Post Office

PROBLEM

There is a straight highway with villages alongside the highway. The highway is represented as an integer axis, and the position of each village is identified with a single integer coordinate. There are no two villages in the same position. The distance between two positions is the absolute value of the difference of their integer coordinates.

Post offices will be built in some, but not necessarily all of the villages. A village and the post office in it have the same position. For building the post offices, their positions should be chosen so that the total sum of all distances between each village and its nearest post office is minimum.

You are to write a program which, given the positions of the villages and the number of post offices, computes the least possible sum of all distances between each village and its nearest post office, and the respective desired positions of the post offices.

INPUT

The input file name is `POST.IN`. The first line contains two integers: the first is the number of villages V, $1 \le V \le 300$, and the second is the number of post offices P, $1 \le P \le 30$, $P \le V$. The second line contains V integers in increasing order. These V integers are the positions of the villages. For each position X it holds that $1 \le X \le 10000$.

OUTPUT

The output file name is `POST.OUT`. The first line contains one integer S, which is the sum of all distances between each village and its nearest post office as reported in the second line. The second line contains P integers in increasing order. These integers are the locations of the distinct villages in which the post offices will be built. There may be several different solutions for the locations, and your program needs to output only one of them.

EXAMPLE INPUT AND OUTPUT

`POST.IN`

```
10 5
1 2 3 6 7 9 11 22 44 50
```

`POST.OUT`

```
9
2 7 22 44 50
```

PARTIAL CREDIT

If your program's output does not satisfy the output requirements, then your score is 0. Otherwise, your score will be computed based on Table 1 as follows. If your program outputs sum *S* and the actual least possible sum is *Smin*, then calculate *q=S/Smin* and find your score *c* in the bottom row.

q=S/Smin	*q*=1.0	1.0<*q*≤1.1	1.1<*q*≤1.15	1.15<*q*≤1.2	1.2<*q*≤1.25	1.25<*q*≤1.3	1.3<*q*
c	10	5	4	3	2	1	0

Table 1.