Write an assembly language program that links the generation of triangular numbers with the creation of an arithmetic sequence. The program will first generate a sequence of triangular numbers using macros, then generate an arithmetic sequence using a procedure, and finally, modify the arithmetic sequence by reversing it with the help of a stack while adding an additional layer of complexity.

1. **Triangular Numbers Generation:**

   Write a macro to generate a sequence of triangular numbers. Triangular numbers are calculated as T_n = n * (n + 1) / 2, where n is the position in the sequence starting from 1. The macro should generate and print the first N triangular numbers.

   Input: 5

   Output: 1, 3, 6, 10, 15

2. **Arithmetic Sequence Generation:**

   Implement a procedure to generate an arithmetic sequence up to a given limit. The sequence should start with a specified first term and have a defined common difference. The procedure will generate the sequence and store it in memory.

   Input: First term = 2, Common difference = 3, Limit = 20

   Output: [2, 5, 8, 11, 14, 17]

3. **Prime Number Reversal of the Sequence:**

   Modify the previous arithmetic sequence generation by implementing a procedure that first identifies and pushes only prime numbers from the sequence onto a stack. Once the prime numbers are on the stack, pop them off to reverse the order, then print the reversed sequence of prime numbers.

   Input: [2, 5, 8, 11, 14, 17]

   Output: [17, 11, 5, 2]