Problem 13. Work out the first ten digits of the sum of the following one-hundred 50-digit numbers.

37107287533902102798797998220837590246510135740250
46376937677490009712648124896970078050417018260538
74324986199524741059474233309513058123726617309629
91942213363574161572522430563301811072406154908250
23067588207539346171171980310421047513778063246676
89261670696623633820136378418383684178734361726757
28112879812849979408065481931592621691275889832738
44274228917432520321923589422876796487670272189318
47451445736001306439091167216856844588711603153276
...

Knowledge required: How to add numbers taught in elementary school, How to read a file.

Solution Outline: We start by implementing function called `adjust` which takes in two numbers in string format and returns the two strings which are both adjusted to same size by adding 0's at the end of the smaller string (i.e `len(str_1) == len(str_2)`).

Then we implement a function that does majority of the job the `add` function which adds two numbers in string format and returns the result in string format. The algorithm is similar to the way we used to add numbers in elementary school. Some programming languages provide builtin support for such big numbers, but this defeats our purpose as we have not learnt anything other than, how to use such a library.

Python Solutions

BuiltIn BigInt support

```python
final_result = 0
with open('nums.txt', 'r') as f:
    for num in f.readlines():
        final_result += int(num)

final_result = str(final_result)
print(final_result[:10])
```

Elementary school Addition

```python
def adjust(numA, numB):
    if len(numA) > len(numB):
        numA, numB = numB, numA

    diff = len(numB) - len(numA)
    numA += '0' * diff

    return numA, numB


def add(numA, numB):
    # reverse them as we normally add from right to left
    numA = numA[::-1]
    numB = numB[::-1]

    # modify them so that their lengths are equal
    numA, numB = adjust(numA, numB)

    result = ''
    carry = 0
    N = len(numA)
    for i in range(N):
        curr = int(numA[i]) + int(numB[i]) + carry
        dig = curr % 10
        carry = curr // 10
        result += str(dig)

    if carry:
        result += str(carry)

    # reverse of result will be the answer
    return result[::-1]

final_result = ''

# read the numbers stored in nums.txt
with open('nums.txt', 'r') as f:
    for num in f.readlines():
        num = num.strip()
        final_result = add(final_result, num)

print(final_result[:10])
```