7. strcmp() :- STRING COMPARE // ALPHABETECALLY

syntax

int strcmp (string1 , string2);

== 0 string1 == string2

> 0 string1 > string2

< 0 string1 < string2

1. a b c d w a
   a b p a >
   -

2. A B C D
>
   A B C

3. A B C P S T
   A b C S >

-

3. A B C =
A B C

```
--- //  COMPARE  TWO  NAMES
#include<stdio.h>
#include<string.h>
int     main()
{
       char  a[10] , b[10];

printf(" ENTER TWO NAMES \n ");
scanf("%s%s" , a , b);

if( strcmp( a , b ) == 0 )
{
      printf(" BOTH  STRING  ARE  SIMILAR \n ");
}
else    if( strcmp( a , b ) > 0 )
{
 printf(" STRING1  IS GREATER THAN STRING2  \n
                                                 ");
```

t = strcmp(a, b)
if (t == 0)

```
}
else
{
  printf(" STRING2  IS GREATER THAN STRING1  \n ");
}
}
```

// PALINDROME

```
#include<stdio.h>
#include<string.h>
int     main()
 {
              char  a[10] , b[10];

             printf(" ENTER NAME \n ");
              scanf("%s", a);      // ABC

              strcpy ( b , a);

              strrev (b);
```

```
            if( strcmp( a , b ) == 0 )
            {
                printf(" PALINDROME \n ");
            }
  else
            {
               printf(" NOT  PALINDROME \n ");
            }
}
```

8. stricmp() :- STRING compare // alphabet

syntax

stricmp(string1 , string2);

not case senstive

ABC

same

abc

9. gets() :- // INPUT SPECIAL CHARACTERS

syntax
gets (string);

10. puts() :- // PRINT SPECIAL CHARACTERS

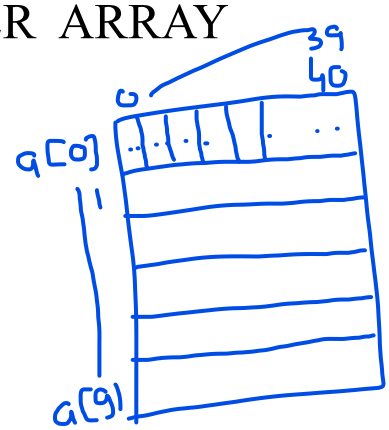
syntax
puts(string);

stdio.h

```
---------------------------------------------------------------------
---- #include<stdio.h>    int   main()
  {
            char   a[40];

          printf(" ENTER NAME \n ");

            gets( a );    // amit kumar

            puts( a );    // amit kumar
  }
---------------------------------------------------------------------
-
```

TWO DIM. CHARACTER ARRAY

char a[10] [40]
ROW COL



---

*/

// INPUT AND PRINT N NAMES

```
#include<stdio.h>
int     main()
{
            char   a[10][40];
            int      i , n;

           printf(" ENTER SIZE \n ");
```

```
        scanf("%d", &n);

       printf(" ENTER NAMES \n ");

        for(  i = 0 ; i < n ; i++ )
         {
               scanf("%s" , a[i]);

       }  // INPUT N NAMES

         printf(" NAMES = \n ");
    for(  i = 0 ; i < n ; i++ )
         {
               printf(" %s \n " , a[i]);
       } // PRINT N  NAMES
}
```

// LINEAR SEARCH OR SEQUENTIAL SEARCH (FOR NAME)

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

```
int    main()
{
       char  a[10][40] , x[40] ;

       int   i , j , n ,t ;

     printf(" ENTER SIZE \n ");

      scanf("%d", &n);

     printf(" ENTER NAMES \n ");

      for( i = 0 ; i < n ; i++ )
       {
          scanf(" %s" , a[i]);

     }  // INPUT N NAMES

        printf(" ENTER NO. TO BE SEARCHED \n");
scanf("%s", x);
t = 0 ;
```

```
        for( i = 0 ; i < n ; i++ )
         {
            if( strcmp(a[i],x ) == 0)
             {
             printf(" FOUND AND POSITION = %d\n", i + 1 );

                t = 1 ; break;
        }    }  // for
if ( t == 0 )
         {        printf(" NOT FOUND \n");     }
          }
}
```

---- n = 4 , t = 0 , x = BB a[ ] = AA, PP , BB, DD

for i = 0 to 3

strcmp(a[i], x) i
= 0 strcmp(AA,BB)
X i = 1 strcmp(PP
,BB) X

i = 2 strcmp(BB ,BB) ---> FOUND AND POSITION = 2 + 1 = 3

t = 1 ; break;

// INPUT N NAMES AND SORT N NAMES USING BUBBLE SORT

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

  int   main()
  {
       char  a[10][40] , t[40] ;
       int   i , j , n  ;

      printf(" ENTER SIZE \n ");
       scanf("%d", &n);

      printf(" ENTER NAMES \n ");

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

```
    {
        scanf(" %s" , a[i]);

    }  // INPUT N NAMES



//  SORTING
    for( i = 0 ; i < n-1 ; i++ )
     {
       for ( j = 0 ; j < n-i-1 ; j++ )
        {
           if ( strcmp( a[j] , a[j+1] ) > 0 )
            {
               strcpy( t ,a[j] );

               strcpy( a[j] , a[j+1] );

               strcpy( a[j+1] , t);
            } // IF
```

```
            } // J
  } // I
      printf(" SORTED NAMES  \n ");

       for( i = 0 ; i < n ; i++)
        {
            printf(" %s \n " , a[i]);
        }
   }
    /*
```

---

i = 0 i = 0 i = 0
j = 0 j = 1 j = 2

PPA	CCB	CCB	CCB	
CCB	PPA	PPA	PPA	1ST ITRATION
TUX	TUX	TUX	ABC	
ABC	ABC	ABC	TUX	

---

i = 1 i = 1
j = 0 j = 1

```
   CCB      CCB   CCB
   PPA      PPA   ABC          2ND ITRATION
   ABC      ABC   PPA
   TUX      TUX   TUX
-------------------------------------------------------------
i = 2
j = 0

   CCB      ABC
   ABC      CCB          3RD  ITRATION
   PPA      PPA
   TUX      TUX

-------------------------------------------------------------
    */
```