Name:- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Answer the following Questions:

**Q1: [5 Points]** Write a function that sorts a linked list. (Use any sort algorithm, make sure to state which one you’re going to use). Provide the results of running your project.

|  |
| --- |
| Use this space to answer this question |

**Q2: [5 Points]** for the following array:

30 5 40 11 20 9 15 2 60 25 80 3 73 35 4 75 20 6

Sort the array using the Sell sort algorithm, start with a gap equal to 7. Show the details of the process and No Code is required.

|  |
| --- |
| Use this space to answer this question |

**Q3: [5 Points]** Trace the execution of the call mystery(10) for the following recursive function. What does this function do?

int UMKC (int n) {

if (n == 0)

return 0;

else

return n \* n + UMKC (n – 2);

}

Use a stack (table) to trace the execution and show the details of the recursion process

|  |
| --- |
| Use this space to answer this question  n=10  return (10) \* (10) + UMKC (10 -2)  n = 8  return (8) \* (8) + UMKC (8 – 2)  n = 6  return (6) \* (6) + UMKC (6 – 2)  n=4  return (4) \* (4) + UMKC (4 – 2)  n=2  return (2) \* (2) + UMKC (2 – 2)  n=0  return 0  return 2 \* 2 + 0 = 4  return 4 \* 4 + 4 = 20  return 6 \* 6 + 20 = 56  return 8 \* 8 + 56 = 120  return 10 \* 10 + 110 = 220  final result of the function is 220 |

**Q4: [10 Points]**

1. Use the random number generator to generate random integer values and use them to populate 3 arrays of type integer where each array will have 1000 item. (3 arrays of size 1000 each and populate them with random values, the arrays should not be identical)
2. Modify the bubble sort algorithm by adding a counter to count how many times the elements of the array changed value (swapping is considered as a one change)
3. For each array, apply the modified bubble sort algorithms and save the counter value.
4. Repeat step 1 to 4, using shell sort algorithm, quick sort algorithm, merge sort algorithm and selection sort algorithm.

Use the results to complete the table below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1st array counter | 2nd array counter | 3rd array counter |
| Bubble sort |  |  |  |
| Shell sort |  |  |  |
| Quick sort |  |  |  |
| Merge sort |  |  |  |
| Selection sort |  |  |  |

State your observations and conclusion in the space provided below: provide a screen shot for the results of running your program for each algorithm.

|  |
| --- |
| Use this space to answer this question |