Brute Force String Matching

- **String** is a group of characters. It contains white spaces, numbers & symbols.

  Example : Hello John, how are you?
- **Pattern** is a substring.

  Example : John
- Length of Pattern String < Length of Text String for comparison.

Goal of brute force string matching

- Check whether a pattern string is present in the text string and if yes, at which position.

Example

- **Text string : "Have a good day!" Length = n = 16**
- **Pattern string : "day" Length = m = 3**
- Align the pattern string against the text string.
- Compare the characters of the pattern string with the characters of text string from left to right.
- If corresponding characters do not match keep shifting the pattern string by one place and align it to the next character of the text string.
- In this example he comparison of characters is continued by shifting the pattern string.
- At 11th position we get a match but pattern string "a" does not match with that of the text string as it has a blank space.
- Shift the pattern string and repeat the comparison.
- We find the presence of pattern string in the text string at position 12.

ENGINEERING MENTOR
STUDY SMARTER, SCORE BETTER

Algorithm

- **Input: A text array T [0..n – 1] , A pattern array P[0..m – 1]**
- **Output: a) Position of the pattern string in the text string**
- **b) –1 if the search is unsuccessful**

```
1.  for  i ← 0 to n – m do
2.          j ← 0
3.          while j <m  and  P[j ]= T [i + j ] do
4.                  j ← j + 1
5.           if j = m  return i
6.  return -1
```

ENGINEERINGMENTOR
STUDY SMARTER, SCORE BETTER