Strings

Def: String is a NULL terminated character array. The NULL character is stored to signify the end of the character array.

char str[ ]= " HELLO ";

H	E	L	L	O	\0

str

The length of the string is= 6 (including NULL)

Initialization of strings :

- C allows empty string, but does not allow empty character. Ex: char str[ ]= " " ;
- char s[ ] = { 'H' , 'E' , 'L' , 'L' , 'O' , '\0' };
- char s[10] = " HELLO " ; then rest of the locations filled by NULL character including end of the character.
- char s[3];
  str = "HELLO";

Reading strings :

String can be read in three ways,

1. scanf( )
2. gets( )

scanf( ) :

scanf ("%s", str);

- The problem is that if it encounters blank space, reading is terminated.
- We may also specify the field width to indicate the maximum no of characters that can be read in.
- fflush (stdin); // To clear input buffer.

gets ( ) :

- It holds the starting address of the string.
- It automatically terminated using NULL character.
  gets (str);

Printing strings :

- **printf( ) :**
  - printf ("%s",str);
  - printf(" 5.3s ", str); or printf(" -5.3s ", str);

- **puts () :**
  - Writes a line of input on the screen.
  - Terminates the line with a new line character.
  - It returns EOF(-1) if error occurs and a positive number if successful.

1.

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

int main()
{
    char str[]= "Hello World";
    printf ("\n|%s|",str);
    printf ("\n|%20s|",str);
    printf ("\n|%-20s|",str);
    printf ("\n|%.5s|",str);
    printf ("\n|%6.5s|",str);
    printf ("\n|%-6.5s|",str);

}
```

```
C:\Users\Souvik\Desktop\C Pr...
|Hello World|
|          Hello World|
|Hello World          |
|Hello|
| Hello|
|Hello |
--------------------------------
Process exited after 0.3838 seconds
Press any key to continue . . .
```

2. Input a string and print it.

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

int main()
{
    char str[30];
    printf("Enter a string...");
    gets(str);
    printf("The string is...\n");
    puts(str);
}
```

```
C:\Users\Souvik\Desktop\C Program\string...
Enter a string...India is our motherland
The string is...
India is our motherland

--------------------------------
Process exited after 9.483 seconds with ret
Press any key to continue . . .
```

String Library Functions:

	Functions	Descriptions
1.	strlen( )	To find the length of a string
2.	strcpy( )	To copy one string into another string
3.	strcat( )	To concatenate or add two strings
4.	strcmp( )	To compare two strings
5.	strcmpi( )	To compare two strings with ignoring the case sensitive option
6.	strrev( )	To reverse a string
7.	Atoi( )	Convert string to integer

1. **To find the length of a string with and without using string.h.**

1.cpp | 2.cpp | string length.cpp

```
1  #include<stdio.h>
2  #include<string.h>
3
4  int main()
5  {
6      char str[30];
7      int len,i;
8      printf("Enter a string:  ");
9      gets(str);
10     len=strlen(str);
11     printf("\nLength is: %d",len);
12     /*****without****/
13     for(i=0;str[i]!=NULL;i++);
14     printf("\nLength is: %d",i);
15 }
```

```
C:\Users\Souvik\Desktop\C Pr...
Enter a string:  Hello World

Length is: 11
Length is: 11
--------------------------------

Process exited after 14.53 seconds
Press any key to continue . . .
```

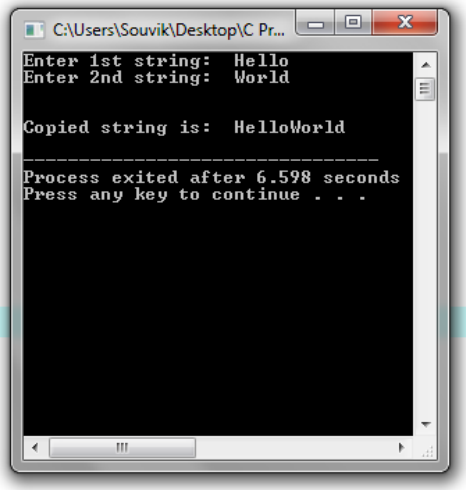
2. **To copy a string into another with and without using string.h.**

```
2  #include<string.h>
3
4  int main()
5  {
6      char str1[30],str2[30];
7      int len,i;
8      printf("Enter a string:  ");
9      gets(str1);
10     strcpy(str2,str1);
11     printf("\nCopied string is: ");
12     puts(str2);
13     /*****without****/
14     for(i=0;str1[i]!=NULL;i++)
15     {
16         str2[i]=str1[i];
17     }
18     str2[i]='\0';
19     printf("\n\nCopied string is:  ");
20     puts(str2);
21 }
```

```
C:\Users\Souvik\Desktop\C Prog...
Enter a string:  Hello World
Copied string is: Hello World

Copied string is:  Hello World

--------------------------------
Process exited after 7.423 seconds w
Press any key to continue . . .
```

