Programming Practice

2018-12-06

Week 14

Notice

Notice

- Project 3rd Week submission due date has been extended.
  →12/18(Tue) 14:00

- No further delay allowed after 12/18(Tue) 14:00.

Notice – Final Exam

- The Final exam is on 12/16(SUN) 13:00.

Practice Lecture

Big Integer

- **Big integer** is used for mathematical operation which involves very **big integer** calculations that are outside the limit of all available primitive data types(int, long long etc.).

- Ex) 19573293052938594982738592839102938573293829502,

295739482908593028928573920185763958482910536930293739204
76939285039139582 7,

684029503920583920395848483920192576739203982938493029375
948398271615285986709356281928358392830932934275930135720
579306096750000126582737772835192835638273592837592837592

Homework Problems

1. Big Integer(Multiplication)

Problem. **1**

Big Integer(Multiplication)

Description

Calculate multiplication of two big integers X and Y.

X and Y are 0 or positive integers up to 100 digits.

Input

First line contains a single big integer X.

Second line contains a single big integer Y.

Output

Print the result of X * Y.

Sample

[input]

```
103958275843
9275391759692
```

[output]

```
964253735134767338784035 6
```

[input]

```
9876543212345678 9
123456789
```

[output]

```
1219326311415942675019052 1
```

Problem. **1**

Big Integer(Multiplication)

Sample

[input]

```
39572934540193529304384759302019282938492830 59
```

```
95928374610192837409876543211234568923852810
```

[output]

```
37961672889963241232069788195682476578968907926104592006600368240403072276874
9382542545790
```

[input]

```
10000000000000000000000000000000000000000000000000000000000000000000000000000
```

```
99999999998888888887777777776666666666555555555554444444444433333333332222
222221111111111
```

[output]

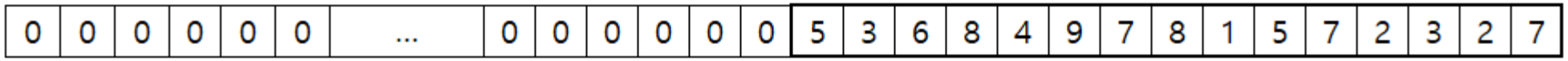
```
99999999998888888887777777776666666666555555555554444444444433333333332222
22222211111111100000000000000000000000000000000000000000000000000000000000000
```

Big Integer

- You can manage these 'big' numbers with integer array. Separate the 'big' number into single digits, and save each digit in each element.

- Since problem description specifies maximum 100 digits,
  use an int array of length 100, and save to the right-end of the array for convenience.

- Ex) When value is: `536,849,781,572,327`

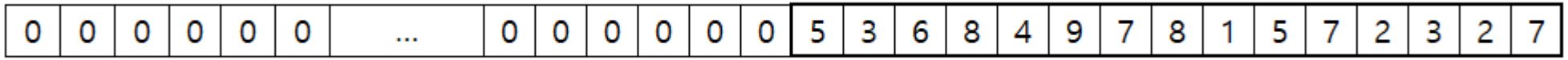


Tip) Saving to the right-end of int array

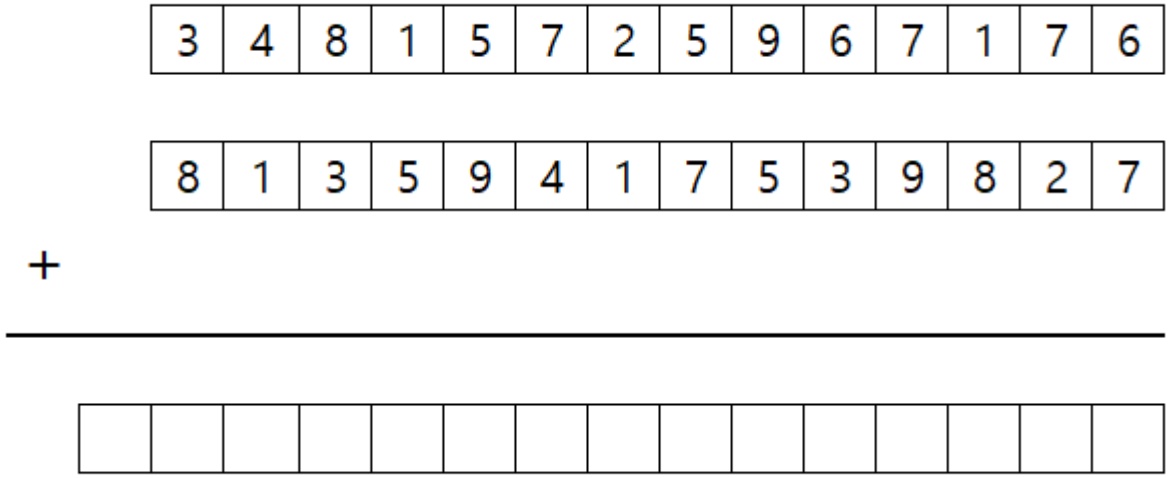
- First, get whole input as a string with %s. (Since problem description specifies maximum 100 digits, use a char array of length 100+).

- Use `strlen` to find out how many digits are in the big integer.

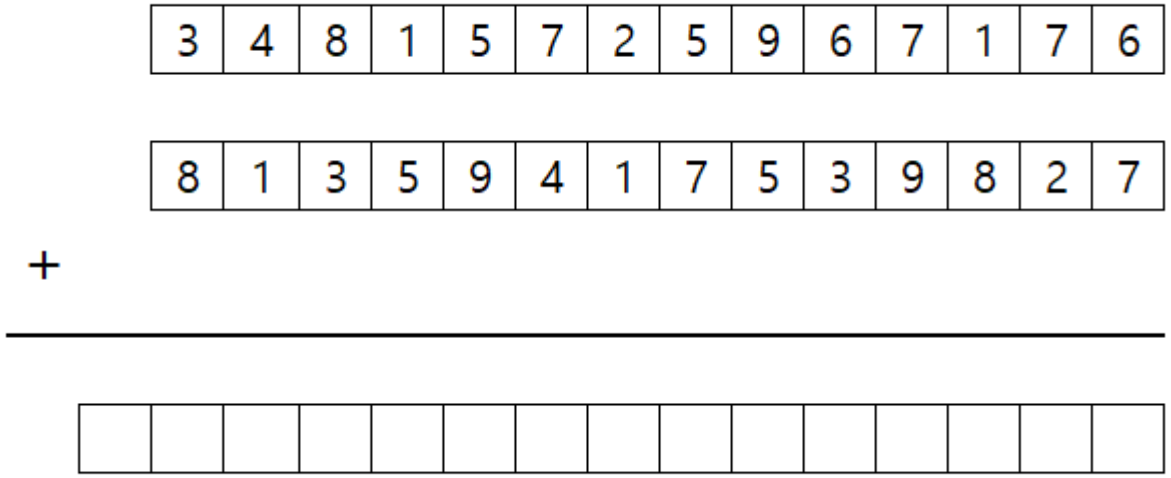
- Start from the right-end of the string (*i.e.* total index - 1).
  Parse each character to integer, and save each digit as a single element of an int array (of length 100).



