

17/01/2022

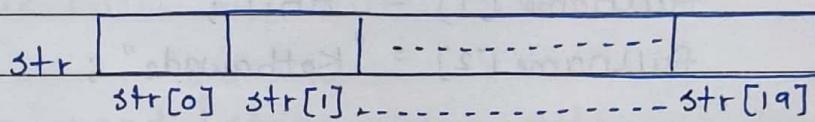
"String"

⇒ A sequence of characters enclosed within " " is called as string. We can implement string in C language using array of characters. String will be automatically end with '\0'. (null character)

- syntax to declare single string:

char varname [size], ...;

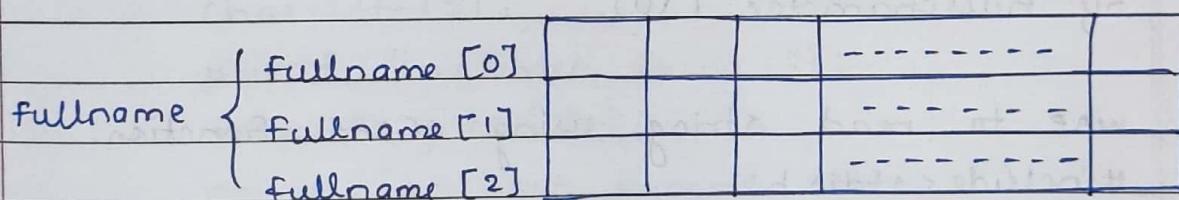
Ex. char str[20];



- syntax to declare multi string:

char varname [size1][size2], ...;

Ex. char fullname [3][20];



- Initialization of string:

① Single string value

1) At the time of declaration:

Ex. char str[5] = {'s', 'h', 't', 'e', 'e'};

OR

char str[5] = " shree";

2) After declaration:

str[0] = 's' ;

str[1] = 'h' ;

str[2] = 't' ;

str[3] = 'e' ;

str[4] = 'e' ;

② Multiple string value.

1) At the time of declaration :

Ex. - char fullname [3][10] = {"Ayush", "Anil",
"Kothawade"};

2) After declaration :

```
char fullname [3][10];
fullname [0] = "Ayush";
fullname [1] = "Anil";
fullname [2] = "Kothawade";
```

* Reading string :

① Using scanf functn :- reading only one word string.
(one word) it means that, it can be replaced whitespace char
by null character (\0)

128. WAP to read string using scanf function.

→ #include <stdio.h>

```
int main ()
{
```

```
    char str[5];
    printf("Enter string");
    printf("\nEnter string :");
    scanf("%s", &str);
    printf("\nYour Entered string : %s", str);
```

}

129. WAP to read ~~st~~ one line string using scanf functn.

→ #include <stdio.h>

```
int main ()
{
```

```

char str[5];
printf("In Enter String:");
scanf("%[^\\n]s", &str);
printf("In Your Entered string : %s", str);
}

```

- ② Using gets function :- Read only one line string.
 (one line) it means that, it will automatically replaced new line char (\n) by null char (\0).

130. ~~128.~~ WAP to read ^{one line} string using gets function.

→ #include <stdio.h>

```

int main()
{
    char str[5];
    printf("In Enter String:");
    scanf gets(str);
    printf("In Your Entered string :");
    puts(str);
}

```

- ③ ~~Re~~Using char by char. Funct :- Read multi line string.
 (multi line) it means that, it will read the string character by character until the given terminating condition.

131. ~~129.~~ WAP to read multiline string using char by char.

→ #include <stdio.h>

```

int main()
{

```

```

    char str[60], ch;
    int i=0;

```

```

printf ("In Enter String :");
ch = getchar();
for (i=0; ch != '$'; i++)
{
    str[i] = ch;
    ch = getchar();
}
printf ("In Your Entered string :");
printf ("Y.S", str);
}

```

* String function in string.h

Function Syntax

Description

1) strlen	int i = strlen(a);	calculate length of string and return an integer value.
2) strrev	strrev(a);	Reverse the string.
3)strupr	strupr(a);	convert string into capital letters.
4) strlwr	strlwr(a);	convert string into small letters.
5) strcpy	strcpy(b,a);	copy contents of a into b
6) strcat	strcat(b,a);	Append contents of one string into another.
7) strcmp	int i = strcmp(a,b);	compare two strings and return a integer value. Value is one of the following: a) Return 0 if both string are b) Return Greater than 0 if 1st string is greater c) Return less than 0 if 2nd string is greater.

