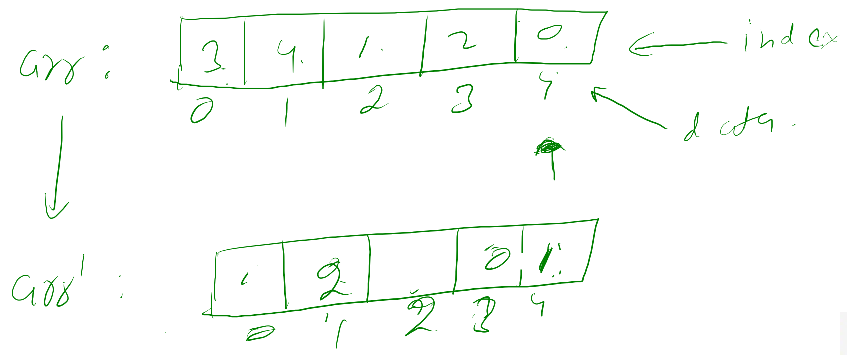
①

[0 ≤ arr[i] ≤ N-1]

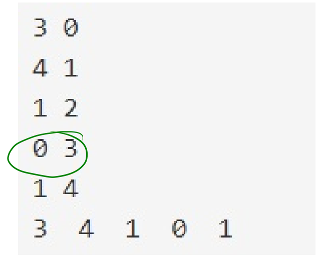
N = 5

(0,1,2,3,4)

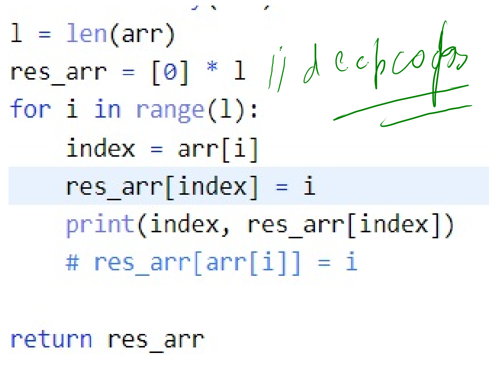


(inverse of array)

