Ivana Zdravevska

// define structure

1. set structure called Matrix
   a11 <double>
   a12 <double>
   a21 <double>
   a22 <double>
   END Matrix

// define function prototypes

2. set function prototype **file_open**(arguments:)
3. set function prototype **file_close**(arguments)
4. set function prototype **print_matrix**(arguments)
5. set function prototype **get_matrix**(arguments)
6. set function prototype **get_scalar**(arguments)
7. set function prototype **calc_sum**(arguments)
8. set function prototype **calc_diff**(arguments)
9. set function prototype **scalar_mult**(arguments)
10. set function prototype **calc_prod**(arguments)
11. set function prototype **calc_inv**arguments)

//main

9. set two variables of matrix type (matrix1, matrix2).
10. Initialize an integer variable choice to zero. (choice for menu)
11. Set a char variable. (answer for another calculation)
12. call display_menu function.
13. Do
    ask the user for one of the above calculations using integer numbers and store the result in choice.
    While choice is less than one or choice is greater than six.
        display an error message.
        prompt the user for a valid option.
    END while.
    Switch statement.
        set case 1:
            call **get_values** function (arguments: vector_1, vector_2, choice).
            call **calc_add** function (arguments: vector_1, vector_2).
        set case 2:
            call **get_values** function (arguments: vector_1, vector_2, choice).
            call **cal_sub**(arguments: vector_1, vector_2).

set case 3:
call **get_values** function (arguments: vector_1, vector_2, choice).
call **scalar_mul**(arguments: vector_1, k).
set case 4:
call **get_values** function (arguments: vector_1, vector_2, choice).
call **scalar_prod**(arguments: vector_1, vector_2).
set case 5:
call **get_values** function (arguments: vector_1, vector_2, choice).
call **norm**(arguments: vector_1).
set case 6 to end the program.
END switch
Ask the user to perform another calculation
While answer is not Y, y, N, n
Display error message
Get input
END While
clear the screen.
while answer equals = y or Y.
END Do.

//function declarations

14. function **display_menu**(parameters: none)
    display menu
    End Function
15. function **get_values**(parameters: Vector v1& , Vector v2&, integer choice&)
    if choice equals 1 or choice equals 2 or choice equals or 4
    get first element of v1.
    get second element of v1.
    get first element of v2.
    get second element of v2.
    else if choice equals 3
    get first element of v1.
    get second element of v1.
    get scalar.
    else
    get first element of v1.
    get second element of v1.
    End Function.
16. function **cal_add**(parameters: Vector v1, Vector v2)
    add first element of v1 and v2.
    add second element of v1 and v2.
    display result.

End function.

17. function **cal_sub**(parameters: Vector v1, Vector v2)
    subtract first element of v1 and v2.
    subtract second element of v1 and v2.
    display result.
    End function.
18. function **scalar_mul**(parameters: Vector v1, integer k)
    multiply k by each element of v1.
    display result
    End function.
19. function **scalar_prod**(parameters: Vector v1, Vector v2)
    multiply the first element of v1 by the first element of v2.
    multiply the second element of v1 by the second element of v2.
    display result.
    End function.
20. function **norm**(parameters: Vector v1)
    multiply the first element of v1 by itself.
    multiply the second element of v2 by itself.
    add them
    take the square root of the addition
    display result
    End function.