Big Integer (addition)

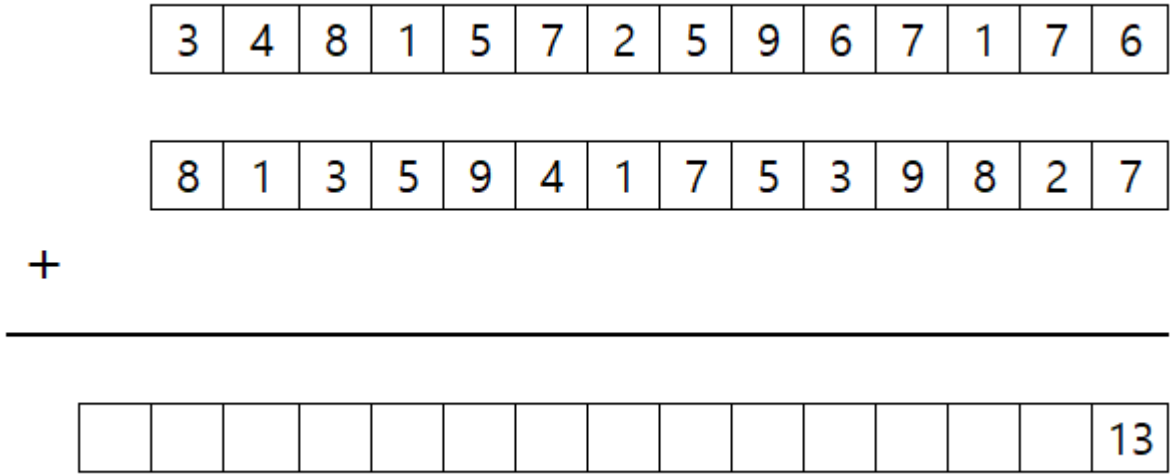


Big Integer (addition)



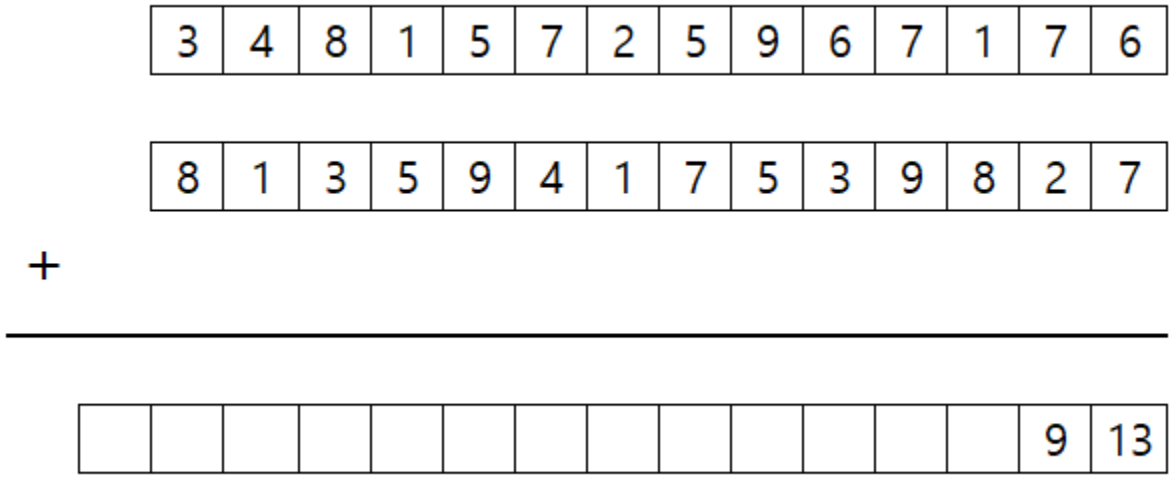
Prepare a sum array (with big enough size)

Big Integer (addition)



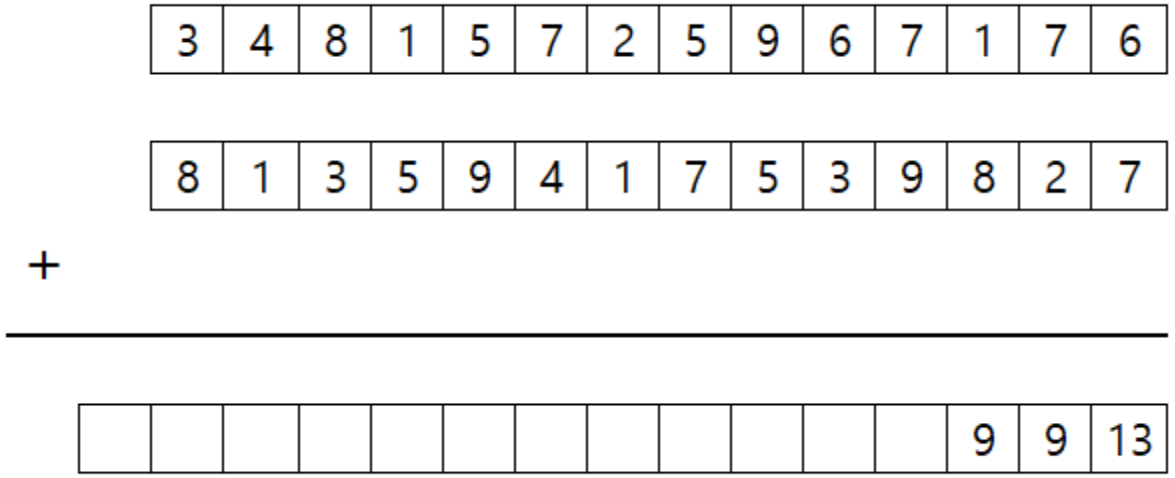
Add each decimal place, one by one.

Big Integer (addition)



Add each decimal place, one by one.

Big Integer (addition)



Add each decimal place, one by one.

Big Integer (addition)

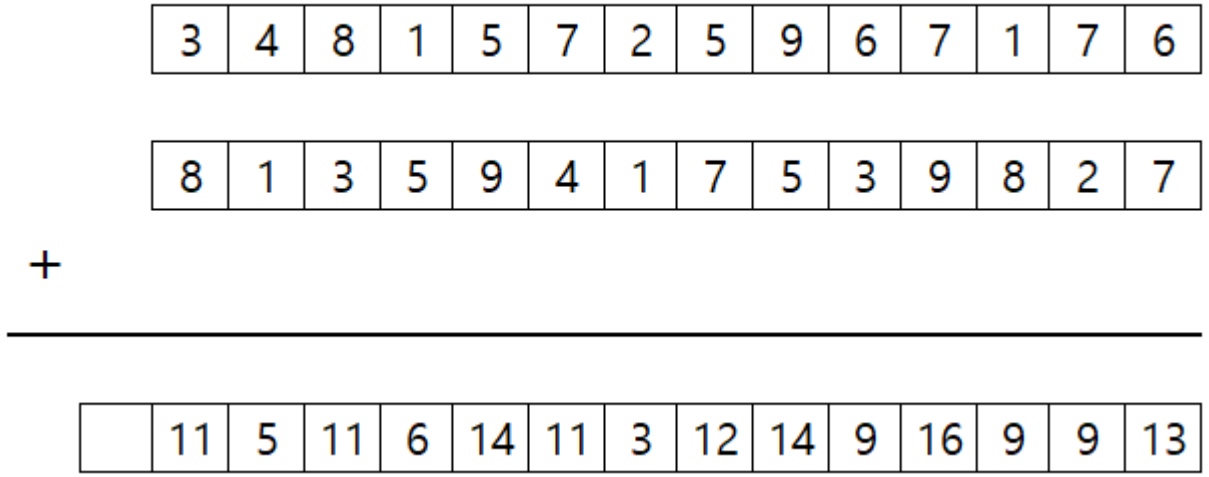
	3	4	8	1	5	7	2	5	9	6	7	1	7	6
	8	1	3	5	9	4	1	7	5	3	9	8	2	7
+														
		11	5	11	6	14	11	3	12	14	9	16	9	9	13