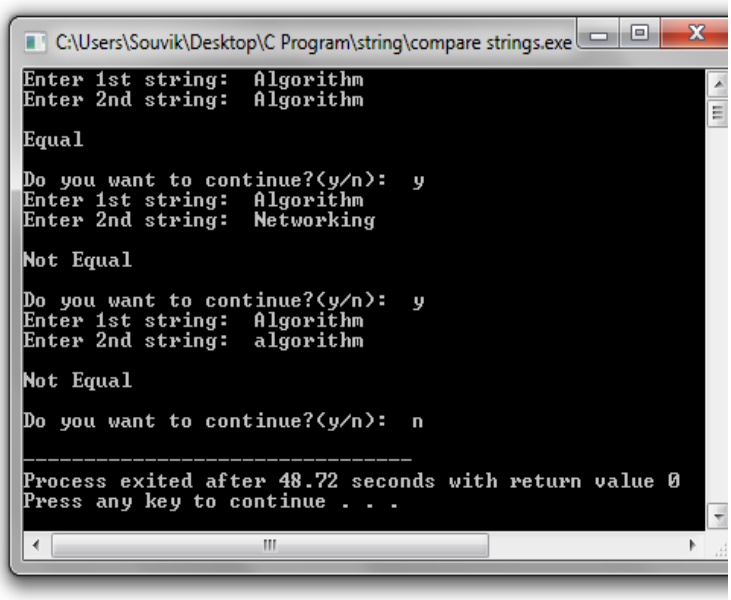
3. **To concatenate two strings with and without using string.h.**

```c
#include<string.h>

int main()
{
    char str1[30],str2[30];
    int len,i,j;
    printf("Enter 1st string:  ");
    gets(str1);
    printf("Enter 2nd string:  ");
    gets(str2);
//  strcat(str1,str2);   //str1=str1+str2
    /*****without****/
    //for(i=0;i<=strlen(str1);i++);
    i=strlen(str1);
    for(j=0;str2[j]!=NULL;j++)
    {
        str1[i]=str2[j];
        i++;
    }
    str1[i]='\0';
    printf("\n\nCopied string is:  ");
    puts(str1);
}
```



4. **To compare two strings with and without using string.h.**

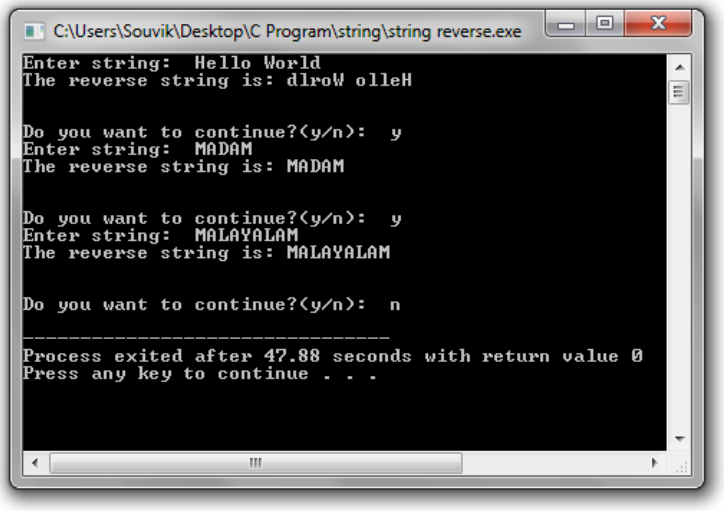
```c
do{
fflush(stdin);
printf("Enter 1st string:  ");
gets(str1);
printf("Enter 2nd string:  ");
gets(str2);
j=strlen(str1);
if(j==strlen(str2))
{
    for(i=0;i<j;i++)
    {
        if(str1[i]!=str2[i])
            {
                flag=1;
                break;
            }
    }
    if(flag==1)
        printf("\nNot Equal");
    else
        printf("\nEqual");
}
else
    printf("\nNot Equal");
fflush(stdin);
printf("\n\nDo you want to continue?(y/n):  ");
//scanf("%c",ans);
ans=getchar();
}while(ans=='y'||ans=='Y');
```



5. To reverse a string with and without using string.h

s

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

int main()
{
    char str1[30],str2[30],ans;
    int i,j,len;
    do{
        j=0;
    fflush(stdin);
    printf("Enter string:  ");
    gets(str1);