(3,0)
(4,1)
(1,2)
(2,3)



```
3 0
4 1
1 2
0 3
1 4
3  4  1  0  1
```

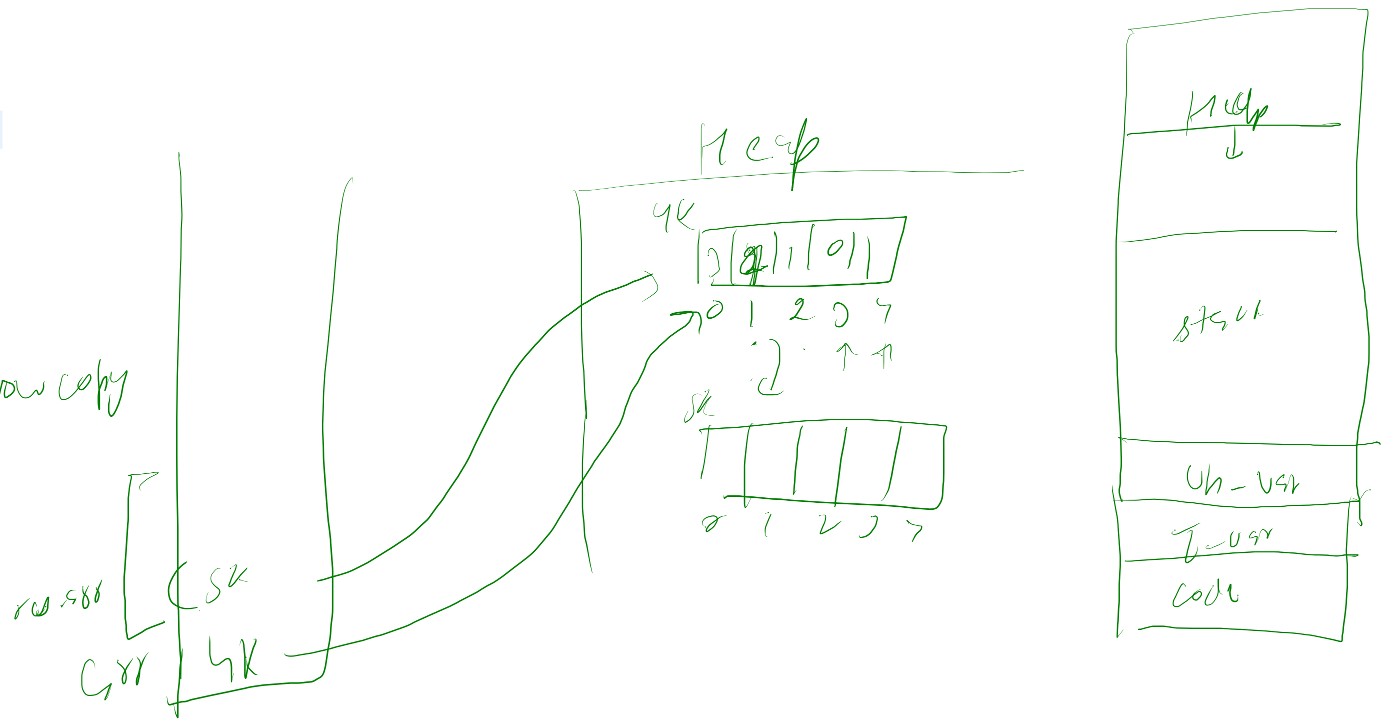


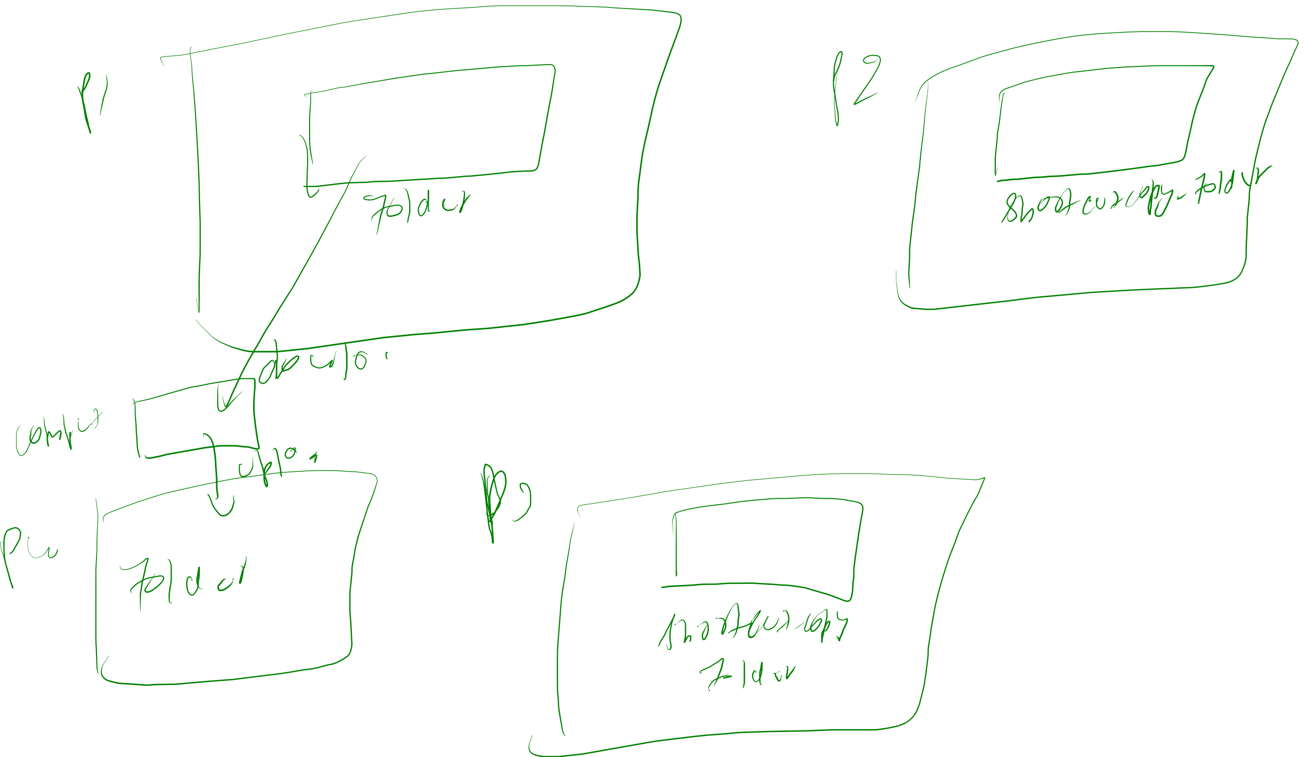
```
l = len(arr)
res_arr = [0] * l
for i in range(l):
    index = arr[i]
    res_arr[index] = i
    print(index, res_arr[index])
    # res_arr[arr[i]] = i

return res_arr
```