Add each decimal place, one by one.

Big Integer (addition)

	3	4	8	1	5	7	2	5	9	6	7	1	7	6
	8	1	3	5	9	4	1	7	5	3	9	8	2	7
+														
		11	5	11	6	14	11	3	12	14	9	16	9	9	13

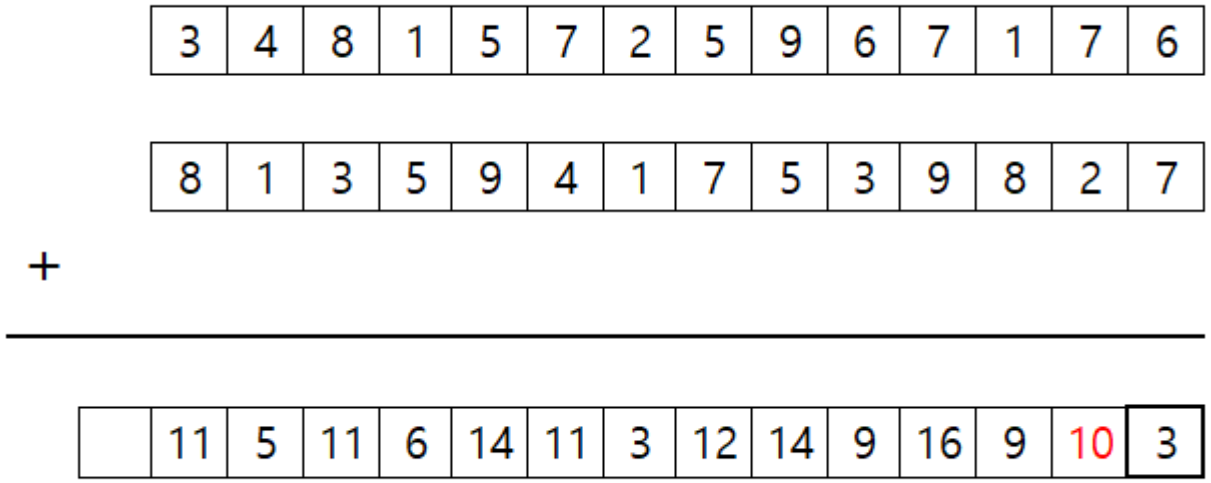
Big Integer (addition)



For each `s[i]` *(right-to-left order)*:

```
s[i]    = s[i] % 10;
s[i-1] += s[i] / 10;
```

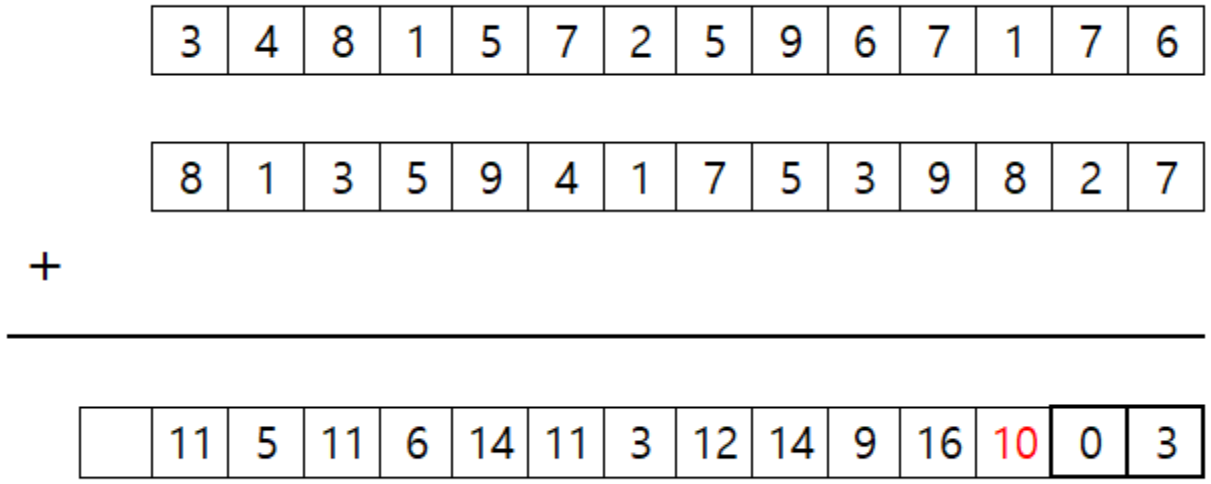
Big Integer (addition)



For each `s[i]` *(right-to-left order)*:

```
s[i]    = s[i] % 10;
s[i-1] += s[i] / 10;
```

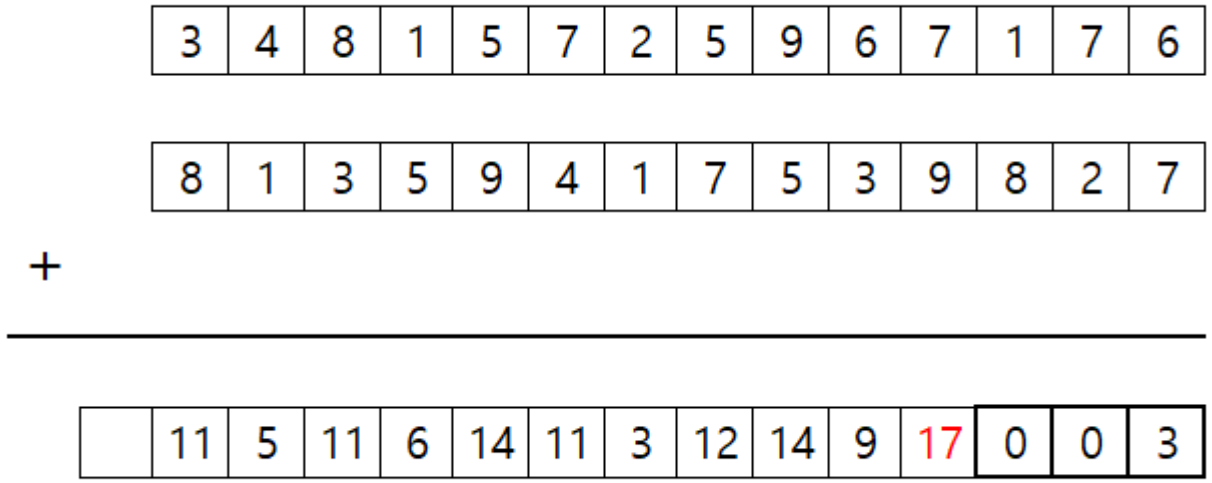
Big Integer (addition)



