Wentworth Institute of Technology
Department of Computer Science and Networking
Assembly Language
Lab 7 – Chapter 6

Objective:

Coding with conditional jumps.

Procedure:

1. Summing Array Elements in a Range

a) Create a procedure that outputs the values of a signed DWORD array in one line. The procedure receives the pointer to the array in EDI, the number of elements in ECX.
b) Create a procedure that computes the sum of all array elements falling within the range *j . . k* inclusive. NOTE: it means that the *values* of the array elements should be within the range. The procedure receives the pointer to the array in EDI, the number of elements in ECX, and the range values in EBX and EDX. The sum is returned in EAX.
c) Write a test program that calls the pair of the procedures twice. Before each set of calls, initialize the array with random values between -50 and +50. After the array is output, ask user for the values of j and k. After the sum is computed, output its value.

   Make sure you pass the offset and the length to the registers only once, before the first call; only new values of *j* and *k* need to be passed.

An example: if the values in the array are 4, 8, -3, 6, 5, 0, 9, 1, 1 and *j* = -1, *k* = 6, then the sum is 4 + 5 +6 + 0 + 1 + 1 =17

Disclosure: the exercise is taken from the Kip Irvine textbook, 7th edition and updated with additional requirements.

Bonus (10 points)
Create another procedure which computes the sum of absolute values of the elements within the specified range, and tests it in the main program the same way as in the first two series.