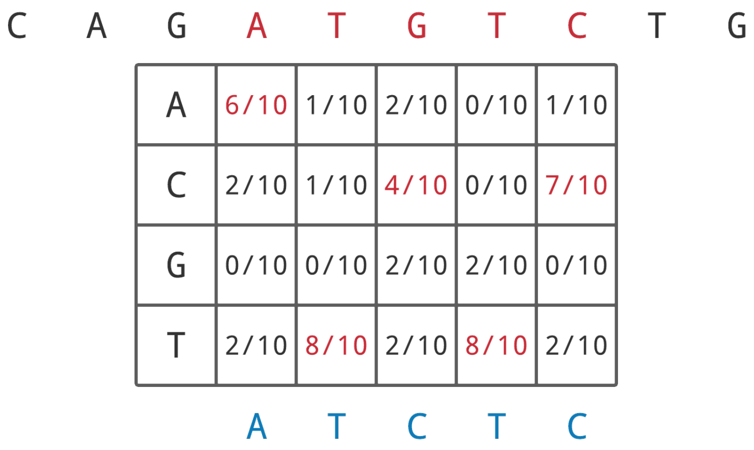
2C Find a Profile-most Probable *k*-mer in a String

Profile-most Probable *k*-mer Problem
Find a profile-most probable k-mer in a string.

Input: A string *Text*, an integer *k*, and a 4 × *k* matrix *Profile*.
Output: A *Profile*-most probable *k*-mer in *Text*.



A	6/10	1/10	2/10	0/10	1/10
C	2/10	1/10	4/10	0/10	7/10
G	0/10	0/10	2/10	2/10	0/10
T	2/10	8/10	2/10	8/10	2/10

Formatting

Input: A string *Text*, an integer *k*, and a 4 × *k* matrix *Profile* of floats.
Output: A string representing a *Profile*-most probable *k*-mer in *Text* (If multiple answers exist, you may return any one).

Constraints

- The length of *Text* will be between 1 and 10^3.
- The integer *k* will be between 1 and 10^1.
- *Text* will be a DNA string.

Test Cases

Case 1

Description: The sample dataset is not actually run on your code.

Input:

```
ACCTGTTTATTGCCTAAGTTCCGAACAAACCCAATATAGCCCGAGGGCCT
5
0.2 0.2 0.3 0.2 0.3
0.4 0.3 0.1 0.5 0.1
0.3 0.3 0.5 0.2 0.4
0.1 0.2 0.1 0.1 0.2
```

Output:

```
CCGAG
```

Case 2

Description: This dataset checks for off-by-one errors at the beginning of *Text*. Notice that the optimal solution (AGCAGCTT) occurs at the very beginning of *Text*, so if your code does not check this *k*-mer, then your code will output a different (incorrect) *k*-mer as the solution.

Input:

```
AGCAGCTTTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATCTGAACTGGT...
...TACCTGCCGTGAGTAAAT
8
0.7 0.2 0.1 0.5 0.4 0.3 0.2 0.1
0.2 0.2 0.5 0.4 0.2 0.3 0.1 0.6
0.1 0.3 0.2 0.1 0.2 0.1 0.4 0.2
0.0 0.3 0.2 0.0 0.2 0.3 0.3 0.1
```

Output:

```
AGCAGCTT
```

Case 3

Description: This dataset checks for off-by-one errors at the end of *Text*. Notice that the optimal solution (AAGCAGAGTTTA) occurs at the very end of *Text*, so if your code does not check this *k*-mer, then your code will output a different (incorrect) *k*-mer as the solution.

Input:

```
TTACCATGGGACCGCTGACTGATTTCTGGCGTCAGCGTGATGCTGGTGTGGATGACATTCCGGTGCGCTT...
...TGTAAGCAGAGTTTA
12
0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.1 0.2 0.3 0.4 0.5
0.3 0.2 0.1 0.1 0.2 0.1 0.1 0.4 0.3 0.2 0.2 0.1
0.2 0.1 0.4 0.3 0.1 0.1 0.1 0.3 0.1 0.1 0.2 0.1
0.3 0.4 0.1 0.1 0.1 0.1 0.0 0.2 0.4 0.4 0.2 0.3
```

Output:

```
AAGCAGAGTTTA
```

Case 4

Description: A larger dataset of the same size as that provided by the randomized autograder.