For each `s[i]` *(right-to-left order)*:

```
s[i]    = s[i] % 10;
s[i-1] += s[i] / 10;
```

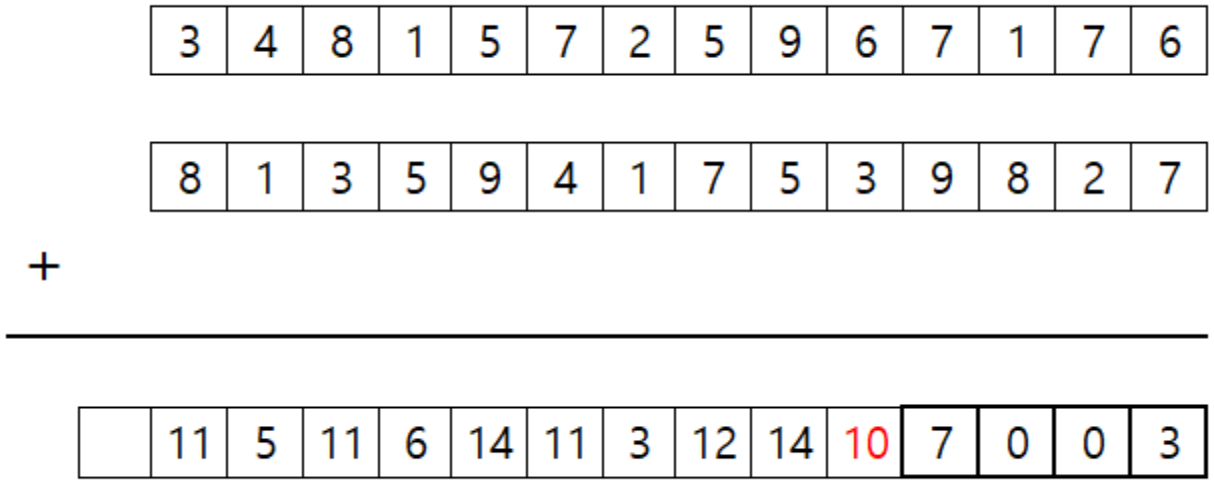
Big Integer (addition)



For each `s[i]` *(right-to-left order)*:

```
s[i]    = s[i] % 10;
s[i-1] += s[i] / 10;
```

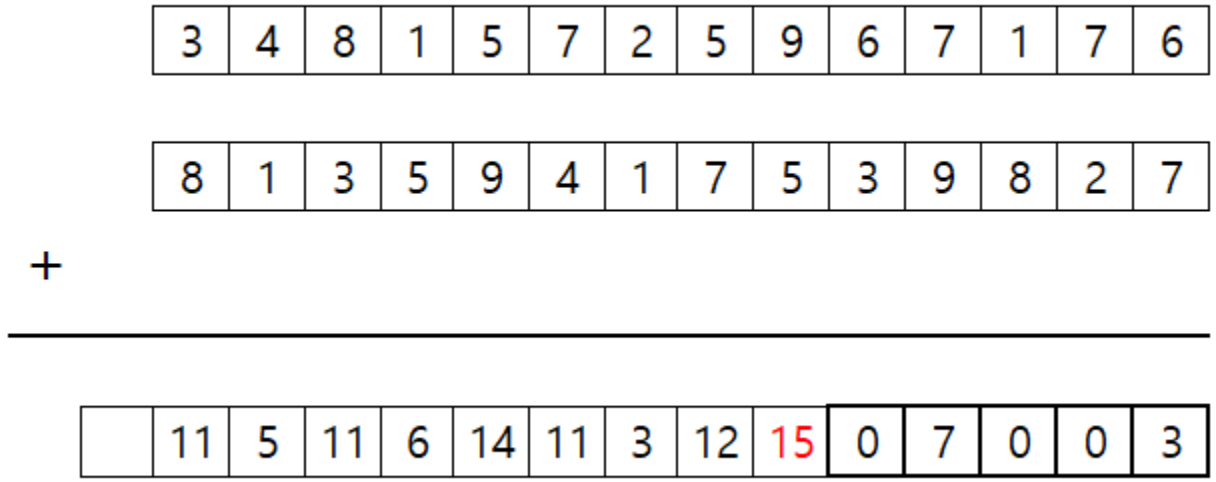
Big Integer (addition)



For each `s[i]` *(right-to-left order)*:

```
s[i]    = s[i] % 10;
s[i-1] += s[i] / 10;
```

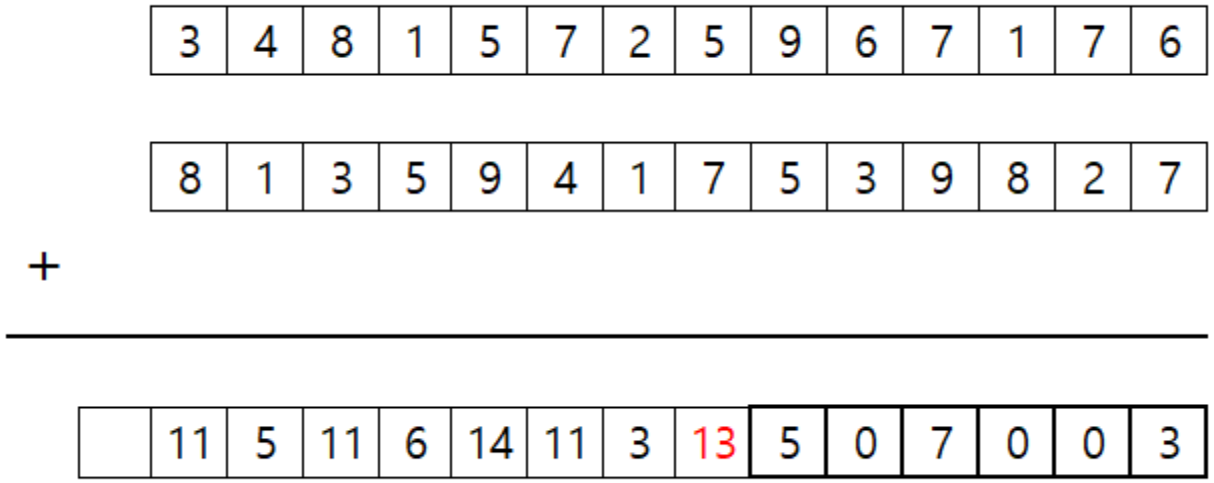
Big Integer (addition)



For each `s[i]` *(right-to-left order)*:

```
s[i]    = s[i] % 10;
s[i-1] += s[i] / 10;
```

