double
    Dim C2() As Double
    Dim C3() As Double
    Dim NSize As Long

    'Determine the number of elements in each vector - here using second column
    NSize = Application.WorksheetFunction.CountA(Range("B:B"))

    ReDim C1(NSize)
    ReDim C2(NSize)
    ReDim C3(NSize)
    Dim i As Integer
    For i = 0 To NSize - 1
        ' VBA and C vector index from zero. Excel columns index from one.
        C1(i) = Sheet1.Cells(i + 1, 1)
        C2(i) = Sheet1.Cells(i + 1, 2)
    Next i
    Calculate_Vector_Sum(C1(0), C2(0), C3(0), NSize)

    For i = 0 To NSize - 1
        Sheet1.Cells(i + 1, 3) = C3(i)
    Next i

End Sub

Public Sub AVG_Vector_Macro()
    Dim C3() As Double
    Dim AvgC3 As Double
    Dim NSize As Long

    NSize = Application.WorksheetFunction.CountA(Range("C:C"))

    ReDim C3(NSize)

    Dim i As Integer
    For i = 0 To NSize - 1
        C3(i) = Sheet1.Cells(i + 1, 3)
    Next i

    AvgC3 = Calculate_Vector_Average(C3(0), NSize)
    Sheet1.Cells(NSize + 1, 3) = AvgC3

End Sub

```

**FIGURE A.10**

```
#include "CP_headers.h" line 1

double PureEnthalpy(double T1, const double TRef, const double A[]) line 2
{ line 3

    return A[0]*(T1 - TRef) + A[1]/2*(pow(T1, 2) - pow(TRef, 2)) +
           A[2]/3*(pow(T1, 3) - pow(TRef, 3)); line 4
} line 5
```

**FIGURE A.11**

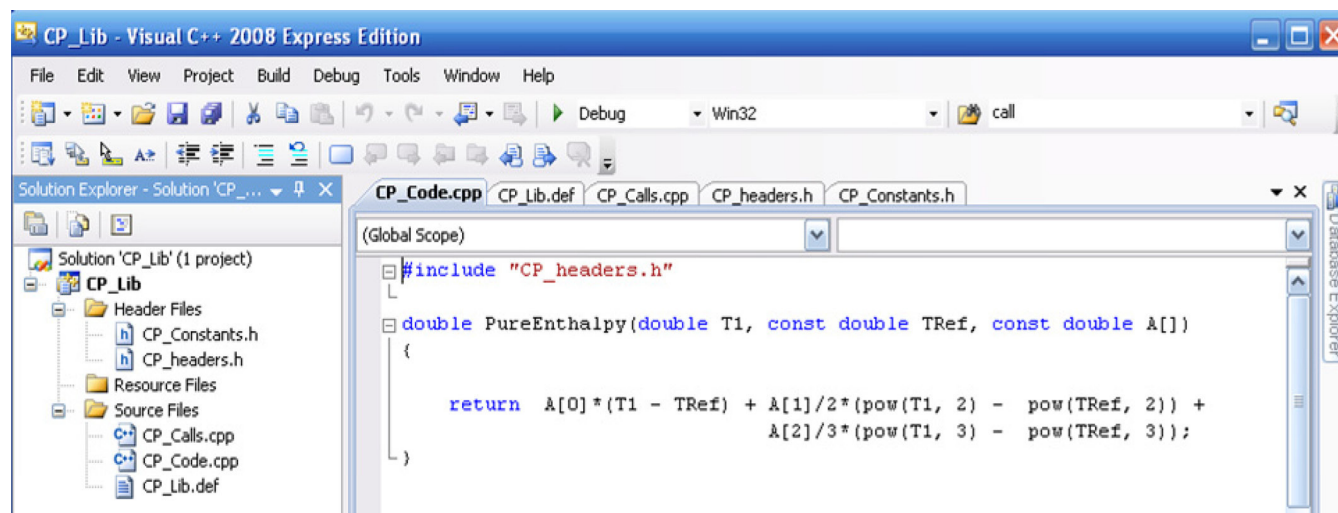


FIGURE A.12

<code>#include "CP_headers.h"</code>	line 1
<code>double H_L_Water(double T1)</code>	line 2
<code>{</code>	line 3
<code>    return PureEnthalpy(T1, T0, WATER_L);</code>	line 4
<code>}</code>	line 5
<code>double H_V_Water(double T1)</code>	line 6
<code>{</code>	line 7
<code>    return PureEnthalpy(T1, T0, WATER_V);</code>	line 8
<code>}</code>	line 9
<code>double H_L_Ethanol(double T1)</code>	line10
<code>{</code>	line11
<code>    return PureEnthalpy(T1, T0, ETHANOL_L);</code>	line12
<code>}</code>	line13
<code>double H_V_Ethanol(double T1)</code>	line14
<code>{</code>	line15
<code>    return PureEnthalpy(T1, T0, ETHANOL_V);</code>	line16
<code>}</code>	line17

**FIGURE A.13**

