28-05-2024

Q:- WAP to insert sub-string into a main - string from starting position 'm' [index]:-

```
=> main ( )
 {
    char a[40], b[20]., c[20];
    int i, j, index;
    printf ("enter main string: ");
    Scanf ("%s", a);
    printf ("enter sub string: ");
    Scanf ("%s", b);
    printf ("enter index to insert: ");
    Scanf ("%d", & index);
    count = 0;
    while (a[c]!='\0')
    count ++;
    if ( index <=0 && index > count +1 )
    {
      printf ("Not possible ");
    }
    else
    { ~~printf ("~~
      for (i=0;, j=0; j < index -1; j++, i++)   || (j++, i++)
        c[i]= a[j];
      for ( j=0;. b[j]! = '\0'; i++, j++)
        c[i] = b[j];
      for (j= index-1; a[j]! = '\0' ; j++, i++)
        c[i]= a[j];
        c[i] = '\0';
      printf ("%s", c);
    }
}
```

```
O/P :-
Enter main string :
  abcdhijk
Enter Sub string :
  efg
Enter index to insert:
  5
abcdefghijk
```

```
abcdhijk
efg
5
strupy
```

STRING HANDLING FUNCTION :-

include < string. h >

predefined( ) = strlen ( ); strrev ( ); strcpy ( ); strncpy ( ); strcat( ); strncat ( ); strcmpi ( )

Ex:- 1. To find string length

```
main ( )
{ char a [20];
  int n;
  printf ("enter a string: ");
  scanf ("%s", a);
  n = strlen (a);
  printf ("%d", n);
}
```

o/p:-
Enter string: happy
Befor reversing ; happy
After reversing: yppah

2. To reverse a String

```
main ( )
{
  char a [20];
  printf ("enter a string:");
  scanf ("%s", a);
  strrev (a);
  printf ("%s", a);
}
```

3. To check the given string is palindrome or not ;-

```
→ main ( )
{
  char a [20], b [20];
  int n;
  printf ("enter a string");
  scanf ("%s", a);
  strcpy (b, a);
  strrev (b);
  n = strcmpi (a, b);
    if (n == 0)
    printf ("Palindrome");
    else
    printf ("Not Palindrome");
}
```

4. To find sub-string in a main string :-

```
-> main ( )
{
  char a [20], b [20];
  char *n;
  printf ("enter a string");
  scanf ("%s", a);
  printf ("enter sub
            string");
  scanf ("%s", b);
  n = strstr (a, b);
  printf ("%d", n - a);
}
```

EXAMPLES =>

1. strlen (): Returns lengthof a string

```
-> main ( )
   {
     char a[100];
     printf (" enter a string : ");
     scanf (" %s ", a);
     int length = strlen(~~str~~ a);
     printf ("lengthof string : %d\n", length);
   }
```

2. strrev(): - Reverse a string

```
-> main ( )
   {
     char a [100];
     printf (" Enter a string : ");
     scanf (" %s ", a);
     printf (" Reverse %s \n", strrev (a));
   }
```

3. strcpy(): - Copy a string

```
-> main( )
   {
     char a [100], b [100];
     printf (" Enter a string : ");
     scanf (" %s ", a);
     strcpy (b, a);
     puts (b);
   }
```

4. strcat;.- String concatination

-> main ()

```
{
  char a[100] , b[100];
  printf ("Enter a string:");
  scanf ("%s", a);
  printf ("Enter b string: ");
  scanf ("%s", b);
  strcat (a, b);
  printf (" String after concatination is: %s", a);
}
```

5. strcmpi ();- Enables a case-insensitive comparision of 2 strings.

returns '0' if the given 2 strings are same

returns '-ve' if the length of str1 < length of str2

returns '+ve' if the length of str1 > length of str2

-> main ( )

```
{
  char a [100], b[100];
  printf ("enter a string: ");
  scanf ("%s", a);
  printf ("enter b string: ");
  scanf ("%s", b);
  int i = strcmpi (a, b);
  printf (" %d", i);
}
```

6. String reverse & concatinate:-

```
-> main ()
{  char a[100], b[100];
   printf (" Enter a string: ");
   Scanf ("%s", a);
   printf ("Enter b string: ");
   Scanf ("%s", b);
   strrev (a);
   strrev (b);
   strcat (a, b);
   printf ("%s", a);
}
```

O/P:-

```
enter main string : hello
enter sub string : World
enter indexto insert : 3
   heworldllo.
```

Q:- WAP to insert sub-string into a main-string from starting position 'm' [index]:- using String handling function = strlen(); strncpy(); strcat()

```
-> int main ( )
  {
   char a[40], b[20], c[100];
   int i, count, index;
   printf ("Enter a string : ");
   scanf ("%s", a);
   printf ("Enter b string : ");
   scanf ("%s", b);
   printf ("Enter index to insert: ");
   scanf ("%d", & index );

   count = strlen (a);
  if (index <= 0 || index > count + 1)
  {
      printf ("Not possible \n");
  }
  else
  {
      strncpy (c, a, index - 1);   //copy before the index
      c[ index - 1 ]= '\0';        //NULL terminates the string

      strcat (c, b);           //concat sub-string b into c

      strcat (c, a + index - 1);
      printf ("%s \n", c);
  }
  return 0;
}
```

O/P:-

```
Enter a string: abcdhijk
Enter b string: efg
Enter index to insert: 5
abcdefghijk
```