

## bubblesort

100 puncte

Miruna a învățat recent la ora de informatică algoritmul bubble sort. Mai jos puteți vedea codul algoritmului în limbajul C++:

```
int steps = 0;
while (true) {
    ++steps;
    bool isSorted = true;
    for (int i = 1; i < n; ++i)
        if (v[i] > v[i + 1]) {
            swap(v[i], v[i + 1]);
            isSorted = false;
        }
    if (isSorted) break;
};
```

Miruna definește ordinul unei permutări ca fiind egal cu numărul de pași necesari algoritmului pentru a sorta permutarea (adică valoarea variabilei *steps* în codul de mai sus la sfârșitul execuției).

### Cerință

Dându-se trei numere naturale  $N$ ,  $M$  și  $K$  aflați care este cea de a  $K$ -a permutare în ordine lexicografică de lungime  $N$  și ordin  $M$ .

### Date de intrare

Pe prima linie a fișierului de intrare `bubblesort.in` se vor afla trei numere naturale  $N$ ,  $M$  și  $K$ , având semnificația de mai sus.

### Date de ieșire

Pe prima linie a fișierului de ieșire `bubblesort.out` veți afișa  $N$  numere naturale despărțite prin spațiu reprezentând permutarea cerută.

### Restricții și precizări

- $1 \leq N \leq 300$
- $1 \leq M \leq N$
- Se garantează existența soluției pe datele de test.

### Exemplu

bubblesort.in	bubblesort.out
7 4 123	1 5 3 6 2 4 7
20 5 1097525229548	2 7 1 3 4 6 5 10 13 11 9 8 15 12 19 17 16 14 20 18

**Timp maxim de execuție/test:** 0.5 secunde/test

**Memorie totală disponibilă:** 16 MB, din care 2 MB pentru stivă