```
const double WATER_L[] = {75.4E-03, //a
                          0.0000, //b
                          0.0000 //c
                          }; line 1

const double ETHANOL_L[] = {103.1E-03, //a
                            0.00, //b
                            0.000 //c
                            }; line 2

const double WATER_V[] = {33.46E-03, //a
                          0.6880E-05, //b
                          0.7604E-08 //c
                          }; line 3

const double ETHANOL_V[] = {61.34E-03, //a
                            15.72E-05, //b
                            -8.749E-08 //c
                            }; line 4

const double T0 = 25; line 5
```

**FIGURE A.14**

```
#include<stdio.h>                                line 1
#include<math.h>                                   line 2
#include "CP_Constants.h"                         line 3

double H_L_Water(double T1);                      line 4
double H_V_Water(double T1);                      line 5
double H_L_Ethanol(double T1);                   line 6
double H_V_Ethanol(double T1);                   line 7

double PureEnthalpy(double T1, const double T0, const double Generic_A[]);
                                                    line 8
```

**FIGURE A.15**

```
LIBRARY CP_Lib
EXPORTS
    H_L_Water
    H_V_Water
    H_L_Ethanol
    H_V_Ethanol
```

```
line 1
line 2
line 3
line 4
line 5
line 6
```

**FIGURE A.16**

```
Option Explicit                                                    line 1

Public Declare Function H_L_Water Lib " C:\POEA\Bridging Excel and C Codes\
Examples \CP_Lib\Debug\CP_Lib.dll" (ByVal T As Double) As Double    line 2

Public Declare Function H_V_Water Lib " C:\POEA\Bridging Excel and C Codes\
Examples \CP_Lib\Debug\CP_Lib.dll" (ByVal T As Double) As Double    line 3

Public Declare Function H_L_Ethanol Lib " C:\POEA\Bridging Excel and C Codes\
Examples \CP_Lib\Debug\CP_Lib.dll" (ByVal T As Double) As Double    line 4

Public Declare Function H_V_Ethanol Lib " C:\POEA\Bridging Excel and C Codes\
Examples \CP_Lib\Debug\CP_Lib.dll" (ByVal T As Double) As Double    line 5
```

**FIGURE A.17**

```

void ConvertVBA_Matrix_to_C_Matrix(double* Matrix, long Nrows, long Ncolumns
, double Matrix_Row_Major[100][100])
{
    //double Matrix_Row_Major[100][100];  it may be necessary to change the size

    int i;
    int j;

    for(i =0; i<Nrows; ++i)
    {
        int columnPos = 0;
        for(j =0; j<Ncolumns; ++j)
        {
            Matrix_Row_Major[i][j] = *(Matrix+i+columnPos+j);
            columnPos = columnPos + Nrows;
        }
    }

    return;
}

```

**FIGURE A.19**

```

#include <stdio.h>
#include <math.h>

double calculate_sum (double z1, double z2);
double calculate_difference (double z1, double z2);
double calculate_exponential (double z1, double z2);

int main()
{
    double x1, x2, f_sum, f_diff, f_exp;

    x1 = 5.0;
    x2 = 3.0;

    f_sum = calculate_sum (x1, x2);
    f_diff = calculate_difference (x1, x2);
    f_exp = calculate_exponential (x1, x2);

