HW_8ab - (2 parts)

HW_8a - **Do the Math** - Array of doubles - Pass an array to a function
- for loop - switch - math operations on the array

Write a program that produces the output shown, based on the following information.

```
/* OUTPUT
Enter 5 numbers:
1.0
2.22
3.3
4.0
5.5
```
← **getValues()** function (User enters 5 numbers, which are read into array)

```
// --- SCREEN CLEARS ----------------------------------------

A. Display the Square Root of each number.
B. Display the Sum of all numbers
C. Display the Average of all numbers
```
showMenu() function

```
Enter your choice:  B
```
← **getChoice()** function (User enters a B)

```
// --- SCREEN CLEARS ----------------------------------------

The sum is 16.02.
```
← **doTheMath()** function

```
Press any key to continue ... /*
```

Include the following:

- This program uses an array to hold 5 numbers (double data type).
- The program asks the user to enter 5 numbers, which are read and assigned to an array.
- Then a menu of choices is displayed (see output).
- Then the user is asked to enter a choice (A, B, or C).
  - o Depending on the choice, a math operation is performed on all of the values in the array.
- Finally, the results are displayed (see output).

1.) In main(), declare an array that can hold up to 5 double values.

2.) Name the array: ***numbers***

3.) The program calls a function named: **getValues()**

- The array is passed to the function.
- The function is void-returning.

  Remember: Arrays and struct objects are always passed by reference.

- The function prompts the user to enter 5 integer values.
- The numbers are assigned to elements in the array.

Note: Because the array is passed by reference, the values in the array can be accessed in main().

4.) The program calls a function named: **showMenu()**

- The function simply displays a menu of choices: A, B, and C. (see output)
- No arguments are passed to the function.
- No value is returned to main()

5.) The program calls a function named: **getChoice()**

- The function prompts the user to enter a choice (A, B, or C).
- No arguments are passed to the function.
- The user's choice is returned to main().

6.) The program calls a function named: **doTheMath()**

- The function uses a switch control structure to determine which math operation to perform on the values in the array.
- The function does a math operation based on the user's choice and outputs the results to the screen.
- The function uses a ***for loop*** to access all of the array elements.
  - With each iteration, a math operation is performed on an element's value.
  - In the case of B, each value that is read is added to a variable called ***sum***.

    Format the sum to 2 places to the right of the decimal. (see output above)
  - In the case where the user enters A, the output looks like this:
    - Format the output to 3 places to the left of the decimal point. (see output)
    - Also indent and align the numbers right. (see output)

```
/* Output for choice A
The square roots are:
     1.000
     1.490
     1.414
     1.732
     2.000
Press any key to continue ../*
```

```
/* Output for choice C
The average is:  3.204
Press any key to continue ../*
```

HW_8b - Display array elements in reverse order.

- Create a new project and name it: **Reverse_order**
- Create a new file and name it: **HW_8b-1.cpp**

Write a program that does the following:

This program uses an array to hold 5 integers. The program asks the user to enter 5 ages, which are read and assigned to an array. The results can be displayed in order or in reverse order, depending on the user's choice (see output).

In main(), declare an array that can hold up to 5 integer values - Name the array: *ages*

Include 4 functions:

getAges - This function prompts the user to enter 5 ages, which are read and assigned to the *ages* array.

- The ages array is passed by reference to the function.
- The function is void-returning.

getChoice - This function prompts the user to enter a choice of how the array values should be displayed.

- No arguments are passed to the function.
- The function returns a value representing the user's choice.

Note: Only one of the following two functions will get called – depending on the user's choice.

- An if-else statement is used in main() to determine which function is called.
- If the user enters O, the array is displayed in order - **displayInOrder()** is called.
- If the user enters R, the array is displayed in reverse order - **displayInReverse()** " "
- If the user enters a value other than O or R, this message is displayed:

```
Invalid entry  -  Must be O or R!
```

---

displayInOrder - This function displays the array values in order, with element[0] first.

- The ages array is passed by reference to the function.
- The function is void-returning.

displayInReverse - This function displays the values in reverse order, with element[4] first.

- The ages array is passed by reference to the function.
- The function is void-returning.

/* OUTPUT

```
Enter 5 ages:

21
22
19
27
29
```

getAges() function

(user enters O)

```
How do you want to see the ages displayed?
Enter O for In Order, or R for In Reverse (R)   O
```

getChoice() function

```
Here are the ages in order:

21   22   19   27   29
```

displayInOrder() function

```
Run program again (y / n)?   y
```

← User enters y

```
----- (screen clears) ---------------------------------

Enter 5 ages:

33
18
17
25
24
```

getAges() function

(user enters R)

```
How do you want to see the ages displayed?
Enter O for In Order, or R for In Reverse (R)   R
```

getChoice() function

```
Here are the ages in reverse order:

24   25   17   18   33
```

displayInOrder() function

```
Run program again (y / n)?   n
```

← User enters n