Problem 3304: Find the K-th Character in String Game I

Problem Information

Difficulty: Easy
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

Alice and Bob are playing a game. Initially, Alice has a string

word = "a"

.

You are given a

positive

integer

k

.

Now Bob will ask Alice to perform the following operation

forever

:

Generate a new string by

changing

each character in

word

to its

next

character in the English alphabet, and

append

it to the

original

word

.

For example, performing the operation on

"c"

generates

"cd"

and performing the operation on

"zb"

generates

"zbac"

.

Return the value of the

k

th

character in

word

, after enough operations have been done for

word

to have

at least

k

characters.

Example 1:

Input:

k = 5

Output:

"b"

Explanation:

Initially,

word = "a"

. We need to do the operation three times:

Generated string is

"b"

,

word

becomes

"ab"

.

Generated string is

"bc"

,

word

becomes

"abbc"

.

Generated string is

"bccd"

,

word

becomes

"abbcbccd"

.

Example 2:

Input:

k = 10

Output:

"c"

Constraints:

1 <= k <= 500

Code Snippets

C++:

```
class Solution {
public:
char kthCharacter(int k) {

}
};
```

Java:

```
class Solution {
public char kthCharacter(int k) {

}
}
```

Python3:

```
class Solution:
def kthCharacter(self, k: int) -> str:
```

Python:

```
class Solution(object):
def kthCharacter(self, k):
"""
:type k: int
:rtype: str
"""
```

JavaScript:

```
/**
* @param {number} k
* @return {character}
*/
var kthCharacter = function(k) {

};
```

TypeScript:

```
function kthCharacter(k: number): string {

};
```

C#:

```
public class Solution {
public char KthCharacter(int k) {

}
}
```

C:

```
char kthCharacter(int k) {

}
```

Go:

```
func kthCharacter(k int) byte {

}
```

Kotlin:

```
class Solution {
fun kthCharacter(k: Int): Char {

}
}
```

Swift:

```
class Solution {
func kthCharacter(_ k: Int) -> Character {

}
}
```

Rust:

```
impl Solution {
pub fn kth_character(k: i32) -> char {

}
}
```

Ruby:

```
# @param {Integer} k
# @return {Character}
def kth_character(k)

end
```

PHP:

```
class Solution {

/**
* @param Integer $k
* @return String
```

```
*/
function kthCharacter($k) {

}
}
```

Dart:

```
class Solution {
String kthCharacter(int k) {

}
}
```

Scala:

```
object Solution {
def kthCharacter(k: Int): Char = {

}
}
```

Elixir:

```
defmodule Solution do
@spec kth_character(k :: integer) :: char
def kth_character(k) do

end
end
```

Erlang:

```
-spec kth_character(K :: integer()) -> char().
kth_character(K) ->
.
```

Racket:

```
(define/contract (kth-character k)
(-> exact-integer? char?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Find the K-th Character in String Game I
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public:
char kthCharacter(int k) {

}
};
```

Java Solution:

```
/**
* Problem: Find the K-th Character in String Game I
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public char kthCharacter(int k) {

}
}
```

Python3 Solution:

```
"""
Problem: Find the K-th Character in String Game I
Difficulty: Easy
Tags: string, math

Approach: String manipulation with hash map or two pointers
Time Complexity: O(n) or O(n log n)
Space Complexity: O(1) to O(n) depending on approach
"""

class Solution:
def kthCharacter(self, k: int) -> str:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def kthCharacter(self, k):
"""
:type k: int
:rtype: str
"""
```

JavaScript Solution:

```
/**
* Problem: Find the K-th Character in String Game I
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

/**
* @param {number} k
* @return {character}
*/
var kthCharacter = function(k) {
```

```
};
```

TypeScript Solution:

```
/**
* Problem: Find the K-th Character in String Game I
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

function kthCharacter(k: number): string {

};
```

C# Solution:

```
/*
* Problem: Find the K-th Character in String Game I
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

public class Solution {
public char KthCharacter(int k) {

}
}
```

C Solution:

```
/*
* Problem: Find the K-th Character in String Game I
* Difficulty: Easy
```

```
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

char kthCharacter(int k) {

}
```

Go Solution:

```
// Problem: Find the K-th Character in String Game I
// Difficulty: Easy
// Tags: string, math
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

func kthCharacter(k int) byte {

}
```

Kotlin Solution:

```
class Solution {
fun kthCharacter(k: Int): Char {

}
}
```

Swift Solution:

```
class Solution {
func kthCharacter(_ k: Int) -> Character {

}
}
```

Rust Solution:

```
// Problem: Find the K-th Character in String Game I
// Difficulty: Easy
// Tags: string, math
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

impl Solution {
pub fn kth_character(k: i32) -> char {

}
}
```

Ruby Solution:

```
# @param {Integer} k
# @return {Character}
def kth_character(k)

end
```

PHP Solution:

```
class Solution {

/**
* @param Integer $k
* @return String
*/
function kthCharacter($k) {

}
}
```

Dart Solution:

```
class Solution {
String kthCharacter(int k) {
```

```
}
}
```

Scala Solution:

```
object Solution {
def kthCharacter(k: Int): Char = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec kth_character(k :: integer) :: char
def kth_character(k) do

end
end
```

Erlang Solution:

```
-spec kth_character(K :: integer()) -> char().
kth_character(K) ->
.
```

Racket Solution:

```
(define/contract (kth-character k)
(-> exact-integer? char?)
)
```