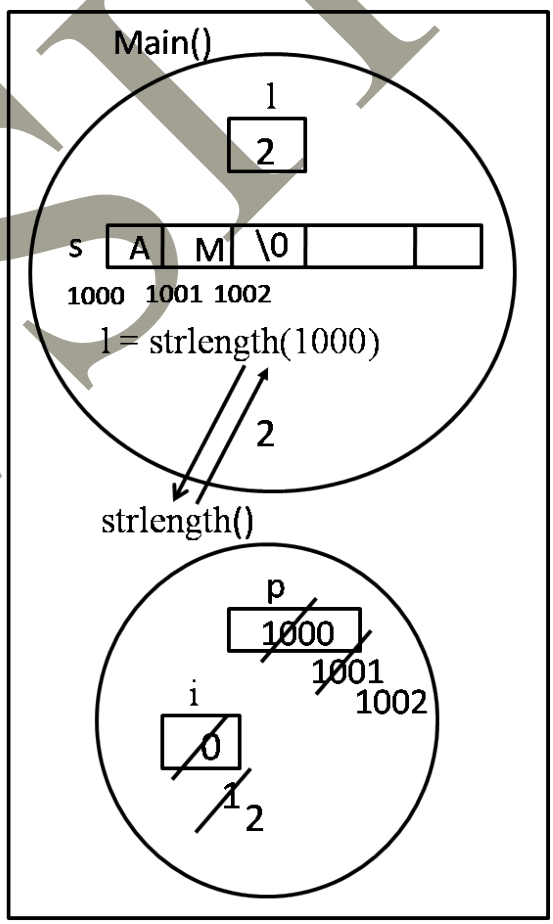
In the last lecture we have discussed POINTERS and ARRAYS(1D array) with examples.

Therefore, we are **now** ready to write **user-defined functions** using pointers for **implementing some string-handling functions** supported by C language.

Problem 1**:**

Write **a user-defined function to find the length of a given string. Also call it from main.**

```
#include <stdio.h>
int main()
{
  char s[80];
  int l;
  int strlength(char *);
  printf("\nEnter a string : ");
  gets(s);
  l=strlength(s);
  printf("\nLength of given string is %d",l);
  return(0);
}
int strlength(char *p)
{
  int i=0;
  while(*p!='\0')
  {
    i++;
    p++;
  }
  return(i);
}
```



CSE/SIT

Problem 2:

Write a user-defined function to copy a given string. Also call it from main.

[**NOTE**: We are NOT allowed to use strcpy() function rather we have to implement it]

```
#include <stdio.h>
int main()
{
        char s[80],t[80];
        void strcopy(char *,char *);
        printf("\nEnter a string : ");
        gets(s);
        strcopy(t, s);
        printf("\nCopy of the given string is %s", t);
        return(0);
 }
void strcopy(char *t, char * s)
{
        while(*s!='\0')
        {
                *t=*s;
                s++;
                t++;
        }
        *t='\0';
 }
```

Problem 3:

Write a user-defined function to concatenate two given strings. Call it from main.

[**NOTE**: We are NOT allowed to use strcat() function rather we have to implement it]

```
#include <stdio.h>
int main()
{
        char s[80], t[80];
        void xstrcat(char *, char *);
        printf("\nEnter a string : ");
        gets(s);
        printf("\nEnter a string : ");
        gets(t);
        xstrcat(s, t);
        printf("\nConcatenated string is %s", s);
        return(0);
 }
```

CSE/SIT

```
void xstrcat(char *s, char * t)
{
        while(*s!='\0')
                s++;
        while(*t!='\0')
        {
                *s=*t;
                s++;
                t++;
         }
        *s='\0';
 }
```

HOMEWORK:
Write a user-defined function to compare two given strings. Also call it from main.

Hint**:**

Here we are not allowed to use strcmp() function rather we have to write our own function to implement strcmp().

To implement strcmp(), we must understand what job it does and how that job is done.

What job it does?

strcmp(s1, s2) compares two strings. It returns zero if s1 & s2 are identical, a negative value if s1 < s2 and a positive value if s1 > s2.

How it is done?

The two strings are compared character by character until there is a mismatch or end of one of the strings is reached, whichever occurs first.
If the **two strings** are **identical** then **it returns zero**. If they are **NOT**, it returns the numeric difference between the ASCII values of the first non-matching pairs of characters.

With this knowledge hopefully you will be able to write your own function that implements strcmp().

CSE/SIT