- 8) `strcmpi` `len = strcmpi(a, b);` Compare 2 strings same as `strcmp` but it will ignore cases. ~~#~~i means ignore cases.
- 9) `strncpy` `strncpy(b, a, n);` compare 2 strings but only 1st n specified characters, where n is an integer number.
- 10) `strncMPI` `strncMPI(b, a, n);` compare 2 strings but only 1st n specified characters also it will ignore cases.
- 11) `strncpy` `strncpy(b, a, n);` copy only n specified characters.
- 12) `strncat` `strncat(b, a, n);` concatenate only n specified characters.

132

+30. WAP to read multiple string and print it.

→ `#include <stdio.h>``int main ()`

{

`char colourname [10][15];``int i ;``printf ("In Enter colourname : ");``for (i=0 ; i<10 ; i++) ;`

{

`scanf ("%s", & colourname [i]);`

}

`for (i=0 ; i<10 ; i++) ;`

{

`printf ("In colourname [%d] : %s " i , colourname
[i]);`

}

18/01/2022

133. WAP to read a string & print its length.

```
→ #include<stdio.h>
# include <string.h>
int main ()
{
    char str[15];
    int l;
    printf("In Enter String : ");
    gets(str);
    //L = strlen(str);
    printf("In Length of String : %d", strlen(str));
}
```

134. WAP to read string & print it in reverse.

```
→ #include <stdio.h>
# include <string.h>
int main ()
{
    char str [15];
    printf("In Enter String : ");
    gets(str);
    printf("In Reverse String : %s", strrev(str));
}
```

135. WAP to read string & print it in upper case.

```
→ #include <stdio.h>
# include <string.h>
int main ()
{
    char str[15];
    printf("In Enter String : ");
    gets(str);
```

```
printf("In Upper Case : %s", strupr(str));  
}
```

1346. WAP to read a string & print it in lower case.

```
→ #include <stdio.h>  
#include <string.h>  
int main()  
{  
    char str[15];  
    printf("Enter String :");  
    gets(str);  
    printf("In Lower Case : %s", strlwr(str));  
}
```

1357. WAP to read 2 string & copy 2nd string into 1st.

```
→ #include <stdio.h>  
#include <string.h>  
int main()  
{  
    char str1[15], str2[15];  
    printf("Enter Source String :");  
    gets(str1);  
    strcpy(str2, str1);  
    printf("Copied String : %s", str2);  
}
```

138. WAP to read 2 string & concatenate it or join it.

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

```
char str1[15], str2[15];
printf ("In Enter First String : ");
gets (str1);
printf ("In Enter second string : ");
gets (str2);
strcat (str1 " ");
strcat (str1, str2);
printf ("In copied string : %s", str1);
}
```

139. WAP to read 2 string & compare it. (case sensitive).

→

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

```
{ char str1[15], str2[15];
printf("In Enter First string : ");
gets(str1);
printf("In Enter Second string : ");
gets(str2);
int i = strcmp(str1, str2);
if (i == 0)
    printf("In strings are equal");
else
    printf("In strings are not equal");
}
```

140.
138. WAP to read 2 string & compare only n char.

→

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

```
char str1[15], str2[15];
int n;
printf("In Enter First String:");
gets(str1);
printf("In Enter Second String:");
gets(str2);
printf("In Enter no. of characters:");
scanf("%d", &n);
int i = strncmp(str1, str2, n);
if (i == 0)
    printf("In Strings are equal");
else
    printf("In Strings are not equal");
}
```

141.

T39. WAP to read 2 strings & compare only n char. (ignore cases)

```
→ #include <stdio.h>
#include <string.h>
int main()
{
    char str1[15], str2[15];
    int n;
    printf("In Enter First String:");
    gets(str1);
    printf("In Enter second string:");
    gets(str2);
    printf("In Enter no. of characters:");
    scanf("%d", &n);
    int i = strncmp(str1, str2, n);
    if (i == 0)
        printf("In strings are equal");
```

```
    else  
        printf("In strings are not equal");  
    }
```

142.

WAP to read 2 strings and append the content of second string up to n characters.

```
#include < stdio.h >  
#include < string.h >  
int main()  
{  
    char str1[15], str2[15];  
    int n;  
    printf("In Enter First string : ");  
    gets(str1);  
    printf("In Enter Second String : ");  
    gets(str2);  
    printf("In Enter no. of characters : ");  
    scanf("%d", &n);  
    strncat(str1, str2, n);  
    printf("In New string : %s", str1);  
}
```

143.