// deepcopy

res_arr = arr // shallow copy

inverse arr





0 0 0 0 0 9 9 9 9 8 9
5 8 9 9

0 0 0 0 | 1 0 0 0 |

add

① digit array

9	9	9	9	9	9	9
0	1	2	3	4	5	6

n = 7

i

j

8	8	8	8	8
0	1	2	3	4

m = 5

~~0~~ 1	0	0	8	8	8	8	~~8~~ 7
0	1	2	3	4	5	6	7

i = ~~6~~ ~~5~~ ~~4~~ ~~3~~ ~~2~~ ~~1~~ ~~0~~ -1

j = ~~4~~ ~~3~~ ~~2~~ ~~1~~ ~~0~~ -1

k = ~~7~~ ~~6~~ ~~5~~ ~~4~~ ~~3~~ ~~2~~ 1

carry = ~~0~~ ~~1~~ ~~1~~ ~~1~~ ~~1~~ ~~1~~ 0

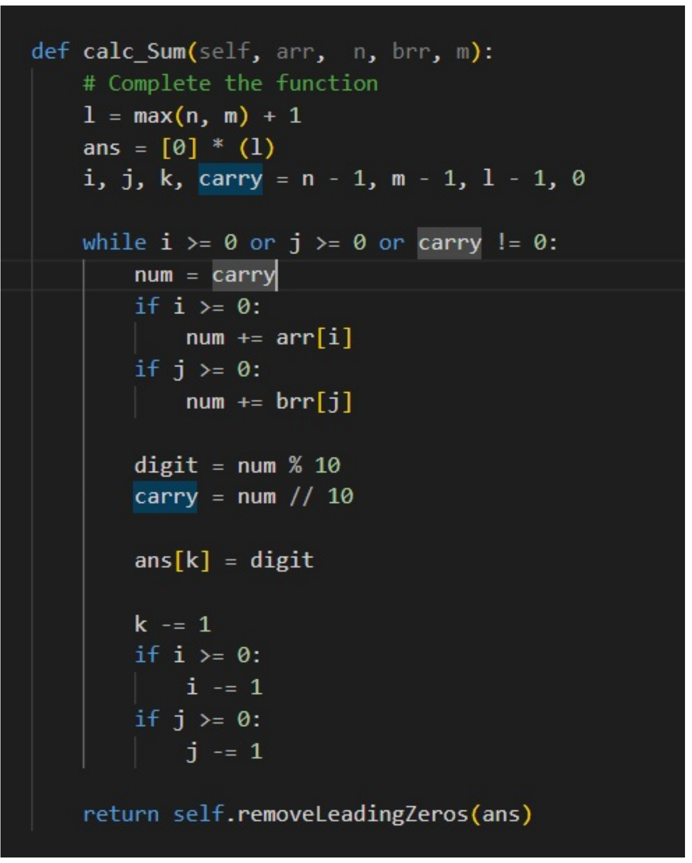
num = 1

$$\begin{array}{r} 9999999 \\ +\ 88888 \\ \hline 10088887 \end{array}$$

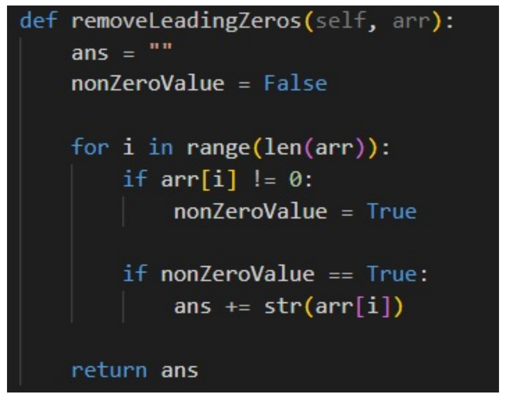
$\boxed{arr1 \geq arr2}$

arr1 − arr2

$$\begin{array}{r} 78842 \\ 663 \\ \hline 78179 \end{array}$$



```python
def calc_Sum(self, arr,  n, brr, m):
    # Complete the function
    l = max(n, m) + 1
    ans = [0] * (l)
    i, j, k, carry = n - 1, m - 1, l - 1, 0

    while i >= 0 or j >= 0 or carry != 0:
        num = carry
        if i >= 0:
            num += arr[i]
        if j >= 0:
            num += brr[j]

        digit = num % 10
        carry = num // 10

        ans[k] = digit

        k -= 1
        if i >= 0:
            i -= 1
        if j >= 0:
            j -= 1

    return self.removeLeadingZeros(ans)
```



```python
def removeLeadingZeros(self, arr):
    ans = ""
    nonZeroValue = False

    for i in range(len(arr)):
        if arr[i] != 0:
            nonZeroValue = True

        if nonZeroValue == True:
            ans += str(arr[i])

    return ans
```

0	9	9	9	9	9
0	1	2	3	4	5

2	2	2	2
0	1	2	3

0	1	0	2	2	2	1
0	1	2	3	4	5	6

K

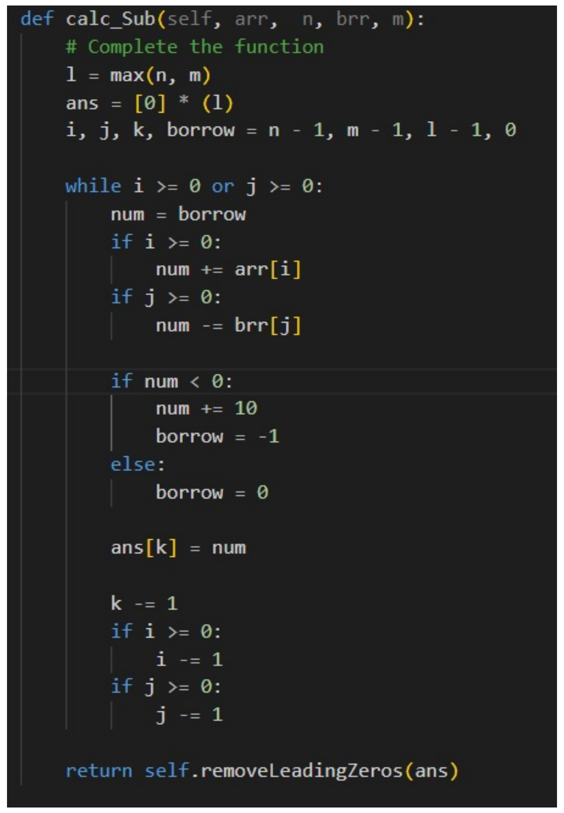
① 9 9 9 9 9 9 9 9 9
9 9 9 9 9 9 9 9 8

② 1 1 1 1 1
1 1 1 1 1

③ 1 0 0 0 0 0 0 0 0
9

```
 0 1 1 -1
 9 3 2 3 2
-    7 6 8
 ---------
 9 2 4 6 4
```

