**8. String to Integer (atoi)**

Implement atoi which converts a string to an integer.

The function first discards as many whitespace characters as necessary until the first non-whitespace character is found. Then, starting from this character, takes an optional initial plus or minus sign followed by as many numerical digits as possible, and interprets them as a numerical value.

The string can contain additional characters after those that form the integral number, which are ignored and have no effect on the behavior of this function.

If the first sequence of non-whitespace characters in str is not a valid integral number, or if no such sequence exists because either str is empty or it contains only whitespace characters, no conversion is performed.

If no valid conversion could be performed, a zero value is returned.

**Note:**

* Only the space character ' ' is considered as whitespace character.
* Assume we are dealing with an environment which could only store integers within the 32-bit signed integer range: [−231,  231− 1]. If the numerical value is out of the range of representable values, INT\_MAX (231− 1) or INT\_MIN (−231) is returned.

**Example 1:**

**Input:** "42"

**Output:** 42

**Example 2:**

**Input:** " -42"

**Output:** -42

**Explanation:** The first non-whitespace character is '-', which is the minus sign.

  Then take as many numerical digits as possible, which gets 42.

**Example 3:**

**Input:** "4193 with words"

**Output:** 4193

**Explanation:** Conversion stops at digit '3' as the next character is not a numerical digit.

**Example 4:**

**Input:** "words and 987"

**Output:** 0

**Explanation:** The first non-whitespace character is 'w', which is not a numerical

  digit or a +/- sign. Therefore no valid conversion could be performed.

**Example 5:**

**Input:** "-91283472332"

**Output:** -2147483648

**Explanation:** The number "-91283472332" is out of the range of a 32-bit signed integer.

  Thefore INT\_MIN (−231) is returned.

**151. Reverse Words in a String**

Given an input string, reverse the string word by word.

**Example:**

**Input:** "the sky is blue",

**Output:**"blue is sky the".

**Note:**

* A word is defined as a sequence of non-space characters.
* Input string may contain leading or trailing spaces. However, your reversed string should not contain leading or trailing spaces.
* You need to reduce multiple spaces between two words to a single space in the reversed string.

**Follow up:**For C programmers, try to solve it *in-place* in *O*(1) space.

**29. Divide Two Integers**

Given two integers dividend and divisor, divide two integers without using multiplication, division and mod operator.

Return the quotient after dividing dividend by divisor.

The integer division should truncate toward zero.

**Example 1:**

**Input:** dividend = 10, divisor = 3

**Output:** 3

**Example 2:**

**Input:** dividend = 7, divisor = -3

**Output:** -2

**Note:**

* Both dividend and divisor will be 32-bit signed integers.
* The divisor will never be 0.
* Assume we are dealing with an environment which could only store integers within the 32-bit signed integer range: [−231,  231 − 1]. For the purpose of this problem, assume that your function returns 231 − 1 when the division result overflows.