



Ticket Office

Input file: ticket.in

Output file: ticket.out

Source code: ticket.pas/.c/.cpp

100 points

Time limit: 0.5 sec

Memory limit: 64 MB

A ticket office sells tickets for concerts. Instead of selling tickets for single seats it sells bunches of tickets for a fixed number of consecutive seats. The office has received a great number of purchase orders. A purchase order for one bunch of seats specifies the lowest seat number in the bunch.

The office may not be able to fulfill all of the purchase orders. Moreover, if it only allocates seats exactly as requested in the purchase orders then a great number of seats could remain empty. Therefore, the office applies the following allocation and pricing strategy. If a purchase order is accepted and the allocated seats are exactly those that are requested then the purchaser pays full-price (2 petaks for the bunch). If a purchase order is accepted, but the allocated seats differ from the requested ones (on at least one position) then the purchaser pays half-price (1 petak for the bunch).

The goal is to maximize the total income.

Task

You are to write a program that computes the maximal income that can be achieved (Subtask A) and allocates the seats to the selected orders (Subtask B) achieving this income.

Input

The first line of the text file `ticket.in` contains two integers, M and L . M ($1 \leq M \leq 30\,000$) is the number of seats and L ($1 \leq L \leq 100$) is the number of consecutive seats in every bunch. Seats are numbered from 1 to M . The second line contains an integer, N ($1 \leq N \leq 100\,000$), the number of purchase orders. The third line contains N integers, defining the purchase orders. The i th number in the line, z ($1 \leq z \leq M-L+1$), means that the i th purchaser requests the bunch of seats starting at seat z and ending at seat $z+L-1$.

Output

The first line of the text file `ticket.out` contains an integer, S , the maximal income (Subtask A). The second line contains an integer, Q , the number of accepted orders. The next Q lines describe the seat allocations (Subtask B). Each line contains a pair of integers, $x y$. The pair $x y$ means that the purchaser x gets the seats starting at seat number y . The lines must be written in the increasing order of seat numbers.

If there are multiple possibilities, your program should output only one; it does not matter which one.

Example

| ticket.in | ticket.out |
|-------------------|------------|
| 20 3 | 9 |
| 7 | 6 |
| 4 2 10 9 16 15 17 | 4 1 |
| | 1 4 |
| | 2 7 |
| | 3 10 |
| | 6 13 |
| | 5 16 |