9 - 0 = 9

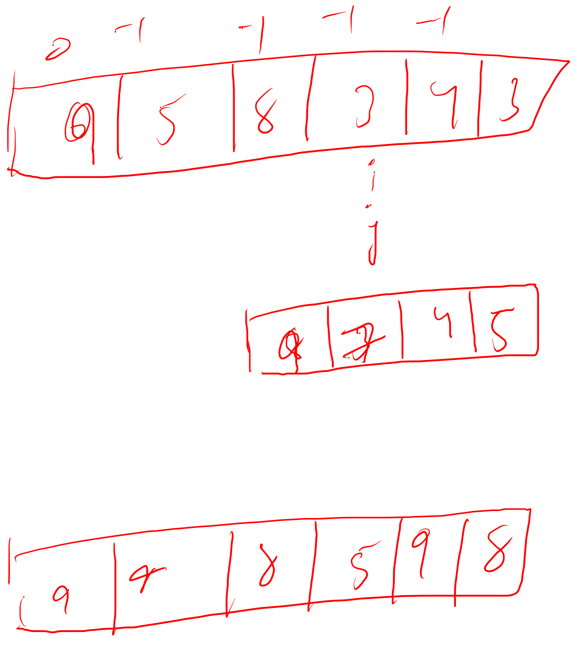


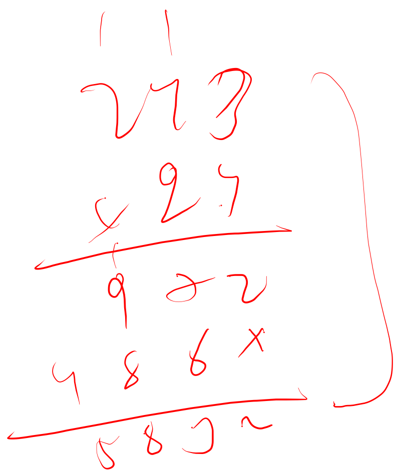
```
def calc_Sub(self, arr,  n, brr, m):
    # Complete the function
    l = max(n, m)
    ans = [0] * (l)
    i, j, k, borrow = n - 1, m - 1, l - 1, 0

    while i >= 0 or j >= 0:
        num = borrow
        if i >= 0:
            num += arr[i]
        if j >= 0:
            num -= brr[j]

        if num < 0:
            num += 10
            borrow = -1
        else:
            borrow = 0

        ans[k] = num

        k -= 1
        if i >= 0:
            i -= 1
        if j >= 0:
            j -= 1

    return self.removeLeadingZeros(ans)
```

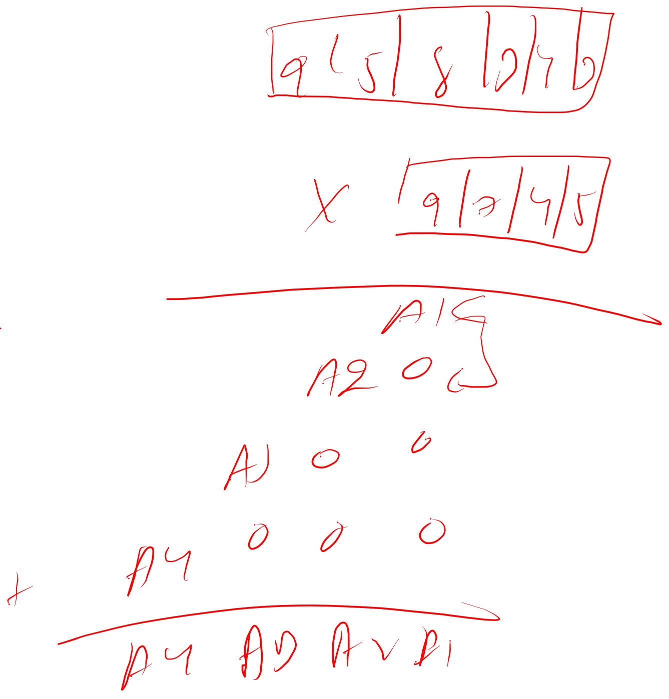
h1-h4

(h1+h2)

(h1*h2)







A : max (n, m)

S : max (n, m)

+, - > * > % > //

$\frac{p}{q} = \frac{r}{s}$ | $p \times s = r \times q$

(Faster)

① (substring)

"abcd"
0 1 2 3

a
ab
abc
abcd

| b
bc
bcd

| c
cd

| d

[(0,0)
(0,1)
(0,2)
(0,3)
| (1,1)
(1,2)
(1,3)
| (2,2)
(2,3)
| (3,3)]

②

ab6acca

a, b, b, a, c, c, a

abba, bb, acca, cc,