Title	**Lecture Notes**
Subject	**Programming for Problem Solving**
Topics	**String.**
Lecture Date	**June 2024**
Faculty Name	**Mr. Dilip Kr. Bharti**
Section	**First Year Section- B**

String

In C programming, strings are arrays of characters ending with a null character ('\0'). Unlike many other languages, C does not have a built-in string data type; instead, strings are represented using arrays of type char.

Here are some important concepts related to strings in C:

1. String Initialization

Initialize a string in several ways:

```
#include <stdio.h>

int main() {
   // Using double quotes
   char str1[] = "Hello, World!";

   // Using an array of characters
   char str2[] = {'H', 'e', 'l', 'l', 'o', '\0'};

   // Using a pointer to a string literal
   const char *str3 = "Hello, World!";

   printf("%s\n", str1);
   printf("%s\n", str2);
   printf("%s\n", str3);

   return 0;
}
```

2. String Input/Output

Reading and printing strings can be done using scanf and printf functions, respectively:

```
#include <stdio.h>
```

```
int main() {
   char str[100];

   printf("Enter a string: ");
   scanf("%99s", str); // %99s ensures that no more than 99 characters are read, leaving
space for '\0'

   printf("You entered: %s\n", str);

   return 0;
}
```

3. String Functions

C standard library provides several functions for manipulating strings, which are declared in the <string.h> header file.

a. strlen - Computes the length of a string.

```
#include <stdio.h>
#include <string.h>

int main() {
   char str[] = "Hello, World!";
   int length = strlen(str);

   printf("Length of the string: %d\n", length);

   return 0;
}
```

b. strcpy - Copies one string to another.

```
#include <stdio.h>
#include <string.h>

int main() {
   char src[] = "Hello, World!";
   char dest[50];

   strcpy(dest, src);

   printf("Copied string: %s\n", dest);

   return 0;
}
```

c. strcat - Concatenates two strings.

```
#include <stdio.h>
#include <string.h>

int main() {
   char str1[50] = "Hello";
   char str2[] = ", World!";

   strcat(str1, str2);

   printf("Concatenated string: %s\n", str1);

   return 0;
}
```

d. strcmp - Compares two strings.

```
#include <stdio.h>
#include <string.h>

int main() {
   char str1[] = "Hello";
   char str2[] = "World";

   if (strcmp(str1, str2) == 0) {
      printf("Strings are equal\n");
   } else {
      printf("Strings are not equal\n");
   }

   return 0;
}
```

e. strrev- Reverse a String.

```
#include <stdio.h>
#include <string.h>

int main() {
   char str1[] = "Hello";
   char str2[10];
   strcpy(str2,str1);
   strrev(str2);
   printf("Original String:%s\n",str1);
   printf("Reverse String:%s\n",str2);

   return 0;
}
```

f. strupr- Change in Upper Case.

```
#include <stdio.h>
#include <string.h>

int main() {
   char str1[] = "hello";
   char str2[10];
   strcpy(str2,str1);
   strupr(str2);
   printf("Original String:%s\n",str1);
   printf("Reverse String:%s\n",str2);

   return 0;
}
```

g. strlwr-Change in Lower Case.

```
#include <stdio.h>
#include <string.h>

int main() {
   char str1[] = "HELLO";
   char str2[10];
   strcpy(str2,str1);
   strlwr(str2);
   printf("Original String:%s\n",str1);
   printf("Reverse String:%s\n",str2);

   return 0;
}
```

//write a program to calculate the length of string with using string library function

```
#include<stdio.h>
#include<string.h>   // string header file for strlen function
void main()
{
char name[20];
int len;
printf("enter name\n");
gets(name);
len=strlen(name);
printf("length of the string is \t%d",len);
}
```

//write a program to calculate the length of string without using string library function

```
#include<stdio.h>
void main()
```

```
{
char name[20];
int len=0,i;
printf("enter name\n");
gets(name);
for(i=0; name[i]!='\0';i++)
len++;
printf("length of the string is \t%d",len);
}
```

String handling function

To deal with string we have various string handling function which are present in **"string.h"** header file.

Function	Use
strlen	Finds length of a string
strlwr	Converts a string to lowercase
strupr	Converts a string to uppercase
strcat	Appends one string at the end of another
strncat	Appends first n characters of a string at the end of another
strcpy	Copies a string into another
strncpy	Copies first n characters of one string into another
strcmp	Compares two strings
strncmp	Compares first n characters of two strings
strcmpi	Compares two strings by ignoring the case
stricmp	Compares two strings without regard to case (identical to strcmpi)
strnicmp	Compares first n characters of two strings without regard to case
strdup	Duplicates a string
strchr	Finds first occurrence of a given character in a string
strrchr	Finds last occurrence of a given character in a string
strstr	Finds first occurrence of a given string in another string
strset	Sets all characters of string to a given character
strnset	Sets first n characters of a string to a given character
strrev	Reverses string

//string copy

Syntax:

strcpy(target string, source string)

Here content of source string will copied to target string.