    // Send Answers to Screen
    printf("sum = %lf\n", f_sum);
    printf("diff = %lf\n", f_diff);
    printf("expon = %lf\n", f_exp);

    return 0;
}

double calculate_sum (double z1, double z2)
{
    double sum;
    sum = z1 + z2;
    return sum;
}

double calculate_difference (double z1, double z2)
{
    double difference;
    difference = z1 - z2;
    return difference;
}

double calculate_exponential (double z1, double z2)
{
    double exponential;
    exponential = pow(z1,z2);
    return exponential;
}

```

**FIGURE A.2**

```

void ConvertC_Matrix_to_VBA_Matrix(double Matrix_Row_Major[100][100],
    long Nrows,
    long Ncolumns, double* Matrix_Col_Major)
{
    int i;
    int j;

    for(i =0; i<Nrows; ++i)
    {
        int columnPos = 0;
        for(j =0; j<Ncolumns; ++j)
        {
            *(Matrix_Col_Major+i+columnPos+j) = Matrix_Row_Major[i][j];
            columnPos = columnPos + Nrows;
        }
    }
    return;
}

```

**FIGURE A.20**

```

void GJ_Elimination_Main(double* Matrix, long Nrows, long Ncols)      line 1
{
    double A[100][100];                                              line 2
    ConvertVBA_Matrix_to_C_Matrix(Matrix, Nrows, Ncols, A);          line 3
    GJ_Elimination(A, Nrows);                                         line 4
    ConvertC_Matrix_to_VBA_Matrix(A, Nrows, Ncols, Matrix);          line 5
}                                                                      line 6

void GJ_Elimination(double A[100][100], long row)                    line 7
{
    long col = row+1;                                                 line 8
    int nonZerIdx = 0;                                                line 9

    for(int i =0; i<row; ++i)                                         line 10
    {
        if(A[i][i] == 0.0)                                           line 11
        {
            for (int i2 = i+1; i2<row; ++i2)                         line 12
            if(A[i2][i] != 0)                                         line 13
            {
                nonZerIdx = i2;                                       line 14
                break;                                                line 15
            }
            for(int j1 =0; j1<col; ++j1)                              line 16
            {
                double tmp = A[i][j1];                                line 17
                A[i][j1] = A[nonZerIdx][j1];                          line 18
                A[nonZerIdx][j1] =tmp;                                line 19
            }                                                         line 20

            double tmpAii = A[i][i];                                  line 21
            for(int j =0; j<col; ++j)                                 line 22
                A[i][j] = A[i][j]/tmpAii;                             line 23

            for(int i1 =0; i1<row; ++i1)                              line 24
            {
                if(i1 != i)                                           line 25
                {
                    double tmpAi1 = A[i1][i];                       line 26
                    for(int j =0; j<col; ++j)                         line 27
                        A[i1][j] = A[i1][j] - tmpAi1*A[i][j];        line 28
                }
            }
        }
    }
}

```

**FIGURE A.21**

```

Public Declare Sub GJ_Elimination_Main Lib "C:\POEA\Bridging Excel and
C Codes\Examples \Simple_C_Matrix_dll\Debug\Simple_C_Matrix_dll.dll"
(ByRef Matrix As Double, ByVal Nrows As Long, ByVal Ncolumns As Long) line 1

'comment line 2

Public Sub Gauss_Jordan_Macro() line 3
    Dim C() As Double line 4
    Dim Nrows, Ncolumns As Long line 5

    Nrows = 3 line 6
    Ncolumns = 4 line 7

    ReDim C(Nrows, Ncolumns) line 8
    Dim i As Integer line 9
    Dim j As Integer line 10
    For i = 0 To Nrows - 1 line 11
        For j = 0 To Ncolumns - 1 line 12
            If Sheet1.Cells(i + 8, j + 2) = " " Then line 13
                C(i, j) = 0 line 14
            Else line 15
                C(i, j) = Sheet1.Cells(i + 8, j + 2) line 16
            End If line 17
        Next j line 18
    Next i line 19

