HOMEWORK 3

Directions: *This assignment is due by **11:00 a.m. on Wednesday, October 8**.*
*Submit your solution as an .asm file in Canvas. This assignment is worth **30 points**.*

Background: Prefix Sum

Given a sequence $x_1, x_2, ..., x_n$ of numbers, the *prefix sum* is a new sequence of numbers $y_1, y_2, ..., y_n$ where

$y_1 = x_1$
$y_2 = x_1 + x_2$
$y_3 = x_1 + x_2 + x_3$
$y_4 = x_1 + x_2 + x_3 + x_4$
$y_5 = x_1 + x_2 + x_3 + x_4 + x_5$
etc.

As an example, to find the prefix sum of [1, 10, 200, 500], compute

$y_1 = 1$,
$y_2 = 1 + 10$,
$y_3 = 1 + 10 + 200$, and
$y_4 = 1 + 10 + 200 + 500$,

so the prefix sum of [1, 10, 200, 500] is [1, 11, 211, 711]. In other words, to compute the i^{th} element of the output array, sum the i^{th} element of the input array with all the preceding elements of the input array.

As further examples, the prefix sum of [–2, 4, 0] is [–2, 2, 2]. The prefix sum of [1, 2, 3, 4] is [1, 3, 6, 10].

Assignment

- Download **HW3.asm** from Canvas. Note the declaration of the SWORD array named `values`.
- In the TITLE line, replace TODO: YOUR NAME with your name.
- Fill in the main procedure to behave as follows. Comment your code. Solutions will be graded based on both correctness and readability.

1. Repeat the following 10 times:
   a. Display `Enter a value:`
   b. Read a *signed* integer from the input (`call ReadInt`; it will be stored in EAX).
   c. If the integer is less than –32768 or greater than 32767, display "Out of range" and exit immediately.
   d. At this point, you know the integer is in the range [–32768, 32767], so you can represent the integer value with only 16 bits. Store the integer *as a signed, 16-bit value* in the SWORD array `values`. *(Hint: You have a 32-bit value in EAX. Which 16 bits of this register contain the 16-bit representation you're interested in, and how can you access them?)*

   After Step 1 has completed, `values` should contain the 10 integer values entered by the user.

2. Display the 10 values *in reverse order*, on one line, separated by spaces. *Do not reverse the array in memory;* just display it in reverse order.

3. Display the prefix sum of the `values` array, on one line, with values separated by spaces. *Use 32-bit values* when computing the prefix sum, so it will not overflow.

Two example runs are given on the next page.

Example 1:

```
Enter a value: 15
Enter a value: 30
Enter a value: 40000
Out of range
```

Example 2:

```
Enter a value: 1
Enter a value: 1
Enter a value: 1
Enter a value: 1
Enter a value: 1
Enter a value: 1
Enter a value: 1
Enter a value: 1
Enter a value: 1
Enter a value: 10
+10 +1 +1 +1 +1 +1 +1 +1 +1 +1
+1 +2 +3 +4 +5 +6 +7 +8 +9 +19
```

Example 3:

```
Enter a value: 15
Enter a value: -30
Enter a value: -1
Enter a value: 0
Enter a value: 1000
Enter a value: -9
Enter a value: 32000
Enter a value: 31000
Enter a value: 30000
Enter a value: 30000
+30000 +30000 +31000 +32000 -9 +1000 +0 -1 -30 +15
+15 -15 -16 -16 +984 +975 +32975 +63975 +93975 +123975
```