//write a program to copy a string into another string with using string library function

```
#include<stdio.h>
#include<string.h>
void main()
{
char source[20],target[20];
printf("enter string\n");
gets(source);
strcpy(target,source);
printf("The source string is \t");
puts(source);
printf("The target string is \t");
puts(target);
}
```

//write a program to copy a string into another string without using string library function

```
#include<stdio.h>
void main()
{
char source[20],target[20];
int i;
printf("enter string\n");
gets(source);
for(i=0;source[i]!='\0';i++)
{
target[i]=source[i];
}
target[i]='\0';
printf("The source string is \t");
puts(source);
printf("The target string is \t");
puts(target);
}
```

strrev()

Syntax:

strrev(string_name);

//write a program to reverse a string with using string library function

```
#include<stdio.h>
#include<string.h>
void main()
{
char name[20];
printf("enter string\n");
gets(name);
strrev(name);
printf("after reverse \n");
puts(name);
}
```

//write a program to reverse without using string library function

```
#include<stdio.h>
void main()
{
char name[20],temp;
int len=0,i;
printf("enter string\n");
gets(name);
for(i=0;name[i]!='\0';i++)
{
len++;
}
for(i=0;i<len/2;i++)
{
temp=name[i];
name[i]=name[len-1-i];
name[len-1-i]= temp;
}
puts(name);
}
```

<u>strcat()</u>

strcat(target string, source string);

After concatenation: targetstringsourcestring

//Write a c program to concatenate two string with using library function

```
#include<stdio.h>
#include<string.h>
void main()
```

```
{
char first[20],second[20];
printf("enter ist string\n");
gets(first);
printf("enter 2nd string\n");
gets(second);
strcat(first,second);
printf("after concatenation\n");
printf("ist string\t");
puts(first);
printf("2nd string\t");
puts(second);
}
```

//Write a c program to concatenate two string without using library function

```
#include<stdio.h>
void main()
{
char first[20],second[20];
int len=0,i;
printf("enter ist string\n");
gets(first);
printf("enter 2nd string\n");
gets(second);
for(i=0;first[i]!='\0';i++)
{
len++;
}
for(i=0;second[i]!='\0';i++)
{
first[len+i]=second[i];
}
first[len+i]='\0';
printf("after concatenation\n");
printf("ist string\t");
puts(first);
printf("2nd string\t");
puts(second);
}
```

strcmp()

strcmp(target string,source string)

strcmp function return the difference between the ASCII value of first mismatch characters.
It can return +1,-1,0
int l=strcmp(“hello”,”hello everyone”);
Output: -1;

//Write a c program to check whether two strings are identical or not using library.

```
#include<stdio.h>
#include<string.h>
void main()
{
char first[20],second[20];
int d;
printf("enter ist string\t");
gets(first);
printf("enter second string\t");
gets(second);
d=strcmp(first,second);
if(d==0)
printf("identical");
else
printf("not identical");
}
```

//Write a c program to check whether two strings are identical or not without using library funciton

```
#include<stdio.h>
void main()
{
char first[20],second[20];
int d=0,i,a,b;
printf("enter ist string\t");
gets(first);
printf("enter second string\t");
gets(second);
a=strlen(first);
b=strlen(second);
if(a!=b)
d=1;
else
{
   for(i=0;first[i]!='\0'&& second[i]!='\0';i++)
{
if(first[i]!=second[i])
```

```
{
d=first[i]-second[i];
break;
}
}
}
if (d==0)
printf("identical");
else
printf("not identical");
}
```

//Write a c program to check whether given string is palindrome or not using library function.

```
#include<stdio.h>
#include<string.h>
void main()
{
int d;
char first[20],second[20];
printf("enter string\t");
gets(first);
strcpy(second,first);
strrev(second);
d=strcmp(first,second);
if(d==0)
printf("palindrome");
else
printf("not palindrome");
}
```

//Write a c program to check whether given string is palindrome or not without using library function.

```
#include<stdio.h>
void main()
{
   int p=0,i,len=0;;
char first[20],second[20];
printf("enter string\t");
gets(first);
for(i=0;first[i]!='\0';i++)
len++;
```

```
for(i=0;i<len;i++)
{
   second[len-1-i] =first[i];
   }
second[i]='\0';
for(i=0;i<len;i++)
{
   if(first[i]!=second[i])
   {
      p++;
      break;
   }
}
if(p==0)
  printf("palindrome");
else
printf("not palindrome");
}
```

//Write a c program to count total no of uppercase letter, lowercase letter , space , digits and words in a string.

```
void main()
{
char str[100];
int i,u=0,l=0,s=0,d=0,sp=0;
printf("enter string\t");
gets(str);
for(i=0; str[i] !='\0' ;i++)
{
if(str[i] >=65  && str[i]<90)
u++;
else if(str[i]==32)
s=s+1;
else if(str[i]>=97 && str[i]<=122)
l=l+1;
else if(str[i]>=48 && str[i]<=57)
d=d+1;
else
sp=sp+1;
}
printf("\nnumber of uppercase %d",u);
```

```
printf("\nnumber of lowercase %d",l);
printf("\nnumber of digit %d",d);
printf("\nnumber of special character %d",sp);
printf("\nnumber of space %d",s);
}
```