for Interview / Written Technical Questions -

String handling & Pointers –

char - single char - it can store only single char
char c = 'A';
char c; c = getch(); - contain/store only one char.

String - group of chars. for implement/use string in C & C++, we have concept of char array. means C & C++ ke inside string store karne ke liye alag se koi data type nahi, for that we need to use char array. but in java we have String class for store String.

String handling - work / operations on String

for String handling in C & C++ we have String.h, contain the String handling functions

**** all string handling functions prefix with str word.**

strcpy() strcat() , strlen() , strupr(), strlwr(), strrev().

for String compare we have - strcmp(), strcmpi(), strncmp(), strncmpi()

every String contain terminator char, that is also known as null char

char nm[10]; - we can store 10 chars string. - Max 10 chars string.
9 chars + 1 string terminate - null char - '\0'

null char - = '\0' or = 0;

Q- null char memory leta hai ya nahi? - ans - always says lete hai.
or
Q- we have char ar[10] - isme kitne char ka string store kar sakte?
Ans- 9 char string + 1 char/index reserved for null char, null char represent '\0'

**** \0 is major use in programming, jaha koi bhi String pe operation perform karna hai without any method for that we need to null char - u have only option of null char for resolve any type of problem.**

char ar[10]; strcpy(ar,"Mayank"); ar = "Mayank";

if we initialize string with = operator, like ar = "Mayank"; that is not valid. - for string initialize we need to use strcpy() function or we can initialize String at declare time or initialize with runtime input.

for string input -
scanf("%s",nm); - input accept but space de diya to end ho gaya
gets(nm); - accept input until enter press - jab tak enter nahi press karoge tab tak ka input accept karega

1st - strcpy(str1,str2) - for string copy.

```
char ar[15];
ar = "Mayank"; - wrong way in C & C++.
strcpy(ar,"Mayank");  now nm is = "Mayank"
```

- Without function initialize the String -

```
ar[0] = 'M';
ar[1] = 'a';
ar[2] = 'y';
ar[3] = 'a';
ar[4] = 'n';
ar[5] = 'k';
ar[6] = '\0';
```

now ar = "Mayank";

in Java no method for string initialize, for String initialize we can use = operator.

For copy one string to another string -

```
int ctr = 0;
while(ar[ctr]!='\0')
{
   arr[ctr] = ar[ctr];
   ctr++;
}
arr[ctr] = '\0'; // insert null char for terminate the string.
```

2nd - strcat(str1,str2) - for String concat

```
   int a = 10;  - NOW WE WANT TO ADD 5 VALUE IN a,
   a = a + 5; / a += 5;   NOW a = 15

ar = "Mayank";  , now we want to concat Kumar in ar.
strcat(ar," Kumar");
ar = "Mayank Kumar";
```

for String concat in Java we can use + or += operator. means no need to use any method for String concat in Java.

means - Java not support the operator overloading concept, but + & += operators default overload milte hai, agar numbers ke saath use karoge to add karega & string ke saath use karenge to concat karega.

3rd - int strlen(str); - return the length of String.

```
ar = "Mayank";
x = strlen(nm);   x = 6 - C & C++
```

```
 // findout length of String in C without using strlen function - own algo

  int ctr = 0;
  while(ar[ctr]!='\0') // while string not empty - ctr increment karo.
    ctr++;
  printf("String length is - %d",ctr);

  ctr = 0;
  while(ar[ctr] != 0)   while(ar[ctr] != '\0')
     ctr++;                 ctr++;
```

4th - strupr(str); - to convert string into upper case

```
    ar = "Mayank";
    strupr(ar);
    ar = "MAYANK";
```

in Java for convert String into upper case we have - toUpperCase() method.

- convert String in Upper case without using any function -

```
    ar = "Mayank";
    int ctr = 0;
    while(ar[ctr]!='\0') // while string null na ho
    {
       if(ar[ctr] > = 97 && ar[ctr] <= 122)
           or
       if(ar[ctr] >= 'a' && ar[ctr] <='z' )
          ar[ctr] = ar[ctr] - 32;
       ctr++;
    }
```

```
// program for input String & display in upper case
void main()
{
    char ar[20];
    int ctr;
    clrscr();
    printf("Plz Enter any String : ");
    scanf("%s",ar);
    ctr = 0;
    while(ar[ctr]!='\0') // while not end of string
    {
       if(ar[ctr]>='a' && ar[ctr]<='z')  // means small a to z hai to
         ar[ctr] = ar[ctr]-32;
       x++;
    }
    printf("\nString in upper case %s",ar);
    getch();
}
```

5th - strlwr(str); - to convert string into lower case

```
      char ar[] = "Mayank";
      strlwr(ar);
```

```
ASCII code -  A - 65 to Z - 90
              a - 97 to z - 122
                 -----------------
                  32        32
```

lower to upper case - -32 / upper to lower case casting - +32

in Java for conver String into lower case we can use toLowerCase() method

- convert String in Lower case without using any function -

```
ar = "Mayank";
ctr = 0;
while(ar[ctr]!='\0') // while string null na ho
{
  if(ar[ctr] > = 65 && ar[ctr] <= 90)
       or
   if(ar[ctr] >= 'A' && ar[ctr] <='Z' )
      ar[ctr] = ar[ctr]+32;
   ctr++;
 }
printf("\nString in Lower case %s",ar);
```

for tOOGLE cASE -

```
ctr = 0;
while(ar[ctr] != 0)   while(ar[ctr] != '\0')
{
   if(ar[ctr] > = 'A' && ar[ctr] <= 'Z')
      ar[ctr] = ar[ctr] + 32;
   else if(ar[ctr] > = 'a' && ar[ctr] <= 'z')
      ar[ctr] = ar[ctr] - 32;
   ctr++;
}
```