    GJ_Elimination_Main C(0, 0), Nrows, Ncolumns line 20

    For i = 0 To Nrows - 1 line 21
        For j = 0 To Ncolumns - 1 line 22
            Sheet1.Cells(i + 12, j + 2) = C(i, j) line 23
        Next j line 24
    Next i line 25
End Sub line 26

```

**FIGURE A.22**

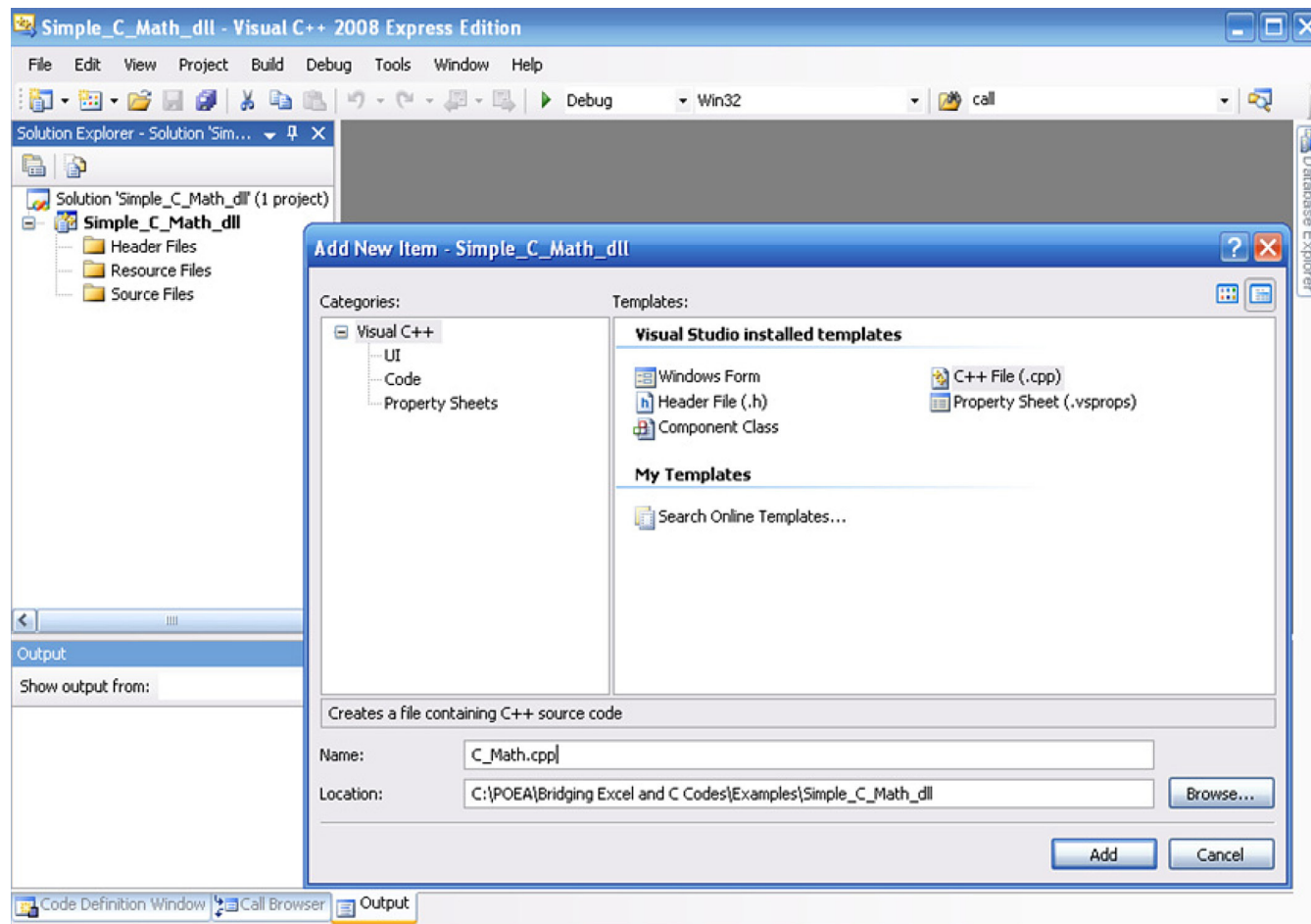


FIGURE A.3

