Problem One:
Given two double precision vectors \( x \) and \( y \) write a function that returns the scalar product in the double precision variable \( w \). 
\[
W = \text{ddot}(n, x, y)
\]
- \( W \) - REAL*8 return variable
- \( N \) - INTEGER number of products to be summed
- \( X \) - REAL*8 array of dimension \( n \)
- \( Y \) - REAL*8 array of dimension \( n \)

Problem Two:
Given a double precision matrix \( A \), two double precision vectors \( x \) and \( y \), and two double precision scalars \( \alpha \) and \( \beta \), write a single subroutine that returns one of the follow double precision vectors replacing the vector \( y \). 
\[
y \leftarrow \alpha Ax + \beta y \text{ or } y \leftarrow \alpha A^T x + \beta y
\]
- \( \text{SUBROUTINE DGEMV} \) ( TRANS, M, N, ALPHA, A, LDA, X, BETA, Y )
  - \( TRANS \) - INTEGER entry, \( TRANS \) specifies the operation to be performed as follows:
    - \( TRANS = 0 \) y := \( \alpha \)A\( x \) + \( \beta \)y.
    - \( TRANS = 1 \) y := \( \alpha \)A\( T x \) + \( \beta \)y.
  - \( M \) - INTEGER. On entry, \( M \) specifies the number of rows of the matrix \( A \).
    M must be at least zero.
  - \( N \) - INTEGER. On entry, \( N \) specifies the number of columns of the matrix \( A \).
  - \( ALPHA \) - REAL*8 On entry, \( ALPHA \) specifies the scalar \( \alpha \).
  - \( A \) - REAL*8 array of DIMENSION (m, n ).
    Before entry, the leading \( m \) by \( n \) part of the array \( A \) must contain the matrix of coefficients.
  - \( LDA \) - INTEGER On entry, \( LDA \) specifies leading dimension of \( A \).
  - \( X \) - REAL*8 Array of DIMENSION \( n \)
  - \( BETA \) - REAL*8 On entry, \( BETA \) specifies the scalar \( \beta \). When \( BETA \) is supplied as zero then \( Y \) need not be set on input.
  - \( Y \) - REAL*8 Array of DIMENSION \( m \) when \( TRANS = 0 \) and \( n \) otherwise.
    On exit, \( Y \) is overwritten by the updated vector \( y \).

Problem Three:
Given a double precision matrices \( A, B, C \), and two double precision scalars \( \alpha \) and \( \beta \), write a single subroutine that returns one of the following results replacing \( C \).
\[
C \leftarrow \alpha AB + \beta C \text{ or } C \leftarrow \alpha A^T B + \beta C \text{ or } C \leftarrow \alpha Ab^T + \beta C \text{ or } C \leftarrow \alpha A^T B^T + \beta C
\]
- \( \text{SUBROUTINE DGEMM} \) ( TRANSA, TRANSB, M, N, K, ALPHA, A, LDA, B, LDB, BETA, C, LDC )
TRANSA - INTEGER
On entry, TRANSA specifies the form of op( A ) to be used in
the matrix multiplication as follows:
   TRANSA = 0,   A.
   TRANSA = 1,   Aᵀ

TRANSB - INTEGER
On entry, TRANSB specifies the form of op( B ) to be used in
the matrix multiplication as follows:
   TRANSB = 0,   B.
   TRANSB = 1,   Bᵀ

M - INTEGER On entry, M specifies the number of rows of the matrix
   op(A) and of the matrix C.
N - INTEGER On entry, N specifies the number of columns of the matrix
   op(B) and the number of columns of the matrix C.
K - INTEGER. On entry, K specifies the number of columns of the matrix
   op(A) and the number of rows of the matrix op(B).

ALPHA - REAL*8 On entry, ALPHA specifies the scalar alpha.
   Unchanged on exit.

A - REAL*8 array
LDA - INTEGER leading dimension of A.

B - REAL*8 Array
LDB - INTEGER leading dimension of B

BETA - REAL*8 On entry, BETA specifies the scalar beta.

C - REAL*8 Array
LDC - INTEGER leading dimension of C