Experiment 4

1. We have two arrays, A and B, each of 10 integers. Write a program that tests if every element of array A is equal to its corresponding element in array B. In other words, the function must check if A [0] is equal to B[0], A[1] is equal to B[1], and so forth.
2. Write a function that reverses the elements of an array so that the last element becomes the first, the second from the last becomes second, and so forth. The function is to reverse the elements in place—that is, without using another array.
3. Write a program that sorts a 50-element array using the selection sort, the bubble sort, and the insertion sort. Each sort is to be executed twice.
   a) For the first sort, fill the array with random numbers between 1 and 1,000.
   b) For the second sort, create a nearly ordered list by exchanging every 10^{th} element with its predecessor (9 and 10, 19 and 20, etc.).
   c) Each sort is to sort the same data. For each sort, count the number of comparisons necessary to order the list.
   d) After each sort execution, print the unsorted data followed by the sort data in 5×10 matrixes. After the sorted data, print the number of comparisons and the number of moves required to order the data. Provide appropriate headings for each printout.
   e) To make sure your statistics are as accurate as possible, you must analyze each loop limit condition test and each selection statement in your sort functions.
   f) Analyze the heuristics you generated and write a few lines concerning what you discovered about these sorts.

Experiment-5

1. Write a function using pointers that given the time in seconds passes back the time in hours, minutes, seconds and a character indicating A.M. or P.M. If the number of seconds is more than 24 hours, the function is to return false as an error indicator.
2. Write a function that reverses the elements of an array so that the last element becomes the first, the second from the last becomes second, and so forth. The function must accept only one pointer value and return void.
3. Given the following declaration and definition
   int num[20];
   and using only pointer notation, write a for loop to read integer values from the keyboard to fill the array.
4. Write a program that will read 10 integers from the keyboard and place them in array. The program then will sort the array into ascending order and descending order and print the sorted lists. The program must not change the original array or create any other integer arrays.

The solution to this problem requires two pointer arrays. The first pointer array is rearranged so that it points to the data in ascending sequence. The second pointer array is rearranged to that it points to the data in descending sequence.