Problem 2: Justify My Love

Until now, e-Mote (a highly successful Internet dating service) has found potential dates for new clients by scoring each such client on 8 binary attributes and finding those persons stored in the e-Mote database that have the highest number of matches with the client's attribute-descriptions. For example, a client with an attribute-description of 10101010 would be more compatible with a date whose attribute-description was 10101001 (6 matches) than 10010101 (2 matches). Given recent NASA announcements, e-Mote is preparing for opening its first interstellar branches by broadening its criteria to allow dates of size $k \geq 2$ consisting of a client and $k-1$ entities from the e-Mote database with the highest mutable compatibility, *i.e.*, the highest number of attributes whose values across the client and the selected $k-1$ selected entities are identical.

Write a program which, given attribute-descriptions of a client and l entities, $1 \leq l \leq 31$, as well as the size k of the requested date, computes and prints all dates with the highest mutual compatibility. Your input will be an $(2+l)$-line textfile, in which the first line is the binary attribute-descriptions of the client, the second line specifies the values of l and k, and the remaining l lines are the attribute-descriptions of the candidate entities (one per line). You may assume that all input files are formatted correctly.

Sample input #1 (available as file "test2a.dat"):

```
10101010
3 2
10010101
10101001
10100101
```

Sample output #1:

```
Most compatible dates:
> client + #2
```

Sample input #2 (available as file "test2b.dat"):

```
10101010
3 3
10010101
10101001
10100101
```

Sample output #2:

```
Most compatible dates:
> client + #2 + #3
```

Sample input #3 (available as file "test2c.dat"):

```
10101010
5 3
10101010
10010101
10101001
10100101
10100110
```

Sample output #3:

```
Most compatible dates:
> client + #1 + #3
> client + #1 + #5
```