5th - strrev() - for string reverse

```
char ar[] = "malayalam";
strrev(ar);
printf("\nString in reverse %s",ar);
```

String reverse without using any function -

```
// loop for calculate the length of String
  ctr = 0;
  while(ar[ctr]!='\0')
     ctr++;

   printf("String in Reverse - ");
   for(i=ctr-1;i>=0;i--)
     printf("%c",ar[i]);
```

```
 String reverse without any function & store in a variable

  ctr = 0;
  while(ar[ctr]!='\0') // loop for count String length.
     ctr ++;

  for(i=0,j=ctr-1;i<ctr;i++,j--); // loop for String reverse
    rev[i] = ar[j];
  rev[i]= '\0';

  ar = Mayank
                                    i = 0     j = 5
  ar[0] = 'M';   ctr = 1     rev[i=0] = ar[j = 5]= 'k'
  ar[1] = 'a';   ctr = 2     rev[i=1] = ar[j = 4]= 'n'
  ar[2] = 'y';   ctr = 3     rev[i=2] = ar[j = 3]= 'a'
  ar[3] = 'a';   ctr = 4     rev[i=3] = ar[j = 2]= 'y'
  ar[4] = 'n';   ctr = 5     rev[i=4] = ar[j = 1]= 'a'
  ar[5] = 'k';   ctr = 6     rev[i=5] = ar[j = 0]= 'M'
  ar[6] = '\0';              rev[i=6] = '\0';

// program for check String is Palindrome or not - with own algo
void main()
{
  char ar[] = "malayalam";
  char arr[10];
  /* - palindrome with functions
  strcpy(arr,ar);  // arr = "malayalam"
  strrev(ar); // convert ar string in reverse
  if(strcmp(ar,arr)==0)
     printf("\n\n ur String is palindrome");
  else
     printf("\n\n Sorry ur String is not palindrome");
 */
 // palindrome without function.

   char str[] = "malayalam";
   int i,len = 0,flag=1;
   clrscr();
   while(str[len] != '\0')
     len++;
   for(i = 0; i <= len/2; i++) // length ke half tak
   {
     if(str[i] != str[len - i - 1])
     {
       flag = 0;
       break;
     }
   }
   if(flag)
     printf("\n %s is a Palindrome String", str);
   else
     printf("\n %s is Not a Palindrome String", str);
   getch();
}
```

```
 malayalam
 012345678

  len = 9;

  compare -
      str[i=0] = str[(len=9)-(i=0)-1 = 8]  compare - m = m
      str[i=1] = str[(len=9)-(i=1)-1 = 7]  compare - a = a
      str[i=2] = str[(len=9)-(i=2)-1 = 6]  compare - l = l
      str[i=3] = str[(len=9)-(i=3)-1 = 5]  compare - a = a
      str[i=4] = str[(len=9)-(i=4)-1 = 4]  compare - y = y

7th - int strcmp(str1,str2) - compare 2 Strings without ignore case - means
case sensitive.

     it return -   0 - if both strings r equals
                 +ve - if 1st string is greater
                 -ve - if 1st string is small.

 ** compare with char by char(with ASCII code) & return the diff of ASCII
    code.

      ASCII code - A - 65 to Z - 90
                   a - 97 to z - 122

     a = "Amit"; b = "Chintu";
     if(strcmp(a,b)>0) - -2 - condition false
     if(strcmp(b,a)>0) - +2 - condition true


Q- what is the return type of strcmp/cmpi/ncmp/ncmpi()

  all functions return type is int

  0 - if both Strings r equals
  +ve - if 1st String is greator
  -ve - if 1st String is smaller

   +ve & -ve - diff of ASCII code.

// compare 2 Strings without any function - with own algo

 - using for loop -
         01234 5                    01234 5
  str1 = "Vikas\0"          str1 = "Vikas\0"   5th index - null char
  str2 = "Vikas\0"          str2 = "Vikas Ji"  5th index - space

   for(i = 0; str1[i] == str2[i] && str1[i] != '\0'; i++);
   if(str1[i] > str2[i])
      printf("\n str1 is Greator than str2");
    else if(str1[i] < str2[i])
      printf("\n str2 is Greator than str1");
    else
      printf("\n str1 is Equal to str2");
```

```
 -- using while loop -

   i=0;
   while(str1[i] == str2[i] && str1[i] == '\0')
     i++;
   if(str1[i] > str2[i])
     printf("\n str1 is Greator than str2");
   else if(str1[i] < str2[i])
     printf("\n str2 is Greator than str1");
   else
     printf("\n str1 is Equal to str2");

// profram for Input 2 Strings & compare both string without any function
void main()
{
   char str1[20],str2[20];
   int i=0;
   clrscr();
   printf("Plz Enter 2 Strings : ");
   scanf("%s %s",str1,str2);
   for(i = 0; str1[i] == str2[i] && str1[i] != '\0'; i++);
   printf("str[%d] = %c, str2[%d] = %c",i,str1[i],i,str2[i]) ;
   if(str1[i] > str2[i])
      printf("\n str1 is Greator than str2");
    else if(str1[i] < str2[i])
      printf("\n str2 is Greator than str1");
    else
      printf("\n str1 is Equal to str2");
   getch();
}
```