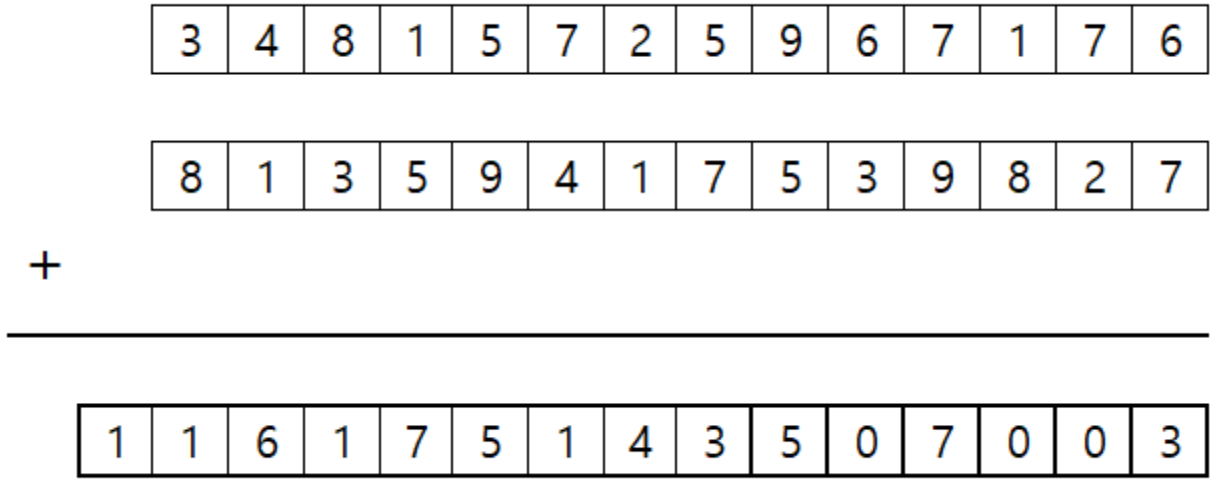
Big Integer (addition)



For each `s[i]` *(right-to-left order)*:

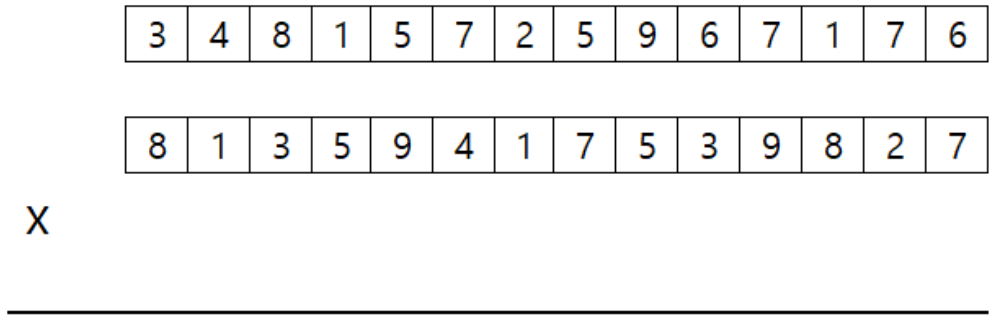
```
s[i]    = s[i] % 10;
s[i-1] += s[i] / 10;
```

Big Integer (addition)

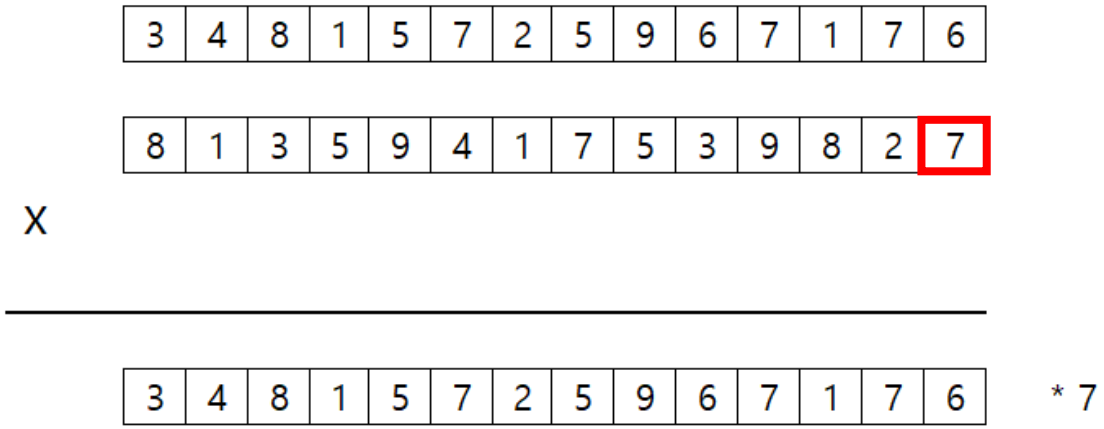


Completed.

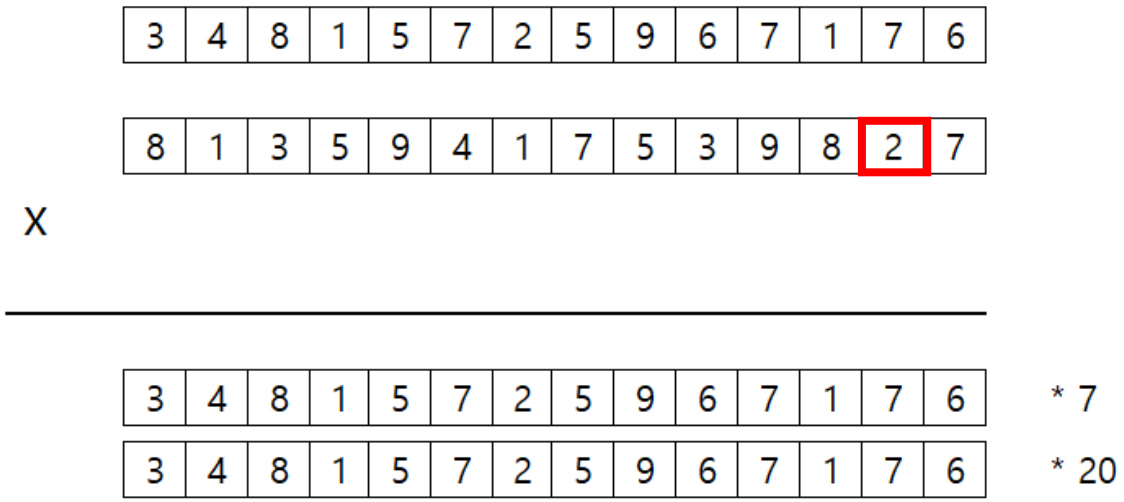
Big Integer (multiplication)



