Unit 6: Character Arrays and Strings

- It is array of characters.
- Collection of characters enclosed in double inverted comma is called string.
- In C, string is terminated with '\0' that is null.

One Dimensional (1-D) Character Array or 1 D String:

1. **Declaration of Strings**
   - String is nothing but one dimensional character array. We can declare string using 'char' datatype.

     Syntax:

     char str_name[20];

     In above syntax,

     Str_name is string name & it can hold name up to 20 characters.
2. **Initialization of Strings.**
   - Similar to array, we can initialize string

     Syntax 1:

     char str_name={all_chracters_enclosed_in_single_inverted_comma}

     Syntax 2:

     char str_name="string in double inverted comma"

   - **Example, showing different ways of initializing strings.**

     ```
     char name[10]={'D','h','a','n','a','n','j','a','y','\0'};
     char name[ ]={'D','h','a','n','a','n','j','a','y','\0'};
     char name[10]="Dhananjay";
     char name[ ]="Dhananjay";
     char *name="Dhananjay";
     ```

   - '\0' is null character, will be added automatically if enough space is available.
   - Consider following 1D String initialization,

     char nm[10]="Dennis",

     for above String declaration following contiguous memeory block will be allocated & it will be terminated with '\0'

D	e	n	n	i	s	\0
nm[0]	nm[1]	nm[2]	nm[3]	nm[4]	nm[5]	nm[6]

3. **Reading Strings.**
   - There are different ways to read strings.
   - While reading string '&' is not mandatory.
   - Suppose we have string declared as follows & we wish to read that.
     char name[10];

   1. **scanf() with %s format specifier.**

      scanf("%s",name);

      scanf() will wait for enter key & afterward it will read string up to space is encountered, & will skip remaining.

   2. **scanf() with scanset.**

      scanf("%[a-z]",name);
      above scanf() will read string containing only lowercase alphabets, if uppercase or any other character is encounter then it will skip the reading, & read only lowercase string.

   3. **scanf() to single line of a string.**

      scanf("%[^\n]",name);
      above scanf() will read the string till new line is encountered, it also allows spaces too.

   4. **scanf() to read multiple lines as a string.**

      scanf("%[^~]",name);

      Above scanf() will read string till '~' is encountered.

   5. **gets()**
      - it is best method to read string & we can read string up to new line character is encounterd.

      Example:-

      gets(name);

   6. **getchar()**
      - we can also use getchar() to read one character at a time & assign that to string, using iterative statements like 'while'.

4. Displaying Strings.

C have two functions to display string 1. printf() & 2. puts().

1. **printf()**

   example

   i. printf("%s",name);

   ii. printf("Enter your name");

2. **puts()**

   example

   i. puts(name);

   ii. puts("Enter your name");

String library functions or String manipulation functions

- All string manipulation functions are available in <string.h> header file.

strlen( )

- Finds number of characters in string excluding null character.
- Return type is int.
- Syntax:
  int_var= strlen(str_name);
- Example:
  1. L=strlrn("Dhananjay");

  Now, value of integer variable L is 9.
  2. L=strlen(name);

  Now, here name is string variable.

Example:

```
#include<string.h>
main(){
int L;
char name[]="Dhananjay";
L=strlen(name);
printf("String length is %d",L);
}
```

Output:-

String length is 9.

strcmp( )

- Compares two strings & returns zero if both are same, otherwise non zero value.
- Syntax:
  strcmp(str1,str2);
- Example:
  Strcmp(s1,s2)

Example:

```
#include<string.h>
main(){
char s1[20],s2[20];
int match;
printf("Enter Sting1:");
gets(s1);
printf("Enter Sting2:");
gets(s2);
match=strcmp(s2,s1);
if(match==0)
printf("Both String are equals");
else
```

	printf("Both Strings are not same"); } **Output:-** Enter String1:Shri Enter String2:Shri Both Strings are equals.

strncmp( )
- Compares two strings up to given integer constant(3rd argument) & returns zero if both are same upto that, otherwise non zero will be return.
- Syntax:
strncmp(str1,str2,int_const);
- Example:
strncmp(s1,s2,7)
	Example:
	#include<string.h>
	main(){
	char s1[20],s2[20];
	int match;
	printf("Enter Sting1:");
	gets(s1);
	printf("Enter Sting2:");
	gets(s2);
	match=strncmp(s2,s1,7);
	if(match==0)
	printf("Both String are equals");
	else
	printf("Both Strings are not same");
	}
	Output:-
	Enter String1:Chandrashekhar
	Enter String2:Chandra
	Both Strings are equals.

