Description

Given a string S, output all different possible set of K characters in the string with P paddings. And sort them in the dictionary order. A padding is expressed as an underline '_'.

For example, if K=2 and P=1, and the given string S is ‘CDBABBD’, the output would be

```
_AB
_AC
_AD
_BB
_BC
_BD
_CD
_DD
A_B
A_C
A_D
AB_
AC_
AD_
B_B
B_C
B_D
BB_
BC_
BD_
C_D
CD_
D_D
DD_
```

Input

The first line of input contains a positive integer T (T <= 30), which indicates the number of test cases. For each case, there is a string S, a positive integer K, and a nonnegative integer P in a line. The length of the S is less than or equal to 100 and S contains only 'A'-'J'; The number K, less than or equal to 10, indicates the length of substrings.

For test 1: T <= 10, K <= 3, P <= 1, |S| <=10

For test 2: T <= 15, K <= 5, P <= 1, |S| <=25

For test 3: T <= 20, K <= 8, P <= 2, |S| <=50

For test 4: T <= 30, K <= 10, P <= 3, |S| <=100

T, K, |S| are all positive integers, P is a nonnegative integer, and K <= |S| for all test cases.

Output

For each test case, print all different possible sets of K characters in the string. And sort them in the dictionary order, one substring per line. Print a blank line after each test case.

Sample Input

```
2
CDBABBD 2 0
ABC 3 2
```

EOF

Sample Output

```
AB
AC
AD
BB
BC
BD
CD
DD

__ABC
_A_BC
_AB_C
```

ABC

A__BC

A_B_C

A_BC_

AB__C

AB_C_

ABC__