Problem A. Master Zhu and Magic Numbers

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Master Zhu has n magic numbers. The i-th number a_i can be represented by a binary string of length i which can contain leading zeroes. When we reverse and concatenate these binary strings, we get a long string s of length $\frac{n(n+1)}{2}$. A substring from $s\left[\frac{i(i-1)}{2}\right]$ to $s\left[\frac{i(i+1)}{2}-1\right]$ inclusive is the binary representation of a_i, **from lowest to highest** digit. Here, the long string s is indexed starting from zero.

One day, Rin inputs Master Zhu's magic numbers into a program to create n new magic numbers. The i-th new number is b_i. Here is the code of the program in a C-like language.

```
for (int i = 1; i <= n; i++) {
  b[i] = 0,
  flag[i] = 0;
}

for (int i = 1; i <= n; i++) {
  for (int j = 1; j <= n; j++) {
    if (((1 << (j - 1)) & a[i]) > 0) {
      if (!flag[i]) {
        b[i] = b[j],
        flag[i] = 1;
      } else {
        b[i] = b[i] & b[j];
      }
    }
  }
  b[i] = b[i] ^ (1 << (i - 1));
}
```

In the code above, "`x << y`" is bitwise shift to the left, which is equivalent to multiplying `x` by $2^{\texttt{y}}$, "`x & y`" is bitwise AND, and "`x ^ y`" is bitwise XOR. We assume that the numbers a_i and b_i can have arbitrarily many bits.

After that, Rin wants to ask q questions. The i-th question is a number c_i which can be represented in binary notation as a string of length len_i and can contain leading zeroes. When we reverse and concatenate these binary strings, we get a long string t of length L_q, where $L_k = \sum_{i=1}^{k} len_i$. A substring from $t\,[L_{i-1}]$ to $t\,[L_i - 1]$ inclusive is the binary representation of c_i, **from lowest to highest** digit. Here, the long string t is indexed starting from zero.

For each question, Rin requires Illya to calculate the number d_i:

```
for (int i = 1; i <= q; i++) {
  d[i] = 0;
  for (int j = 1; j <= min (n, len[i]); j++) {
    if (((1 << (j - 1)) & c[i]) > 0) {
      d[i] = d[i] | b[j];
    }
  }
}
```

Here, "`x | y`" is bitwise OR. We assume that the numbers c_i and d_i can have arbitrarily many bits.

The answer to the i-th question is the number of ones in the binary representation of d_i. Help Illya answer all the questions!

Input

The first line of the input contains two integers n and m denoting the number of magic numbers and the number of ones in the long string s ($1 \le n \le 5000$, $1 \le m \le 10^6$).

The second line contains m integers, the i-th integer p_i denotes that $s[p_i] = 1$, and all other digits of s are equal to 0 ($0 \le p_i < \frac{n(n+1)}{2}$, all p_i are distinct).

The third line contains two integers q and r denoting the number of questions and the number of ones in the long string t ($1 \le q \le 5000$, $1 \le r \le 10^6$).

The fourth line contains q integers, the i-th integer len_i denotes the length of binary representation of the i-th query c_i ($1 \le len_i \le 10^9$). It is guaranteed that $L_q = \sum_{i=1}^{q} len_i$ is at most 10^9.

The fifth line contains r integers, the i-th integer z_i denotes that $t[z_i] = 1$, and all other digits of t are equal to 0 ($0 \le z_i < L_q$, all z_i are distinct).

Output

For each question i, print a single line with a single integer: the number of ones in the binary representation of d_i.

Example

standard input	standard output
3 4 0 1 4 5 3 6 2 3 3 0 1 2 4 6 7	2 3 3