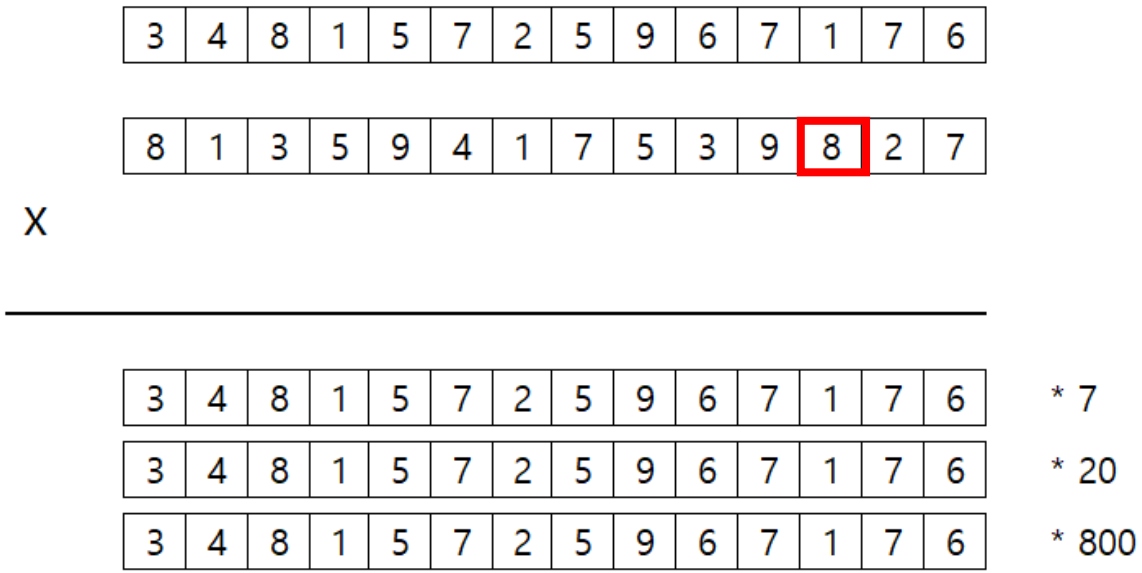
Big Integer (multiplication)



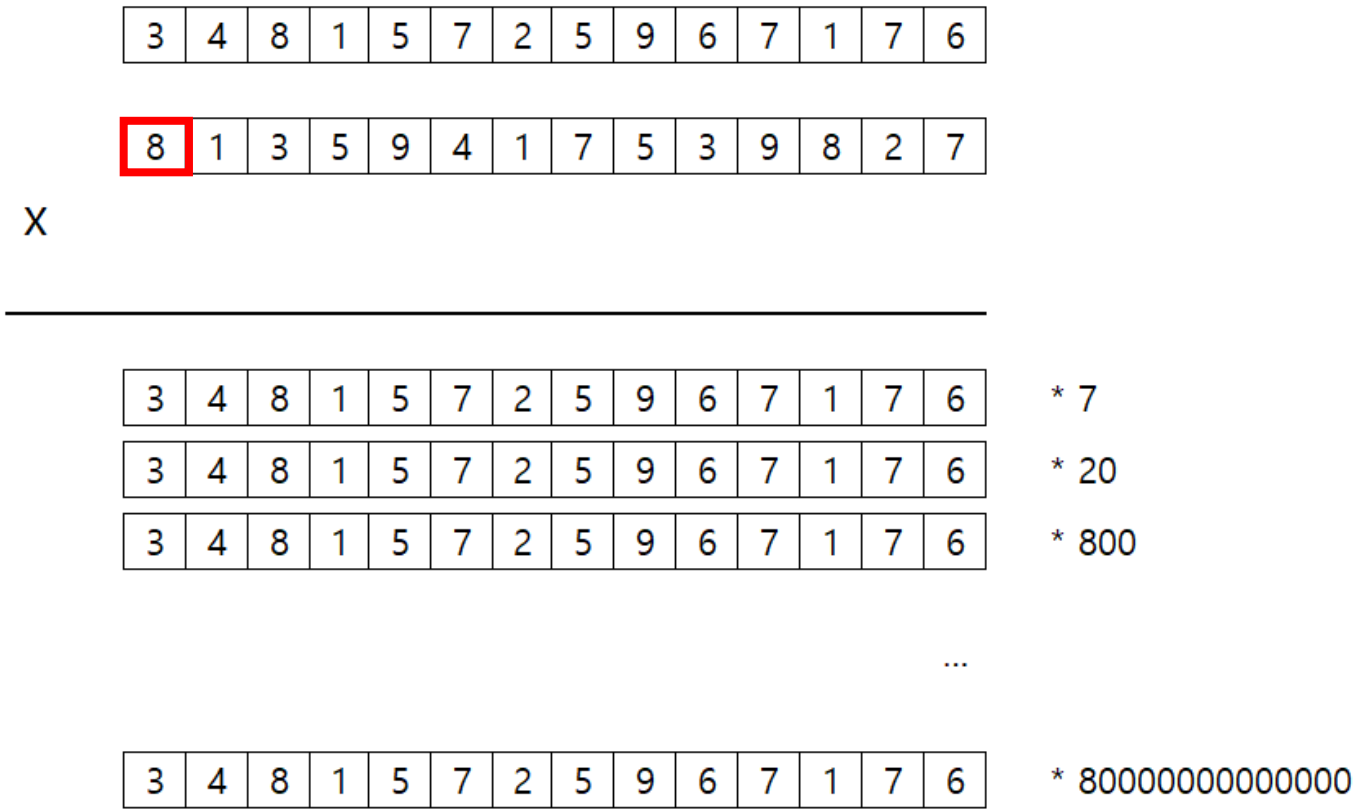
Big Integer (multiplication)