strcmpi( )
- Compares two strings & returns zero if both are same, otherwise non zero value by ignoring case
- Syntax:
strcmpi(str1,str2);
- Example:
strcmpi(s1,s2)
	Example:
	#include<string.h>
	main(){
	char s1[20],s2[20];
	int match;
	printf("Enter Sting1:");
	gets(s1);
	printf("Enter Sting2:");
	gets(s2);
	match=strcmpi(s2,s1);
	if(match==0)
	printf("Both String are equals");
	else
	printf("Both Strings are not same");
	}
	Output:-
	Enter String1:SHRI
	Enter String2:Shri
	Both Strings are equals.

strcat( )

- Function concatenate given string as a single string & it returns a resultant into the first.
- Syntax:
  strcat(str1,str2);
- Example:
  strcat(s1,s2);

Example:

```
#include<string.h>
main(){
char s1[20],s2[20];
printf("Enter first Sting:");
gets(s1);
printf("Enter second String:");
gets(s2);
strcat(s1,s2)
printf("Concatenated String is:%s",s1);
}
```

Output:-

Enter first String:Dhanan
Enter second String:ajay
Concatinated String is: Dhananjay

strncat( )

- Function concatenate given string as a single string upto specified integer value(3^{rd} argument) & it returns a resultant into the first.
- Syntax:
  strncat(str1,str2,int_const);
- Example:
  strncat(s1,s2,3);

Example:

```
#include<string.h>
main(){
char s1[20],s2[20];
printf("Enter first Sting:");
gets(s1);
printf("Enter second String:");
gets(s2);
strncat(s1,s2,3)
printf("Concatenated String is:%s",s1);
}
```

Output:-

Enter first String:Gaja
Enter second String:nandakishor
Concatinated String is: Gajanan

strcpy( )

- Copies one string contents into another string.
- Syntax:
  strcpy(str1,str2);
- Example:
  1. strcpy(s1,s2);
     In above, example contents of s2 will be copied into s1.
  2. Strcpy(name,"Dhananjay");
     In above, example "Dhananjay" will be copied into name.

Example:

```
#include<string.h>
main(){
char s1[20],s2[20];
printf("Enter Sting:");
gets(s1);
strcpy(s2,s1);
printf("Copied String is:%s",s2);
}
```

Output:-

Enter String:Shri
Copied String is:Shri

strncpy( )

- Copies one string contents into another string up to given length with third argument.
- Syntax:
  strncpy(str1,str2,int_const);
- Example:
  3. strncpy(s1,s2,2);
     In above, example contents of s2 will be copied into s1.
  4. Strcpy(name,"Omsai");
     In above, example "Om" will be copied into name.

Example:

```
#include<string.h>
main(){
char s1[20],s2[20];
printf("Enter Sting:");
gets(s1);
strncpy(s2,s1,2);
printf("Copied String is:%s",s2);
}
```

Output:-

Enter String:Omsai
Copied String is:Om

strrev( )

- This function returns reverse of a string into the same string.
- Syntax:
  strrev(str_name);
- Example:
  strrev(name);

Example:

```
#include<string.h>
main(){
char name[20];
printf("Enter name:");
gets(name);
strrev(name);
printf("Reverse is:%s",name);
}
```

Output:-

Enter String:rama
Reverse is:amar

strlwr( )

- the function converts the uppercase string into lowercase string.
- Syntax:
  strlwr(str)
- Example:
  strlwr(name)

Example:

```
#include<string.h>
main(){
char name[20];
printf("Enter name:");
gets(name);
strlwr(name);
printf("Lowercase is:%s",name);
}
```

Output:-

Enter name:DHANANJAY
Lowercase is:dhananjay

strupr( )

- the function converts the lowercase string into uppercase string.
- Syntax:
  strupr(str)
- Example:

