# STRING OPERATIONS WITHOUT INBUILT

In the programs discussed further in the slide:

You will not use any inbuilt string functions.

# DELETE A WORD FROM THE INDEX POSITION

Text : He is a good boy

Substr : good

Output : good Found at index 8

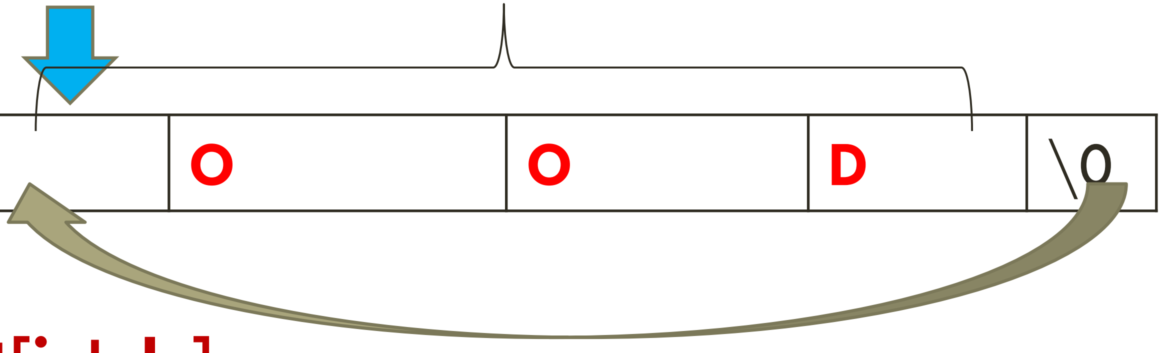
After deletion : He is a boy

i=0

**text[]**

Index returned=3

L : LENGTH OF SUBSTR



|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| H | E |   | G | O | O | D | \0 |

**text[i] = text[i + L ];**

New text[]

|   |   |   |   |
|---|---|---|---|
| H | E |   | \0 |

```
int search(char text[], char substr[])
{   int l, i, j;
    for (l = 0; substr[l] != '\0'; l++); /* finding length of the substring */
    i=0;j=0;
    while(text[i] != '\0' && substr[j] != '\0') //Proceed until either u finish the text or substring
     {
        if (text[i] == substr[j]) // if text and substring is same get a count of no of characters same
        {
            j++;  // is the count same as length of substring
        }
        else
        {
            j = 0; // else reset and be ready to search again the first character of the string
        }
        i++;  // Anyways text has to proceed
     }
    if (j == l)  // is the count j== length of substring l?
    {
        return (i - j); /* substring found */
    }
    else
    {
        return  - 1; //Not found
    } }
```

```
#include <stdio.h>
int search(char[],char[]);
void delete_word(char[],char[],int);
int main()
{    char str[80], word[50];
     int index;
     printf("Enter string:\n"); gets(str);
     printf("Enter word to delete:\n"); gets(word);
     index = search(str, word); //Found substr at index
     if (index !=  - 1)
     {   delete_word(str, word, index); //Call delete routine
         printf("String without word:\n%s", str); //str[] updated after shifting
     }
     else
     {  printf("The word not present in the string.");}
     return 0;}
```

```
void delete_substr(char text[], char substr[], int index)
{ int i, l;
    for (l = 0; substr[l] != '\0'; l++); /* finding length of word */

    for (i = index; text[i+l] != '\0'; i++)
    {
        text[i] = text[i + l]; /* shift the elements left to delete*/
    }
    text[i]='\0';
}
```

# TRY AS HOMEWORK

Given a text and a pattern, can you replace one pattern with another?

Input:

Text[] :He is a good boy

Word to replace[]: good

Replacetext[]: bad

Output:

Newtext[]: He is a bad boy

# Namah shivaya