```
#include <math.h>
#include "C_Math.h"

double calculate_sum (double z1, double z2)
{
    double sum;
    sum = z1 + z2;
    return sum;
}

double calculate_difference (double z1, double z2)
{
    double difference;
    difference = z1 - z2;
    return difference;
}

double calculate_exponential (double z1, double z2)
{
    double exponential;
    exponential = pow(z1,z2);
    return exponential;
}
```

**FIGURE A.4**

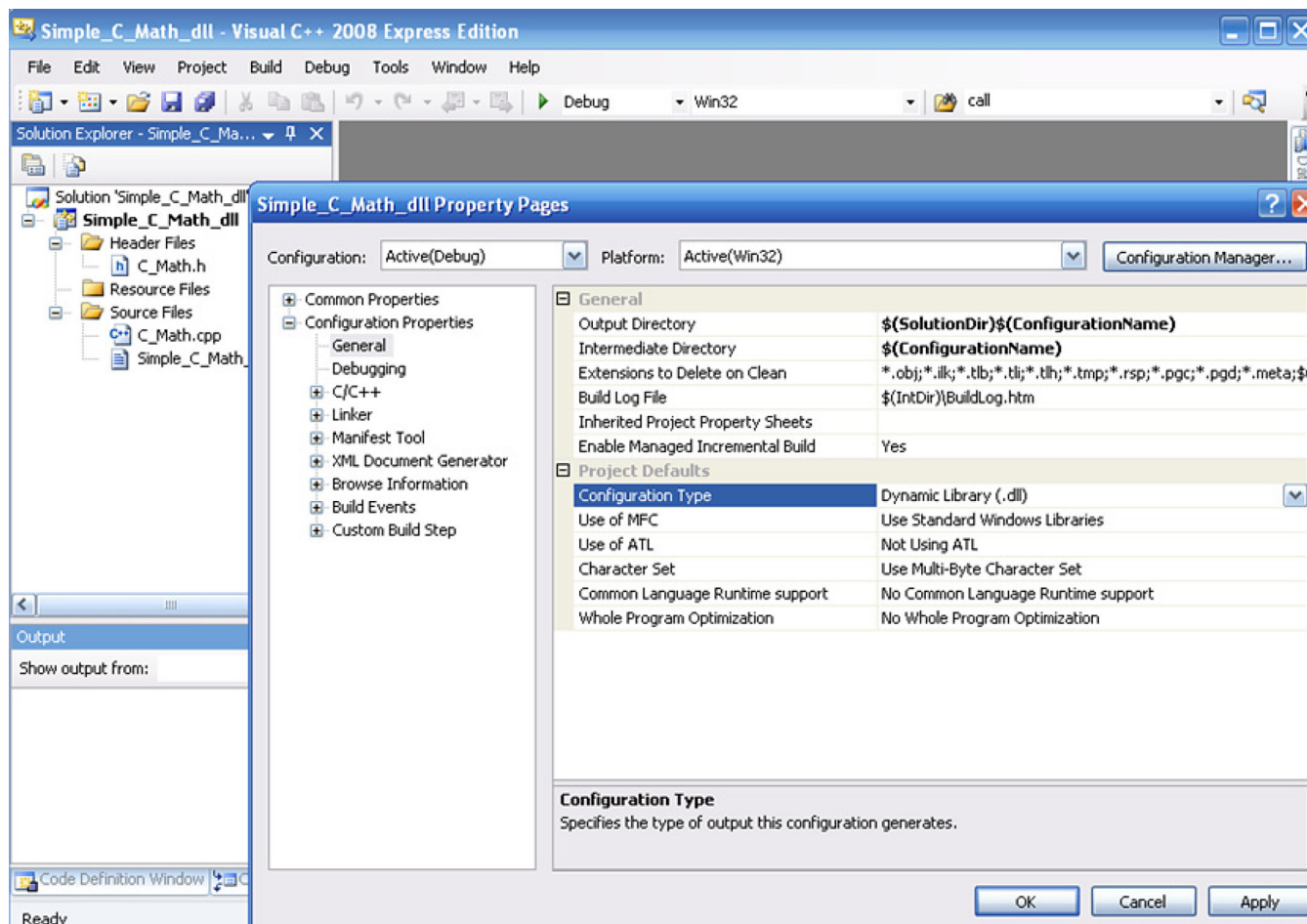


FIGURE A.5

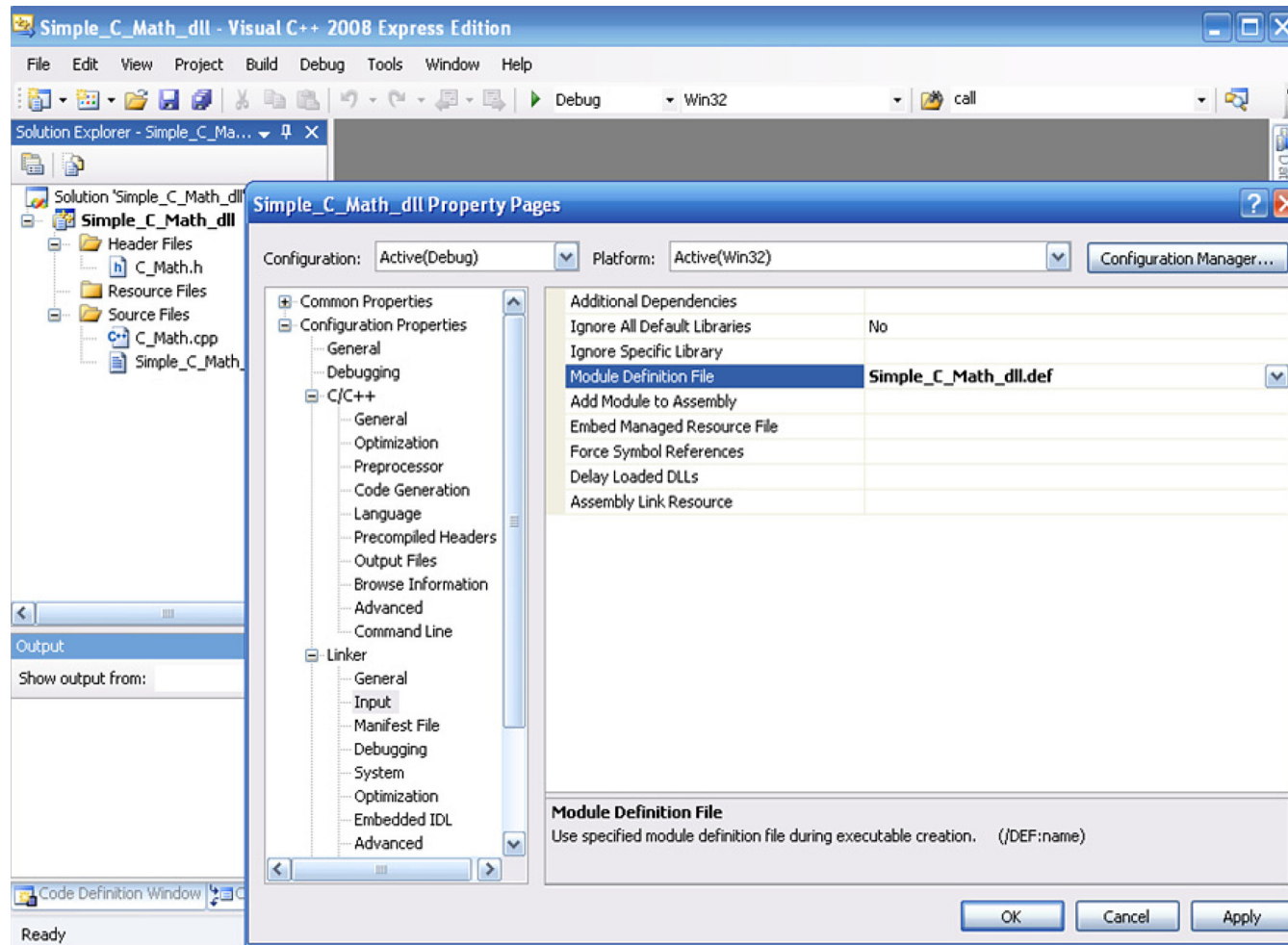


FIGURE A.6

