

Unit-3 Itertools, Iterators & Generator Expressions

(Note: Q1-Q8: Itertools, Q9: Iterators, Q10-15 Generator Expressions)

1. You are given two lists A and B. Your task is to compute their Cartesian product AxB

Example:

```
A = [1, 2]
B = [3, 4]
AxB = [(1, 3), (1, 4), (2, 3), (2, 4)]
```

Sample Input:

```
1 2
3 4
```

Sample Output:

```
(1, 3) (1, 4) (2, 3) (2, 4)
```

2. You are given a string S. Your task is to print all possible permutations of size k of the string in lexicographic sorted order.

Sample Input:

```
HACK 2
```

Sample Output:

```
AC
AH
AK
CA
CH
CK
HA
HC
HK
KA
KC
KH
```

3. You are given a string S. Your task is to print all possible combinations, up to size k, of the string in lexicographic sorted order.

Sample Input:

```
HACK 2
```

Sample Output:

```
A
C
H
K
AC
AH
AK
CH
CK
HK
```

4. You are given a string S. Your task is to print all possible size k replacement combinations of the string in lexicographic sorted order.

Sample Input:

HACK 2

Sample Output:

AA
AC
AH
AK
CC
CH
CK
HH
HK
KK

5. You are given a string S. suppose a character 'c' occurs consecutively X times in the string. Replace these consecutive occurrences of the character 'c' with (X,c) in the string.

Sample Input:

1222311

Sample Output:

(1, 1) (3, 2) (1, 3) (2, 1)

6. You are given a list of N lowercase English letters. For a given integer K, you can select any K indices (assume 1-based indexing) with a uniform probability from the list. Find the probability that at least one of the K indices selected will contain the letter 'a'

Input Format:

The input consists of three lines. The first line contains the integer N, denoting the length of the list. The next line consists of N space-separated lowercase English letters, denoting the elements of the list. The third and the last line of input contains the integer K, denoting the number of indices to be selected.

Output Format:

Output a single line consisting of the probability that at least one of the K indices selected contains the letter 'a'. The answer must be correct up to 3 decimal places.

Sample Input:

4
a a c d
2

Sample Output:

0.8333

7. Write a function `my_enumerate` that works like `enumerate`.
8. Implement a function `izip` that works like `itertools.izip`.
9. Write an iterator class `reverse_iter` that takes a list and iterates it from the reverse direction.
10. Write a program that takes more than one filename as arguments and prints all the lines which are longer than 40 characters.
11. Write a function `findfiles` that recursively descends the directory tree for the specified directory and generates paths of all the files in the tree.

12. Write a function to compute the number of python files (.py extension) in a specified directory recursively.
13. Write a function to compute the total number of lines of code in all python files in a specified directory recursively.
14. Write a function to compute the total number of lines of code, ignoring empty and comment lines, in all python files in the specified directory recursively.
15. Write a program `split.py` that takes an integer `n` and a filename as command line arguments and splits the file into multiple small files with each having `n` lines.