Example:

```
#include<string.h>
main(){
char name[20];
printf("Enter name:");
gets(name);
strlwr(name);
```

strupr(name)	printf("Uppercase is:%s",name);
	}
	Output:-
	Enter name:dhananjay
	Uppercase is:DHANANJAY

strset( )
- The function set all characters in the string 'str' to the character 'ch'.
- Syntax:
strset(str,ch);
- Example:
strset(pswd,'*')
	Example:
	#include<string.h>
	main(){
	char pswd[20];
	printf("Enter password:");
	gets(pswd);
	strset(pswd,'*');
	printf("Password is:%s",pswd);
	}
	Output:-
	Enter password:omsairam
	Password is:********

strstr( )
- Finds first occurrence of a string in another string & copies that substring with remaining string to resultant
- Syntax:-
str_name=strstr(str1,str2);
- Example
- s3=strstr(s1,s2);
	Example:
	#include<string.h>
	main(){
	char s1[ ]="omsairam";
	char s2[ ]="ram", *s3;
	s3=strstr(s1,s2)
	printf("Substring is:%s",pswd);
	}
	Output:-
	Substring is:ram

strchr( )
- it scans a string for first occurrence of a given character in string.
- If character is found strchr() returns its address.
- Syntax:
strchr(str,ch);
- Example:
strchr(s,'r');
	Example:
	#include<string.h>
	main(){
	char s1[ ]="omsairam", *p,ch='r'
	p=strstr(s1,ch);
	if(p)
	printf("%c is found",ch)
	else
	printf("%c is not found",ch);
	}
	Output:-
	r is found

Programming Examples on Strings

1. **Write a program in c by using strncat( ) to concatenates the first 4 characters of string str2 to str1 where str1=”Programming” and str2=”Language”.**

```
#include<stdio.h>
#include<string.h>
main(){
        char str1[]=”Programming”;
        char str2[]=”Language”;
        /* Concatenating first 4 characters from str2 to str1*/
        strncat(str1,str2,4)
        printf(“After concatenating first 4 characters string is: %s”,str1);
}
```

Output:-

After concatenating first 4 characters string is: ProgrammingLang

2. **Write a program by using strncat() to concatenate the first 4 characters of a string str2 to str1, where str1=”Best of ” and str2=”Luck to all”**

```
#include<stdio.h>
#include<string.h>
main(){
        char str1[ ]=”Best of”;
        char str2[ ]=”Luck to all”;
        /* Concatenating first 4 characters from str2 to str1*/
        strncat(str1,str2,4);
        printf(“After concatenating first 4 characters string is: %s”,str1);

}
```

Output:-

After concatenating first 4 characters string is: Best ofLuck

3. **Write a program by using strncmp() to compare contents of str1 to str2, where str1=”New Delhi” str2=”NewYork” upto first three position.**

```
#include<stdio.h>
#include<string.h>
main(){
char str1[ ]=”New Delhi”;
char str2[ ]=”New York”;
int match;
```

```
        /* Comparing  first 3 characters from both string */
        match=strncmp(str1,str2,3);
        if(match==0)
        printf("Both strings are equal up to first thee position");
        else
        printf("Strings are not equal upto first three position");

}
```

Output:-
Both strings are equal up to first three positions.

4. **Write a program by using strcpy() to copy contents of str1 to str2 where str1="Mumbai" and str2="New Delhi" before execution of program.**

```
#include<stdio.h>
#include<string.h>
main(){
        char str1[ ]="Mumbai";
        char str2[ ]="New Delhi";
        /* copying contents of str1 to str2 */
        strcpy(str2,str1);
        printf("After copying string is: %s",str2);
}
```

Output:-
After copying string is: Mumbai.

5. **Write a program to find length of string without using standard function.**

```
#include<stdio.h>
#include<string.h>
main(){
        int i,L=0;
        char S[20];
        printf("Enter one string:");
        scanf("%s",S);
                for(i=0;S[i]!='\0';i++){
                        L++:
                }
        printf("Length is %d",L);
}
```

OutPut:-

```
Enter one string: programming
Length is 11.
```

6. Write a program to count characters that appears in string for number of times.

```
#include<stdio.h>
#include<string.h>
main(){
    int cnt=0,i;
    char S[20],ch;
    printf("Enter one string:");
    scanf("%s",S);
    printf("Enter one character to see its appearance:");
    scanf("%c",&ch);
    for(i=0;S[i]!='\0';i++)
        if(S[i]==ch)
            cnt++;
    printf("%c appears %d number of times",ch,cnt);
}
```

Output:-

```
Enter one String:Dhananjay
Enter one character to see its appearance: a
a appears 3 number of times.
```

7. Write a program to convert all characters of a string from lowercase to uppercase, without using standard function.

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

    int i;
    char S[20];
    printf("Enter one string in lowercase :");
    scanf("%s",S);
    for(i=0;S[i]!='\0';i++)
    S[i]=S[i]-32;
    printf("Uppercase string is %s",S);
    }
```

Output:-

```
Enter one String in lowercase:dhananjay
Uppercase string is DHANANJAY
```