



Car Parking

PROBLEM

A parking center by the Great Wall has a long row of parking places. One end of the row is considered left and the other end is considered right. The row is full of cars. Each car has a type and several cars may be of the same type. The types are identified by integer values. A number of workers decide to order the cars parked in the row in ascending order from left to right by the car types using the following method. In what is called a round, each of the workers can simultaneously drive one car out of its place and then park it in a place from where a car has been moved out in the same round. It may be that some workers are not moving a car in a round. For efficiency, a small number of rounds is preferable.

Suppose that N is the number of cars and W is the number of workers. You are to write a program which, given the types of the parked cars and the number of workers, finds such a way to sort the cars that the number of rounds needed is at most $\lceil N/(W-1) \rceil$, that is $N/(W-1)$ rounded up. The minimal number of rounds is never greater than $\lceil N/(W-1) \rceil$.

Consider the following example. There are 10 parked cars of types 1,2,3, and 4 with 4 workers. The initial placement of the cars from left to right identified by their types is 2 3 3 4 4 2 1 1 3 1.

The minimal number of rounds is three, and the rounds can be implemented so that the placement after each round is as follows:

2 1 1 4 4 2 3 3 3 1 – after round 1,
2 1 1 2 4 3 3 3 4 1 – after round 2, and
1 1 1 2 2 3 3 3 4 4 – after round 3.

INPUT

The input file name is CAR.IN. The first line in the input file contains three integers. The first integer is the number of cars N , $2 \leq N \leq 20000$. The second integer is the number of types M , $2 \leq M \leq 50$. The car types are identified by the integers from 1 to M . There is at least one car of each type. The third integer is the number of workers W , $2 \leq W \leq M$. The second line contains N integers, where the i th integer is the type of the i th car in the row, starting from the left end of the row.

OUTPUT

The output file name is CAR.OUT. The first line of the output file contains one integer R , which is the number of rounds in the solution. The next R lines describe the rounds ordered from 1 to R . In each line, the first integer is the number of cars C , which are moved in that round. After that follow $2C$ integers, identifying car positions. The car positions are identified by the integers from 1 to N starting at the left end. The



first two are a pair describing how one of the cars is moved: the first integer is the position from the left end before the round and the second is the position from the left after the round. The next two integers are a pair describing how another car is moved, and so on. There may be several different solutions for these R lines, and your program only needs to output one of them.

EXAMPLE INPUT AND OUTPUT

CAR.IN

```
10 4 4
2 3 3 4 4 2 1 1 3 1
```

CAR.OUT

```
3
4 2 7 3 8 7 2 8 3
3 4 9 9 6 6 4
3 1 5 5 10 10 1
```

PARTIAL CREDIT

Suppose that your program's output for an evaluation run is R and $\lceil N/(W-1) \rceil$ is Q . If in your program's output the R rounds are not described correctly or they do not produce the desired order for the cars, then your score is 0. Otherwise, your score will be calculated from the maximum score as follows.

- $R \leq Q$ 100% Score
- $R = Q + 1$ 50% Score
- $R = Q + 2$ 20% Score
- $R \geq Q + 3$ 0% Score