Roman To Integer

"III" → 3

LX → 70

CD → -100 + 500 = 400

CCC XXX V → 100 + 100 + 100 + 10 + 10 + 10 + 5 = 335

XLIII = -10 + 50 + 1 + 1 + 1 = 43

MCCXLVIII = 1000 + 100 + 100 - 10 + 50 + 5 + 1 + 1 + 1 = 1248

XI I
XII II
XIII III
XIV IV
XV V
XVI VI
XVII VII
XVIII VIII
XIX IX
XX X

I	1
V	5
X	10
L	50
C	100
D	500
M	1000

IV
I V
-1 + 5
4

V I I I
5 + 1 + 1 + 1
8

Strings

```
int sum=0 , index=0
while (index < s.size() -1){
if ( Num( s[index]) <Num(s[index+1]) {
        sum -= num(s[index])
else {
    } sum += num(s[index]) ;
index++;
}
sum += num(s[s.size() -1]);
return sum;
}
```

S = X L V → O(N)

0 1 2

sum: 45

index: ~~0~~ → 2

✓ 10 < 50

→ 50 < 10

```
int num (char ch){
    if (ch == 'I')
        return 1;
    else if (ch == 'V')
        return 5;
    else if (ch == 'X')
        return 10;
```

O(1)

Integer to Roman

1248

1000 + 200 + 40 + 8

M CC XL VIII

1000 200 40 8

M CC XL VIII

1000

MCCXLVIII

O(1)

