Programming Assignment

No. 4

Using a loop and indexed addressing, write assembly codes that rotate the members of a 32-bit integer array forward one position. The value at the end of the array must wrap around to the first position. For example, the array [10,20,30,40] would be transformed into [40,10,20,30].

No. 5

Write an assembly program to compute the sum of ***arrayA*** and the sum of ***arrayB,*** shown below. Use loops and procedures to computes these sums.

```
arrayA word 10h, 20h, 30h,40h,50h
arrayB dword 10h, 20h, 30h, 40h,50h
```

Print ***sumA*** and ***sumB*** before and after exchanging two sums.

No. 6

Write an assembly program to generate 30 random integers between 0 and 990. Print the random numbers diagonally on a light-green screen.

Programming Assignments

No. 1

Write an assembly program to compute the sum of LIST1 and the sum of LIST2. LIST1 has three 16-bit hex and LIST2 has four 16-bit hex numbers, shown below. Store the result in SUM1 and SUM2 and then exchange these two values.

LIST1: 10h, 20h, 30h
LIST2: 10h, 20h, 30h, 40h

Print the contents of SUM1 and SUM2 before and after you exchange the values.

No. 2

Use a loop with indirect or indexed addressing to reverse the elements of an integer array in place. Do not copy the elements to any other array. Use the SIZEOF, TYPE, and LENGTHOF operators to make the program as flexible as possible if the array size and type should be changed in the future. Display the modified array by calling the DumpMem. Add adequate documentations.

No. 3

Write an assembly program to generate and display the first 24 Fibonacci numbers, beginning with 1 and ending with 46368. Add adequate documentations.