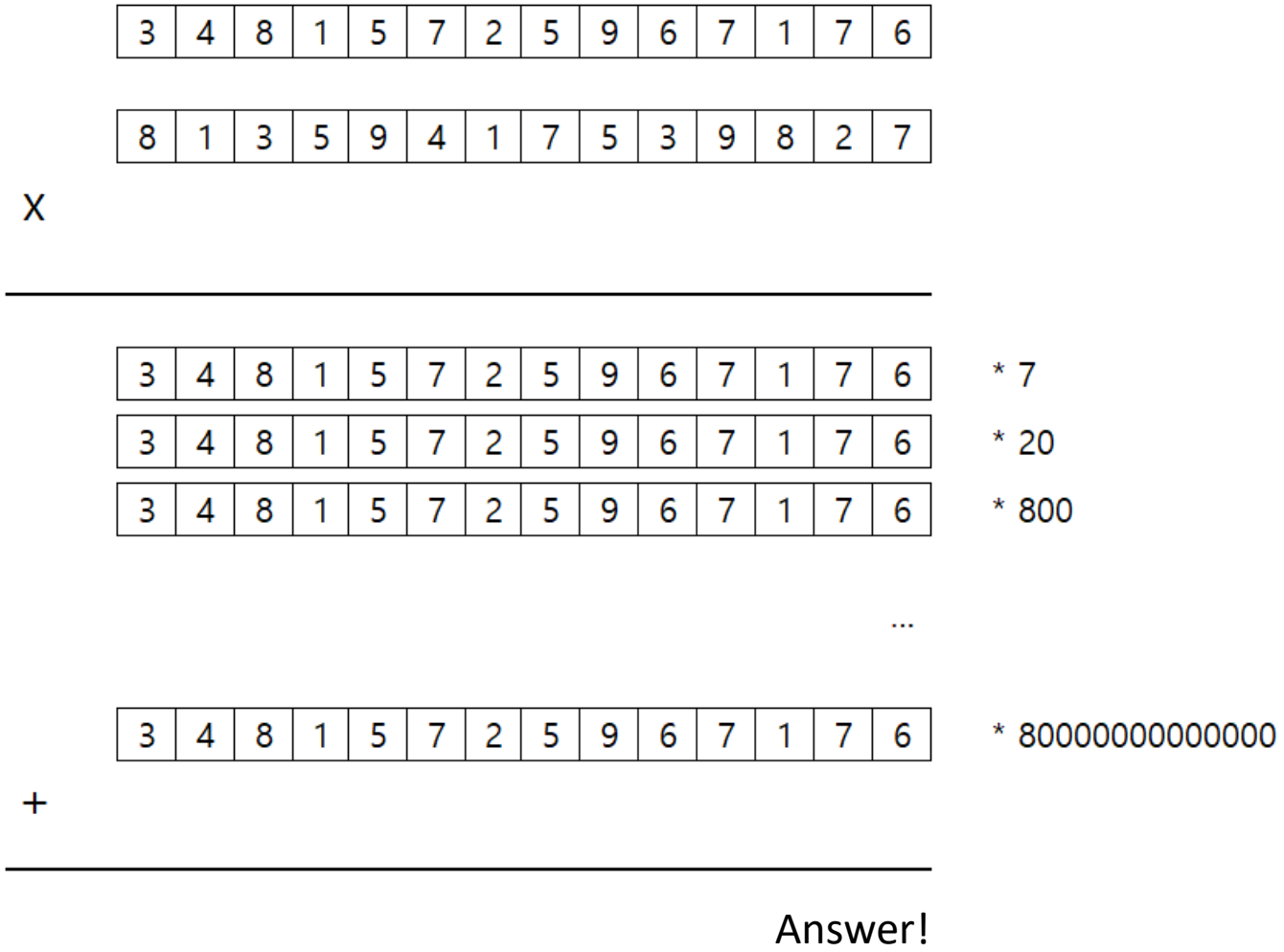
Big Integer (multiplication)



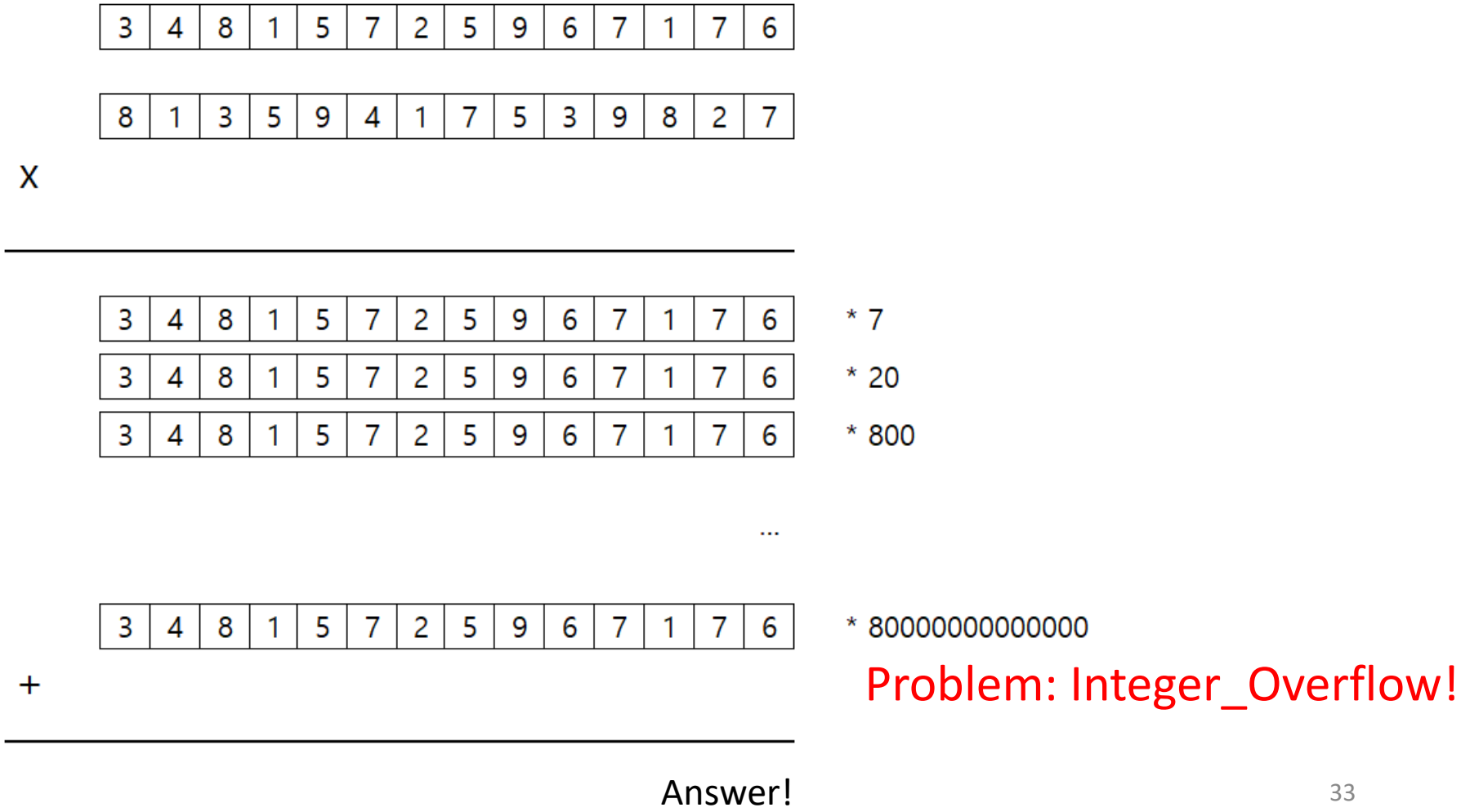
Big Integer (multiplication)



Big Integer (multiplication)