100	10	1
200	20	2
300	30	3
400	40	4
500	50	5
600	60	6
700	70	7
800	80	8
900	90	9

L - 3999

3749

3000 + 700 + 40 + 9

MMM DCC XL IX

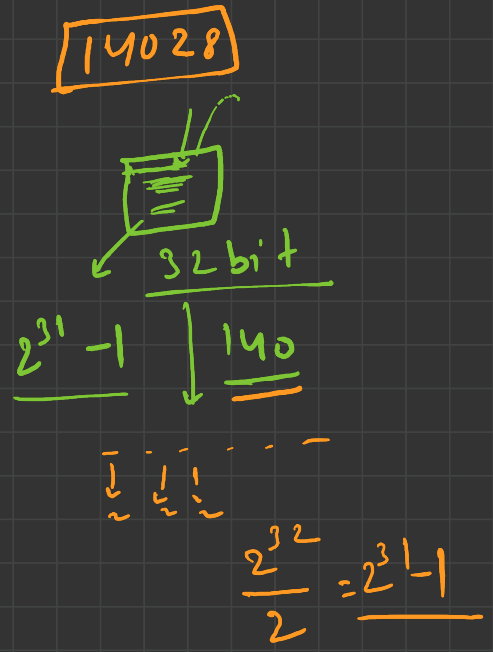
MMMDCCXLIX

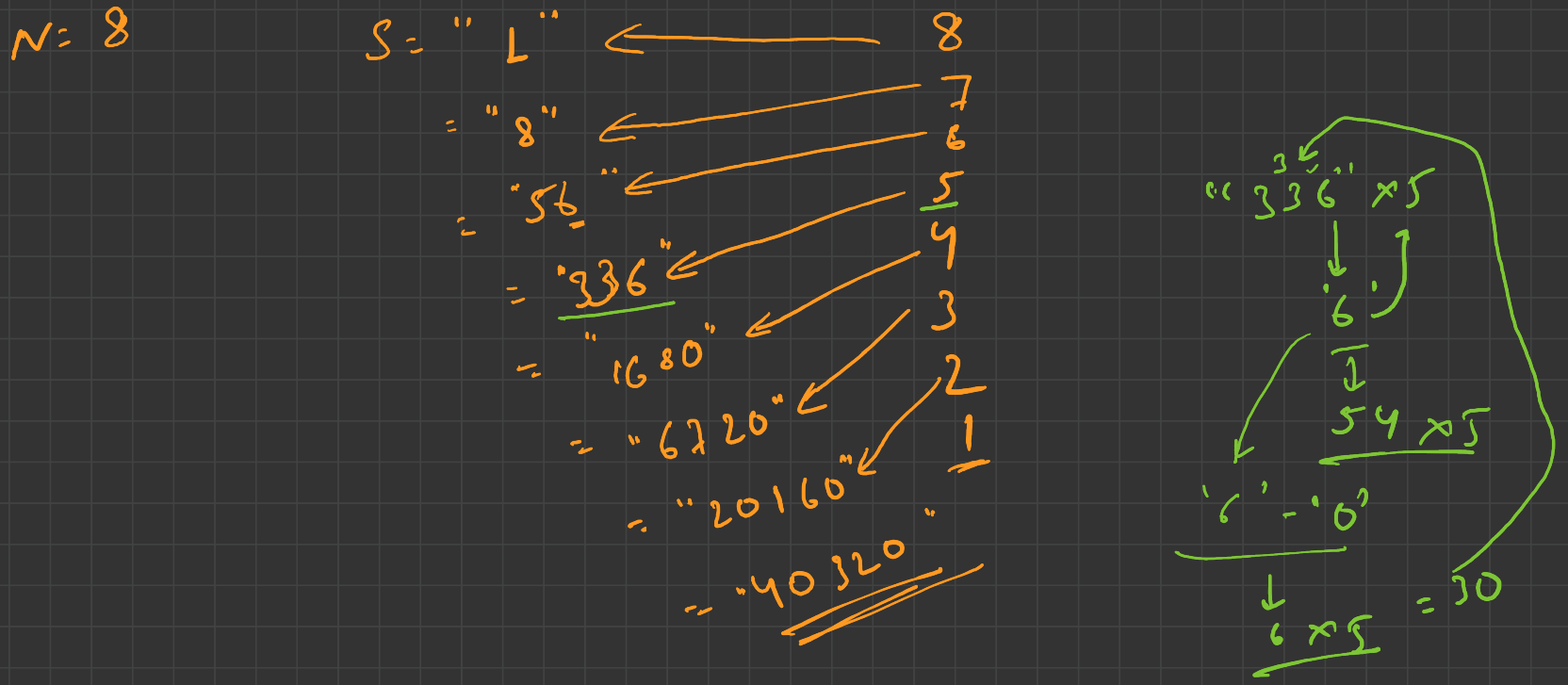
Factorials of large numbers

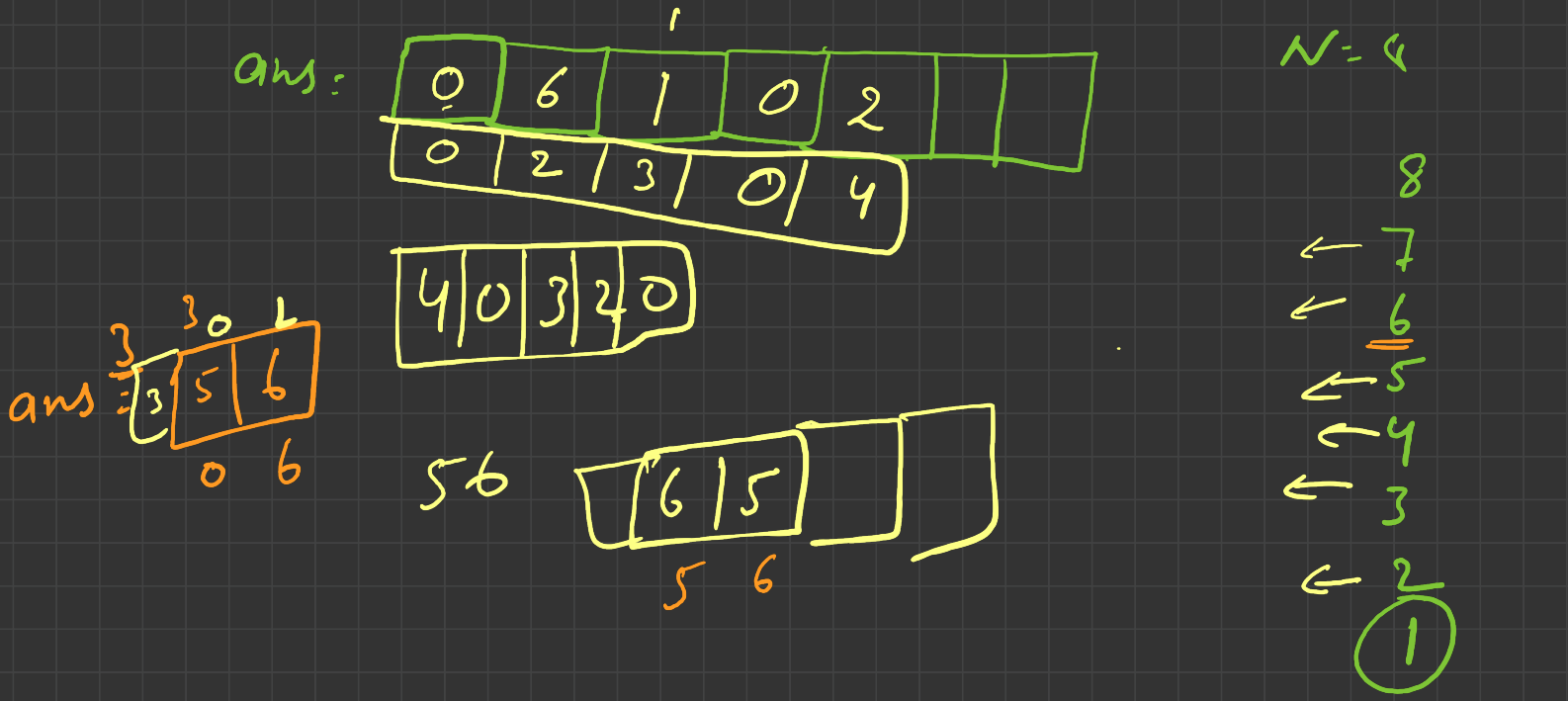
N : 8

INT_OVerflow

long long → 8bit







N = 8

```
vector<int> ans(1,1);
while (N>1) {
  int carry=0, size=ans.size(), result;
  for(i=0; i<size; i++) {
    result = ans[i]*N + carry;
    carry = result/10;
    ans[i] = result%10;
  }
  while(carry) {
    ans.push_back(carry%10);
    carry/=10;
  }
  N--
}
```

```
reverse(ans.begin(), ans.end());
```