1. Vector <structure>
2. x <double>
3. y <double>
4. k <double>
5. name <string>
6. End
7. main
8. start
9. firstvector is Vector Var  
   secondvector is Vector Var
10. scalar is Vector Var
11. outputvector is Vector Var
12. Initialize a variable to get a char
13. initialize a double variable for magnitude
14. initialize a double variable for scalar product
15. initialize an int variable for choice
16. display ADT
17. Call inputfunc (arguments: pointer to firstvector, pointer to secondvector, pointer to scalar)
18. do
19. display choose your option
20. display 1 for addition
21. display 2 for subtraction
22. display 3 for scalar multiple
23. display 4 for scalar product
24. display 5 for magnitude
25. do
26. prompt user to enter his choice
27. if( choice<0 && choice >5)
28. start
29. display wrong entry please renter your choice 1-5
30. prompt the user to enter his choice
31. while(choice<0 && choice >5)
32. switch (choice)
33. case 1:
34. start
35. Call additonfunc(arguments: pointer to firstvector, pointer to secondvector, pointer to outputvector)
36. Call outputfunc(arguments: pointer to outputvector, pointer to scalar)
37. end
38. case 2:
39. start
40. Call subfunc (arguments: pointer to firstvector, pointer to secondvector, pointer to outputvector)
41. Call outputfunc(arguments: pointer to outputvector, pointer to scalar)
42. end
43. case 3:
44. start
45. Call scalarmultifunc(arguments: pointer to firstvector, pointer to secondvector, pointer to outputvector, pointer to scalar)
46. Call outputfunc(arguments: pointer to outputvector, pointer to scalar)
47. end
48. case 4:
49. start
50. scalar product = Call scalarproductfunc(Arguments: pointer to Firstvector, pointer to secondvector)
51. display firstvector.name
52. display (firstvector.x, firstvector.y)
53. display secondvector.name
54. display (secondvector.y secondvector.y)
55. display the scalar is: display scalar
56. display the scalar product is: display scalar product
57. end
58. case 5:
59. start
60. Call magnitudefunc(Arguments: pointer to firstvector, pointer to secondvector, pointer to magnitude)
61. display firstvector.name
62. display (firstvector.x, firstvector.y)
63. display secondvector.name
64. display (secondvector.y secondvector.y)
65. display the scalar is: display scalar
66. display the magnitude is: display magnitude
67. end
68. display do you want to repeat? (enter y for yes)
69. while(repeat==y)
70. end
71. inputfunc (parameters: pointer to firstvector, pointer to secondvector, pointer to scalar)
72. start
73. display enter the name for the first vector
74. prompt the user to enter the first vectors name
75. display enter the first vectors numbers
76. prompt the user to enter the first number
77. prompt the user to enter the second number
78. display enter the name for the second vector
79. prompt the user to enter the second vectors name
80. display enter the second vectors numbers
81. prompt the user to enter the first number
82. prompt the user to enter the second number
83. display enter the scalar
84. prompt the user to enter the scalar
85. end

1. additionfunc (parameters: pointer to firstvector, pointer to secondvector, pointer to outputvector)
2. start
3. outputvector.x= firstvector.x+secondvector.y
4. outputvector.y= firstvrctor.x+secondvector.y
5. end
6. subfunc (parameters: pointer to firstvector, pointer to secondvector, pointer to outputvector)
7. start
8. outputvector.x= firstvector.x-secondvector.y
9. outputvector.y= firstvrctor.x-secondvector.y
10. end
11. scalarmultifunc (parameters: pointer to firstvector, pointer to secondvector, pointer to outputvector, scalar)
12. start
13. initialize an int variable for choice
14. display choose which vector you want to work with
15. prompt the user for his choice
16. switch(choice)
17. case 1:
18. start
19. outputvector.x= firstvector.x\*k
20. outputvector.y= firstvrctor.x\*k
21. end
22. case 2:
23. start
24. outputvector.x= secondvector.x\*k
25. outputvector.y= secondvrctor.x\*k
26. end
27. end
28. scalarproductfunc (parameters: pointer to firstvector, pointer to secondvector)
29. start
30. initialize a double variable for scalar product
31. scalar product = (firstvector.x\*secondvector.x)+(secondvector.y\*secondvector.y)
32. return scalar product
33. end
34. magnitudefunc(parameters: pointer to firstvector, pointer to secondvector, pointer to magnitude)
35. start
36. initialize a double variable for magnitude result
37. initialize an int variable for choice
38. display choose which vector you want to work with
39. prompt the user for his choice
40. switch(choice)
41. case 1:
42. start
43. magnitude = ((firstvector.x\*firstvector.x)+(firstvector.y\*firstvector.y))
44. magnitude = sqrt(magnitude)
45. if (magnitude result < 0)
46. start
47. magnitude = magnitude \*(-1)
48. end
49. end
50. case 2:
51. start
52. magnitude = ((secondvector.x\*secondvector.x)+(secondvector.y\*secondvector.y))
53. magnitude = sqrt(magnitude)
54. if (magnitude result < 0)
55. start
56. magnitude = magnitude \*(-1)
57. end
58. end
59. return magnitude
60. end
61. outputfunc(parameter: pointer to outputvector, pointer to scalar)
62. start
63. display firstvector.name
64. display (firstvector.x, firstvector.y)
65. display secondvector.name
66. display (secondvector.y secondvector.y)
67. display the scalar is display scalar
68. display result vector
69. display (outputvector.x, outputvector.y)
70. end