3.2 String Funtions

Objectives

- To understand the fundamentals of Strings and Characters
- To learn how to work with array of Strings
- To learn the usage of String Manipulation Functions
- To learn how to do sorting of strings

Agenda

- Array of Strings
- String Manipulation Functions
    - Explanation with examples
- Sorting of strings

Arrays of Strings

- Since one string is an array of characters, a two dimensional array of characters is an array of strings in which each row is one string.

```
char names[People][Length];
```

- This can be initialized as follows:

```
char month[5][10] = {“January”,
 “February”, “March”, “April”, “May”};
```

String Manipulation Functions

C supports a wide range of functions that manipulate strings. The most common are listed here:

Name	Function
strcpy(*s1*, *s2*)	Copies *s2* into *s1*
strcat(*s1*, *s2*)	Concatenates *s2* onto the end of *s1*
strlen(*s1*)	Returns the length of *s1*
strcmp(*s1*, *s2*)	Returns 0 if *s1* and *s2* are the same; less than 0 if *s1*<*s2*; greater than 0 if *s1*>*s2*
strchr(*s1*, *ch*)	Returns a pointer to the first occurrence of *ch* in *s1*
strstr(*s1*, *s2*)	Returns a pointer to the first occurrence of *s2* in *s1*

strcpy

- strcpy(destinationstring, sourcestring)
- Copies sourcestring into destinationstring
- For example
- strcpy(str, “hello world”); assigns “hello world” to the string str

Example with strcpy

```
#include <stdio.h>
#include <string.h>
   main()
      {
      char x[] = "Example with strcpy";
        char y[25];
        printf("The string in array x is %s \n ", x);
        strcpy(y,x);
        printf("The string in array y is %s \n ", y);
          }
```

strcat

- strcat(destinationstring, sourcestring)

- appends sourcestring to right hand side of destinationstring

- For example if str had value “a big ”
- strcat(str, “hello world”); appends “hello world” to the string “a big ” to get
- “ a big hello world”

Example with strcat

```
#include <stdio.h>
#include <string.h>
   main()
       {
       char x[] = "Example with strcat";
       char y[]= "which stands for string concatenation";
          printf("The string in array x is %s \n ", x);
          strcat(x,y);
           printf("The string in array x is %s \n ", x);
           }
```

strcmp

- strcmp(stringa, stringb)
- Compares stringa and stringb alphabetically
- Returns a negative value if stringa precedes stringb alphabetically
- Returns a positive value if stringb precedes stringa alphabetically
- Returns 0 if they are equal
- Note lowercase characters are greater than Uppercase

Example with strcmp

```
#include <stdio.h>
#include <string.h>
   main()
       {
      char x[] = "cat";
      char y[]= "cat";
       char z[]= "dog";
       if (strcmp(x,y) == 0)
         printf("The string in array x  %s is equal to that in %s \n ", x,y);
```

(Cntd in next slide)

Example with strcmp (Cntd)

```
if (strcmp(x,z) != 0)
        {printf("The string in array x  %s is not equal to that in z %s \n
   ",
         x,z);
   if (strcmp(x,z) < 0)
      printf("The string in array x  %s precedes that in z %s \n ", x,z);
   else
     printf("The string in array z  %s precedes that in x %s \n ", z,x);
        }
else
       printf( "they are equal");
         }
```

strlen

- strlen(str) returns length of string excluding null character
- strlen(“tttt”) = 4 not 5 since \0 not counted

Vowels Example with strlen

```
#include <stdio.h>
#include <string.h>
   main()
       {
       int i, count;
      char x[] = "tommy tucket took a tiny ticket ";
       count = 0;
       for (i = 0; i < strlen(x);i++)
           {
            if
   ((x[i]=='a')||(x[i]=='e')||(x[i]=='I')||(x[i]=='o')||(x[i]=='u'))
            count++;
           }
      printf("The number of vowels's in   %s is %d \n ", x,count);
       }
```

String Manipulation Functions - Predict the output

```
#include <stdio.h>
#include <string.h>
int main(void)
{
char s1[80], s2[80];
gets(s1);
gets (s2);
printf("lengths: %d %d\n", strlen(s1), strlen(s2));
```

String Manipulation Functions – Predict the output (Cntd)

```
if(!strcmp(s1, s2)) printf("The strings are equal\n");
strcat(s1, s2);
printf ("%s\n", s1);
strcpy(s1, "This is a test.\n");
printf(s1);
if(strchr("hello", 'e')) printf("e is in hello\n");
if(strstr("hi there", "hi")) printf("found hi");
return 0;
}
```

If you run this program and enter the strings "**hello" and "hello", the output is (next slide)**

String Manipulation Functions – Example Program output

```
lengths: 5 5
The strings are equal
hellohello
This is a test.
e is in hello
found hi
```

Program for sorting of strings

```
#include<stdio.h>
int main(){
  int i,j,n;
  char str[20][20],temp[20];
  puts("Enter the no. of string to be sorted");
  scanf("%d",&n);
  for(i=0;i<=n;i++)
      gets(str[i]);
  for(i=0;i<=n;i++)
      for(j=i+1;j<=n;j++)
```

Program for sorting of strings (Cntd)

```
{
        if(strcmp(str[i],str[j])>0){
           strcpy(temp,str[i]);
           strcpy(str[i],str[j]);
           strcpy(str[j],temp);
        }
    }
printf("The sorted string\n");
 for(i=0;i<=n;i++)
    puts(str[i]);
 return 0;
}
```

Summary

- Learned about array of strings
- String manipulation functions with examples
- Learned how to do sorting of strings