

$[0, n]$

Ans: $[3, 0, 1]$
↑ ↑ ↑

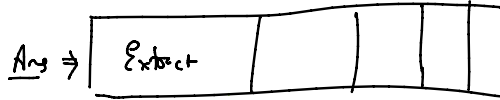
key	value
int	int
3	1
0	1
1	1

$mp[key]$

for (0 to n) {

if ($mp[i] \neq 1$) { i is ^{not} present }

string currWord = "Extract"
vector<str> ans.

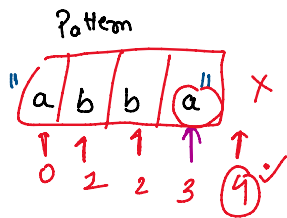


$s =$ "Extract all words from this sentence"
↑ ↑ ↑ ↑ ↑

if ($s[i] == ' '$) {
 ans.push_back(currWord);
 currWord = "";
}
else currWord += s[i];

currWord → dog
currCh → a

currWord = ""



s "dog cat cat fish -"
X
false

Map1

char	String
a	dog
b	cat

Map2

String	char
dog	a
cat	b

Test-1

for (0 to n)

if ($s[i] == ' '$) {

char currCh = pattern[chIdx];

if ($mp1[currCh] == Yes$)

string alreadyMapped = mp1[currCh]
if (alreadyMapped == currWord) → ✓
false → X

Test-2

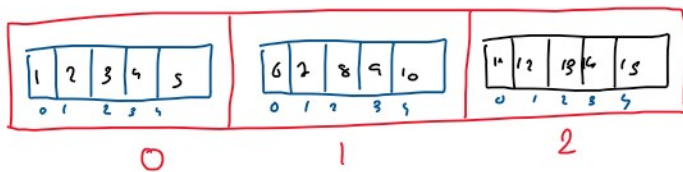
if (0) → 0X

if (1, 2, 3) → ✓

```

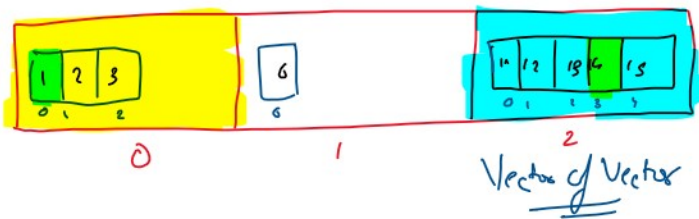
if (mp2[currWord] == Yes)
    char alreadyMapped = mp2[currWord];
    if (alreadyMapped == currCh) ← Yes ✓
    else {
        mp1[currCh] = currWord
        mp2[currWord] = currCh
        currWord = ""
    }
else
    currWord += s[i]
    
```

2D Arrays / 2D Vectors



Array of Array

3 → (15)



3 → 9 elements

Vector of Vector

[0][0]

arr[2][3] ⇒ 14

