

## INDEX OR PATTERN MATCHING:->

The index operation determines the starting ~~the starting~~ location of the first occurrence of one string within a second string, or it gives value 0 if the first string does not appear in the second string.

Index operation is a special case of the more general pattern matching problem that occurs in many applications such as text editing or text processing.

There are so many algorithms are developed for pattern matching. One of the simplest method or algorithm is

BRUTE FORCE pattern matching algorithm

As an illustration suppose we want find the index of  $str1 = 'abcaba'$  in  $str2 = 'abcabcabdabba'$ .

str2: 

a	b	c	a	b	c	a	b	d	a	b	b	a
---	---	---	---	---	---	---	---	---	---	---	---	---

str1: 

a	b	c	a	b	a
---	---	---	---	---	---

The search for str1 in str2 begins at the first position of

Algo :- P

INDEX (~~str~~ Pattern, Text)

INDEX (PATTERN, TEXT)

Step 1 Initialize pos = 1

I = 1 and

J = 1

[ where pos is the beginning position of text being compared with pattern  
I is index running through pattern  
J is index running through text ]

② While  $I \leq m$  and  $J \leq n$  do the following.

( $m$  is length of pattern and  $n$  is length of text)

If  $Pat[I] = Text[J]$  Then

Set  $I := I + 1$  and  $J := J + 1$

(move right one character in each pat and text and continue search)

else

$pos := pos + 1$  ;  
 $J := pos$  ;  
 $I := 1$  .

(Backtrack and start the search over again after shifting one position to the right in text)

If  $I > m$  Then ( \* pat found in Text \* )  
Set Index = pos

else

set Index = 0

(4) End.

function Index (pattern, Text-string) : integer;  
var

I, J, pos, ~~L~~ PL, TL : integer;  
Begin

PL := Length (pattern);

TL := Length (Text);

pos := 1;

I := 1;

J := 1;

while (I <= PL) AND (J <= TL) do

if pattern[I] = Text[J] then

Begin

I = I + 1;

J = J + 1;

end;

else

Begin  
pos := pos + 1;  
J := pos;  
I := 1;

end;

if (I > PL) then

Index := pos (\* Found \*)

else

Index := 0 (\* not found \*)

## Data Encryption →

The basic string operations length, index, concatenation and substring as we know are important string operations in text-editing applications. Sometimes string operation encryption is used in other important appln.

Encryption refers to the coding of information in order to keep it secret.

The Encryption is accomplished by transforming the characters of string into another set of characters to produce a new string that is in encoded form.

This is also called Cryptogram or Ciphertext and may be safely stored or transmitted.

At a later time one can decipher the received codes by reversing the encrypting process to recover the original information which is also called plaintext.

## Encryption methods: →

There are several methods of data encryption but the most common methods are:-

- ① Caesar Cipher (Sezar cipher)
- ② Vignere Cipher
- ③ Substitution table
- ④ Permutation scheme.

### ① Caesar Cipher: →

Caesar cipher scheme of encryption is the simplest scheme which is based on string operation of substitution.

In this method the original string (plaintext) is traversed and each character is replaced by some other character according to a fixed rule.

For example replacing each letter in plaintext by the letter that appears  $k$  positions later in the alphabets ( $k$  is some integer).  
[The alphabets are thought of as being arranged in circular order with A following z.]

In the original Caesar  $k$  was 3, so that each