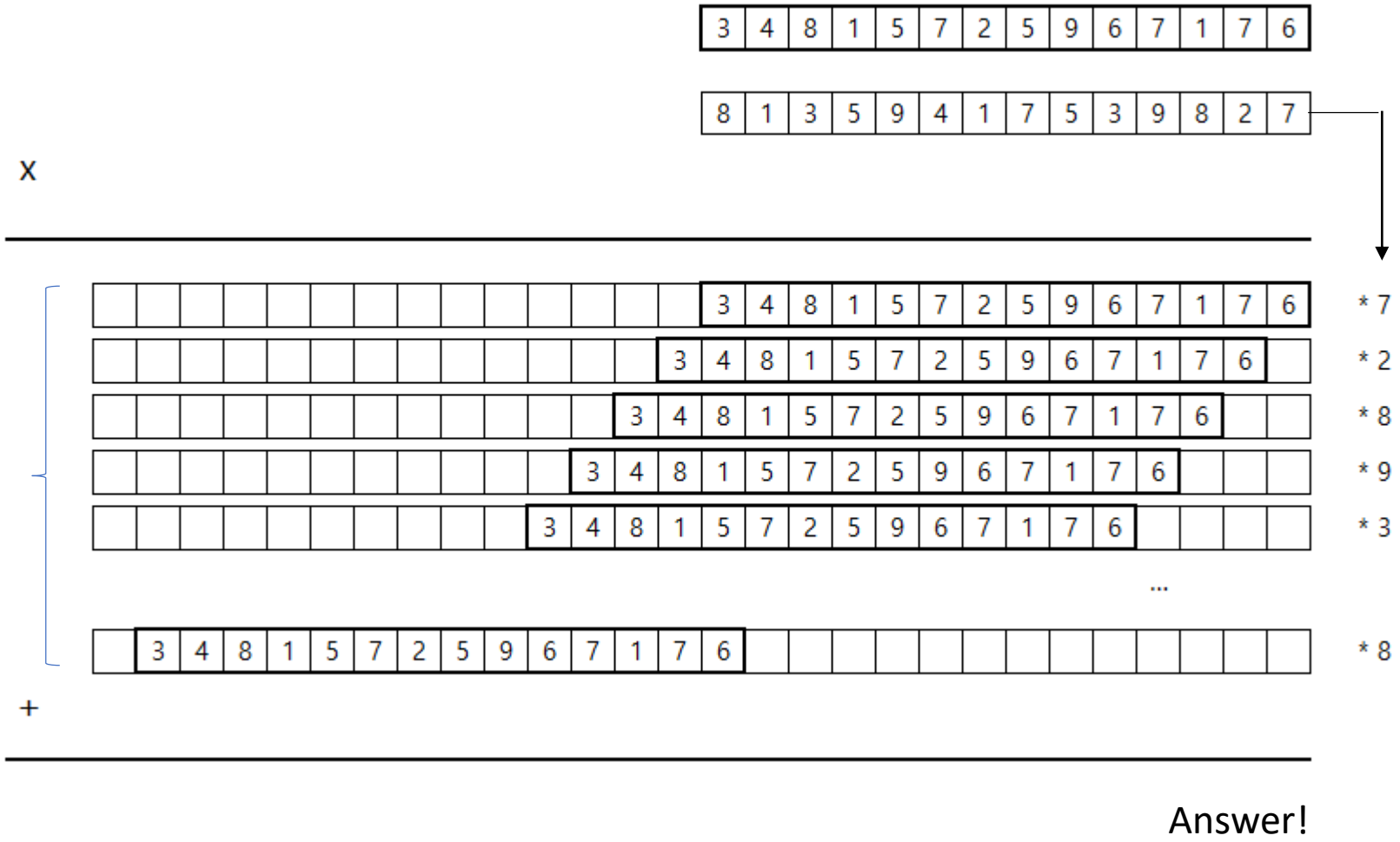


Big Integer (multiplication)



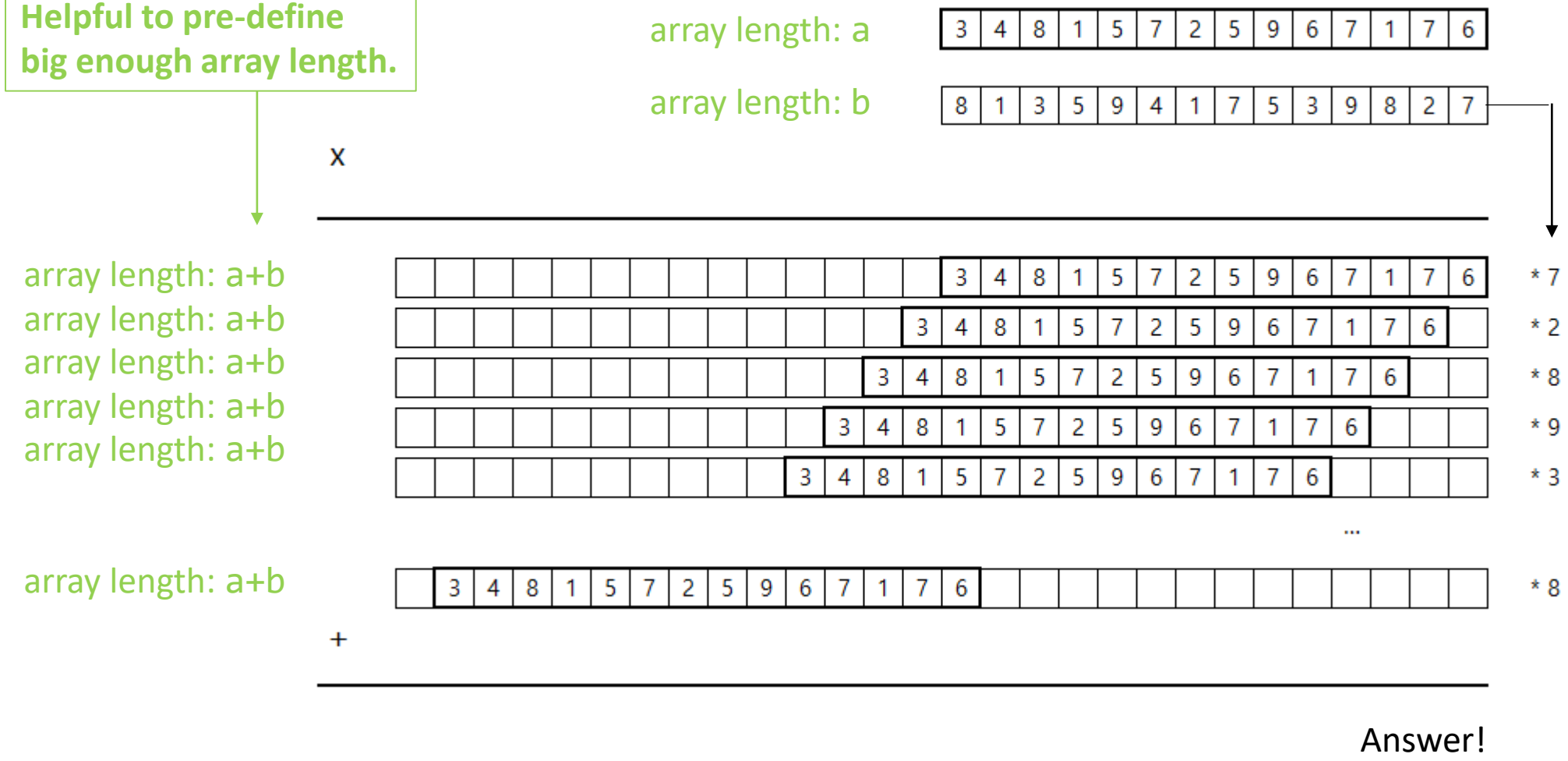
Big Integer (multiplication)

- Solution:



Big Integer (multiplication)

Helpful to pre-define big enough array length.



Answer!