```

#include <stdio.h>
#include <math.h>

void Calculate_Vector_Sum (double Col_1[], double Col_2[],
double Col_C1_C2[], long Nsize);
double Calculate_Vector_Average (double Column[], long Nsize);

main()
{
    double Col_A[4] = {1, 2, 3, 4};
    double Col_B[4] = {5, 6, 7, 8};
    double Col_C[4];

    long Size = 4;

    double Vector_Average;

    Calculate_Vector_Sum ( Col_A, Col_B, Col_C, Size);

    Vector_Average = Calculate_Vector_Average (Col_C, Size);

    // Send Answers to Screen also Note Index 0 to 3
    printf("Col_C[0] = %lf\n", Col_C[0]);
    printf("Col_C[1] = %lf\n", Col_C[1]);
    printf("Col_C[2] = %lf\n", Col_C[2]);
    printf("Col_C[3] = %lf\n", Col_C[3]);

    printf("Vector Average = %lf\n", Vector_Average);

    return 0;
}

void Calculate_Vector_Sum (double Col_1[], double Col_2[],
double Col_C1_C2[], long Nsize)
{
    int i;
    for(i =0; i<Nsize; ++i)
    {
        Col_C1_C2[i] = Col_1[i] + Col_2[i];
    }
    return;
}

double Calculate_Vector_Average (double Column[], long Nsize)
{
    int i;
    double Sum =0;
    for(i =0; i<Nsize; ++i)
    {
        Sum = Sum +Column[i];
    }

    return Sum/Nsize;
}

```

**FIGURE A.7**

```

#include <stdio.h>
#include <math.h>

void Calculate_Vector_Sum (double* Col_1, double* Col_2, double* Col_C1_C2
, long Nsize);
double Calculate_Vector_Average (double* Column, long Nsize);

main()
{
    double Col_A[4] = {1, 2, 3, 4};
    double Col_B[4] = {5, 6, 7, 8};
    double Col_C[4];

    long Size = 4;

    double Vector_Average;

    Calculate_Vector_Sum ( Col_A, Col_B, Col_C, Size);

    Vector_Average = Calculate_Vector_Average (Col_C, Size);

    // Send Answers to Screen also Note Index 0 to 3
    printf("Col_C[0] = %lf\n", Col_C[0]);
    printf("Col_C[1] = %lf\n", Col_C[1]);
    printf("Col_C[2] = %lf\n", Col_C[2]);
    printf("Col_C[3] = %lf\n", Col_C[3]);

    printf("Vector Average = %lf\n", Vector_Average);

    return 0;
}

void Calculate_Vector_Sum (double* Col_1, double* Col_2, double* Col_C1_C2
, long Nsize)
{
    int i;
    for(i =0; i<Nsize; ++i)
    {
        Col_C1_C2[i] = Col_1[i] + Col_2[i];
    }
    return;
}

double Calculate_Vector_Average (double* Column, long Nsize)
{
    int i;
    double Sum =0;
    for(i =0; i<Nsize; ++i)
    {
        Sum = Sum +Column[i];
    }

    return Sum/Nsize;
}

```

**FIGURE A.8**

```

#include "C_Vector.h"

void Calculate_Vector_Sum (double* Col_1, double* Col_2, double* Col_C1_C2
, long Nsize)
{
    int i;
    for(i =0; i<Nsize; ++i)
    {
        Col_C1_C2[i] = Col_1[i] + Col_2[i];
    }
    return;
}

double Calculate_Vector_Average (double* Column, long Nsize)
{
    int i;
    double Sum =0;
    for(i =0; i<Nsize; ++i)
    {
        Sum = Sum +Column[i];
    }

    return Sum/Nsize;
}

```

**FIGURE A.9**