WAP to read string & check whether it is pallindrome or not.

```
→ #include < stdio.h >  
#include < string.h >  
int main()  
{  
    char str[20], rev[20];  
    printf("In Enter String : ");  
    gets(str);  
    strcpy(rev, str);
```

```

    strrev(rev);
    if (strcmp(str, rev) == 0)
        printf("The string is palindrome");
    else
        printf("The string is not palindrome");
}

```

1442. WA Function length that will return the no. of characters in string.

```

→ #include <stdio.h>
int main()
{
    char str[20];
    int length(char* );
    printf("Enter string : ");
    gets(str);
    printf("Length of string : %d, length(str));"
}

```

```

int length str(char* s)
{
    int i;
    for (i=0, s[i]!='0'; i++);
    return i;
}

```

145.

WA Function reverse that will reverse the given string.

```

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

```

```
char str[20];
void reverse (char* );
printf ("In Enter string : ");
gets (str);
reverse (str);
printf ("\n Reverse string : %. s", str);
}

void reverse (char* s)
{
    char copystr[20];
    int i, j = 0;
    strcpy (copystr, s);
    for (i = strlen (s)-1; i >= 0; i--)
    {
        s[j] = copystr[i];
        j++;
    }
}
```

146.

~~WA~~ Function to read a string & print it in uppercase & lowercase

→

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

```
char str[20];
void upper (char* );
void lower (char* );
printf ("In Enter string : ");
gets (str);
upper (str);
```

```

printf ("In Uppercase string : %s", str);
Lower(str);
printf ("In Lowercase string : %s", str);
}

void upper (char *s)
{
    int i ;
    for (i = 0 ; s[i] != '\0' ; i++)
    {
        s[i] = toupper (s[i]);
    }
}

void lower (char *s)
{
    int i ;
    for (i = 0 ; s[i] != '\0' ; i++)
    {
        s[i] = tolower (s[i]);
    }
}

```

147. WA function copystring that will convert one string into another.

```

→ #include <stdio.h>
#include <string.h>
int main ()
{
    char str1[20], str2[20];
    void copystring(char*, char* );

```

```

printf("In Enter string :");
gets(str1);
copystring(str2, str1);
printf("In Copied String: %s", str2);
}

void copystring(char *s2, char *s1)
{
    int i;
    for (i = 0; s1[i] != '\0'; i++)
    {
        s2[i] = s1[i];
    }
    s2[i] = '\0';
}

```

148. WA function concatenate to append one string into another.

→

```

#include <stdio.h>
#include <string.h>
int main()
{
    char str1[20], str2[20];
    int length(char*);
    void concatenate(char*, char*);
    printf("In Enter First String :");
    gets(str1);
    printf("In Enter Second String :");
    gets(str2);
    concatenate(str1, str2);
    printf("In Concatenated String: %s", str1);
}

int length(char*s)
{

```

```

int i ;
For (i=0 ; s[i]!='\0' ; i++) ;
return i ;
}

void concatenate (char *s1 , char *s2)
{
    int i, j ;
    j = length (s1) ;
    s1 [j++] = ' ' ;
    for (i=0 ; s2[i]!='\0' ; i++) {
        s1 [j] = s2 [i] ;
        j++ ;
    }
    s1 [j] = '\0' ;
}

```

149. WA function compare string that will compare one string to another.

```

→ #include < stdio.h >
#include < string.h >
int main ()
{
    char str1 [20], str2 [20];
    int compare (char *, char *), r ;
    void compare (char *, char *) ;
    printf ("\n Enter First string : ");
    gets (str1) ;
    printf ("\n Enter Second string : ");
    gets (str2) ;
    r = compare (str1, str2) ;
    if (r == 0)

```

```
printf("In Both strings are equal");
else
    printf("In strings are not equal");
}

int compare(char *s1, char *s2) {
    int i, l1, l2;
    l1 = strlen(s1);
    l2 = strlen(s2);
    if (l1 == l2)
    {
        for (i = 0; s1[i] != '\0'; i++)
        {
            if (s1[i] != s2[i])
            {
                return 1;
            }
        }
        return 0;
    }
    else
        return 1;
}
```