    //str2=NULL;
    len=strlen(str1);
    for(i=len-1;i>=0;i--)
    {
        str2[j]=str1[i];
        j++;
    }
    str2[j]=NULL;
    printf("The reverse string is: ");
    puts(str2);

    fflush(stdin);
    printf("\n\nDo you want to continue?(y/n):  ");
    ans=getchar();
    }while(ans=='y'||ans=='Y');

}
```



String Scanset:

A scanset is used to define set of characters which may be read or assigned to the corresponding string. It is defined by placing characters inside the bracket [ ].

Syntax:

1. scanf(" % [---character set---]" , string_name) ;
   Ex: scanf(" %[AEIOU]",str);
   i/p: AIR INDIA (Enter)
   o/p: AI

   i/p: School

   o/p: Garbage

2. scanf(" % [^---character set---]" , string_name) ;
   Ex: scanf(" %[^0123…9]",str); (Used as terminating character)

   i/p: XYX123@gmail.com(Enter)

   o/p: XYZ

3. scanf(" % Value[---character set---]" , string_name) ;

   scanf(" %10[AEIOU]",str); //Max 10 characters will be printed

Substring generation:

1. Extracting a substring from left of a string:

```
Step 1: [Initialize] SET I=0
Step 2: Repeat Step 3 while STR[I] != '\0' AND I<N
Step 3:    SET Substr[I] = STR[I]
Step 4:    SET I = I + 1
        [END OF LOOP]
Step 5: SET Substr[I] ='\0'
Step 6: EXIT
```

2. **Extracting a substring from right of a string:**

```
Step 1: [Initialize] SET I=0, J = Length(STR) - N + 1
Step 2: Repeat Step 3 while STR[J] != '\0'
Step 3:      SET Substr[I] = STR[J]
Step 4:      SET I = I + 1, J = J + 1
        [END OF LOOP]
Step 5: SET Substr[I] ='\0'
Step 6: EXIT
```

3. **Extracting a substring from the middle of a string:**

```
Step 1: [INITIALIZE] Set I = N, J = 0
Step 2: Repeat steps 3 to 6 while str[I] != '0' and N>=0
Step 3:      SET substr[J] = str[I]
Step 4:      SET I = I + 1
Step 5:      SET J = J + 1
Step 6:      SET N = N - 1
        [END of loop]
Step 7: SET substr[J] = '\0'
Step 8: EXIT
```

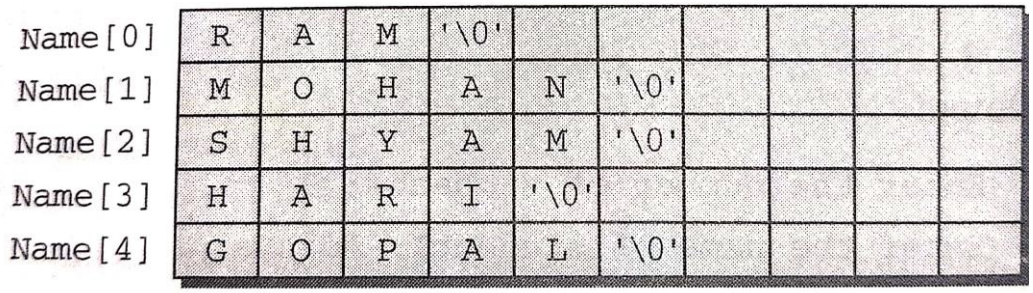
ARRAY OF STRINGS

Basically used for n number of strings.

General syntax:

<data type> <array_name> [row][column];

char name[5][10]={“Ram”, “Mohan”, “Shyam” , “Hari”, “Gopal”};



Name[0]	R	A	M	'\0'					
Name[1]	M	O	H	A	N	'\0'			
Name[2]	S	H	Y	A	M	'\0'			
Name[3]	H	A	R	I	'\0'				
Name[4]	G	O	P	A	L	'\0'			

Sample logic for user input:

```
for( i=0; i<10 ; i++ )
    gets( name[ i ] );
```

Questions:

1. What are the drawbacks for getchar() and scanf()?
2. WAP that reads your name and displays ASCII value corresponding each character.
3. WAP to sort the names of the students.
4. WAP to read a sentence until a ‘#’ is entered. [Hint: use %c]
5. WAP to read a sentence and count number of words in the sentence.
6. WAP to check a string is palindrome or not.
7. WAP to convert the characters of a string from Upper Case to Lower case.
8. WAP to convert the characters of a string from Lower Case to Upper case.
9. Print the following pattern

   C
   CO
   COM
   COMP
   COMPU
   COMPUT
   COMPUTE
   COMPUTER
10. WAP to count number of characters, words and lines in the given text.