Reading and Writing Strings:

Following are the commonly used two methods to input strings. Let the string is declared as : char str[10];

1. scanf("%s",str); 2. gets(str);

Similarly to print strings, we can use 1. printf("%s",str); 2. puts(str);

gets and puts are used only for reading and writing strings; where as, scanf and printf can be used to input/print any type of data. There we have to write the type specifier, such as %s for string, %d for integer, %f for floating point and %c for character. Using gets and puts, only one item can be read or write at a time, where as any number of items can be read or write using scanf or printf.

Accessing individual characters of a string: When a character array is declared, each location holds each character of the string. For example:

char course[5]="POLY"; will be stored as

Characters	P	O	L	Y	\0
Index	0	1	2	3	4

In the above array, course[0] is P, course[1] is O etc. So we can access all the characters using a loop.

For example: for(i=0;i<5;i++)

printf("\n%c",course[i]); will print all the characters line by line.

Reading and Writing single characters: C supports two other functions to read and write single characters. These are **getchar** and **putchar** functions. getchar is used to input a character and putchar is used to print a character. Consider the following example:

```
char ch;
scanf("%c",&ch);        This code is equivalent to      char ch;
printf("%c",ch)                                         ch=getchar();
                                                        putchar(ch);
```

STRING LIBRARY FUNCTIONS

There are various string manipulation functions included in the C Standard Library which are prototyped in <string.h>. The commonly used functions are given below.

1. **strlen :** This function is used to find the length of a string. Length of a string means the number of characters in that string. For example, the length of "POLY" is 4, the length of "MY GOD" is 6 etc.

Example:
```
char place[20]="Kottakkal";
int l;
l=strlen(place);
```

In the above example, the value of l become 9.

2. **strcpy :** This function is used to copy a string to another string.

Example: char s1[15]="Electronics",s2[15];

we cannot directly assign a character array to another array, instead, we have to use the library function strcpy. To copy s1 to s2, we can write: strcpy(s2,s1); where s1 is the source string s2 is the destination string

3. **strcat :** This function is used to concatenate (join) two strings.

For example, char s1[15]="Our",s2[10]="Country";

Then strcat(s1,s2); will concatenate s1 with s2, and s1 now will contain the string "OurCountry"

4. **strcmp :** This function is used to compare two strings. String comparison is done base on the ASCII sequence of the characters in the string. Let s1 and s2 are two character arrays, then r=strcmp(s1,s2); will compare s1 and s2 and the comparison result will be stored in the variable r. The value of r is 0, if s1 is equal to s2. The value of r is positive, if s1 is greater than s2. The value of r is negative, if s1 is less than s2.

For example, Let s1="poly", and s2="exam", then strcmp(s1,s2) will return a positive number. ('p' has larger ASCII code than 'e')

String and Pointers:

C language supports another method to create strings using pointer of type char. For example: char *str = "Hello";

We can also assign the string at the run time. For example,

```
char *str;
str="Hello";
```

The above example is equivalent to char str[6]="Hello"

(Note: if the string is represented as character array, then C does not support copying a string to another by assignment operation, and there we have to use strcpy function). We can use the library functions related with string on the character pointer str.

String related programming questions:

1.Print a string character by character (line by line)
2. Print the characters of a string in reverse order
3. Count number of 'a' in a string
4. Count number of words in a sentence (string)
5. Print a sentence word by word
6. Find the length of a string using strlen and without using strlen
7. Print the given string in a triangular pattern. Example : "WELCOME"
8. Copy a string to another using strcpy and without using strcpy

```
W
WE
WEL
WELC
WELCO
WELCOM
WELCOME
```

9. Find the reverse of a string
10. Check whether the given string is palindrome or not
11.Read first name, middle name, last name and create full name (use strcat)
12. Compare two strings and print "Equal " or "Not equal"
13. Compare two strings and print in alphabetical order

MODULE IV : FUNCTIONS

A function is a sub program written for a particular task. C functions can be classified in to two categories:

1. Library functions. Example: *sqrt, printf, scanf, puts, strlen* etc.
2. User defined functions. Example *main*

The difference between these two categories is that the library functions are ready to use in the language. No need to write by the programmer. But user defined functions are to be written by the programmer (user).

User defined functions: Every C program must have a *main* function to indicate the beginning of program execution. If the size of a program become larger, then it is difficult to debug, test and maintain the program. In such cases, the program is divided into subprograms, and each sub programs do a particular task. Such sub programs are called functions.

Advantages of Functions:

- Facilitates top-down programming
- Easy to debug
- Reduces the program length
- A function may be used several times without repeating the code
- Can be shared by other programs
- Enables data sharing beween functions

C standard Library:
The collection of library functions is called the C standard Library. These are built in functions in C. It performs common mathematical calculations, string manipulations, character manipulations, input/output operations etc. When a C program is compiled, the code for the library functions used in that program is automatically added to the program.

Header files:
The C standard library consists of different category of functions. Most C compilers support the following category:

- input/output operations
- Mathematical operations
- Character manipulations
- String Manipulations
- Storage allocation

The declaration of variables/constants and other identifiers are present in their respective header files, and must be included in the program to access them. For example, If we want to use printf or scanf function, the header file<stdio.h>should be included in the program. The following table shows some commonly used library functions.

Category	**Header file**	**Functions**	**Use**
Mathematical	math.h	sqrt	Find square root
		sin	Find sine value
		pow	Calculates a value raised to a power (Eg: x^n)
Input/Output	stdio.h	scanf	Formatted input
		printf	Formatted output
		gets	String input
Character Manipulations	ctype.h	isalpha	Test for alphabetic character
		isupper	Test for uppercase letter
		tolower	Converts to lower case letter
String manipulations	string.h	strlen	Find string length
		strcat	Concatenate two strings
		strcmp	Compare two strings
Storage Allocation	stdlib.h	calloc	Alloctates a block of memory
		free	Frees a block of memory
		realloc	Reallocates a block of memory