

## Permutation in C++

```
#include <iostream>
using namespace std;

void permutations(int cb, int nboxes, int items[], int
ssf, int ritems, string asf) {
    if (cb > nboxes) {
        if (ssf == ritems) {
            cout << asf << endl;
        }
        return;
    }

    for (int i = 0; i < ritems; i++) {
        if (items[i] == 0) {
            items[i] = 1;
            permutations(cb + 1, nboxes, items, ssf + 1,
ritems, asf + to_string(i + 1));
            items[i] = 0;
        }
    }

    permutations(cb + 1, nboxes, items, ssf, ritems, asf
+ "0");
}

int main() {
    int nboxes = 3;
    int ritems = 2;
    int cb = 1;
    int ssf = 0;
    int items[ritems] = {0}; // Initialize items array with
0s

    permutations(cb, nboxes, items, ssf, ritems, "");

    return 0;
}
```

### Key Variables:

Var	Meaning
cb	current box index
ssf	selected so far – number of items placed
items[]	array of 0/1, indicating whether each item (1 to ritems) is used
asf	answer so far – the configuration of items across boxes

### 📄 Dry Run Table:

cb	items	ssf	asf	Description
1	[0,0]	0	""	Start, box 1
2	[1,0]	1	"1"	Place item 1 in box 1
3	[1,1]	2	"12"	Place item 2 in box 2
4	[1,1]	2	"120"	✓ Output: item1 in box1, item2 in box2
3	[1,1]	2	"102"	item2 in box3
3	[1,0]	1	"10"	skip box 2
4	[1,1]	2	"102"	✓ Output
2	[0,1]	1	"2"	item2 in box 1
3	[1,1]	2	"21"	item1 in box2
4	[1,1]	2	"210"	✓ Output
3	[0,1]	1	"20"	box2 empty
4	[1,1]	2	"201"	✓ Output
2	[0,0]	0	"0"	box1 empty
3	[1,0]	1	"01"	item1 in box2
4	[1,1]	2	"012"	✓ Output
3	[0,1]	1	"02"	item2 in box2
4	[1,1]	2	"021"	✓ Output
3	[0,0]	0	"00"	box2 empty
4	-	0	"000"	✗ Not valid – ssf < ritems

Output:-

102
210
201
012
021