**UNIT III – Functions and pointers**

**PART-A**

1. **What is function and list the advantages.**

A function is a block of code that performs a specific task.

#### Example:

#### int add (int a, int b) // function definition

{

int c;

c=a+b;

return c;

}

**Advantages of Using Functions**

 Code Reusability

 Better readability

 Information hiding

 Easy to debug and test

 Improved Maintainability

1. **Classify function based on arguments and return type with example?**

1.Functions without arguments and without return type Ex: add();

2.Functions with arguments and without return type Ex: add(a,b);

3.Functions without arguments and with return type Ex: c=add();

4.Functions With arguments and with return type Ex: c=add(a,b);

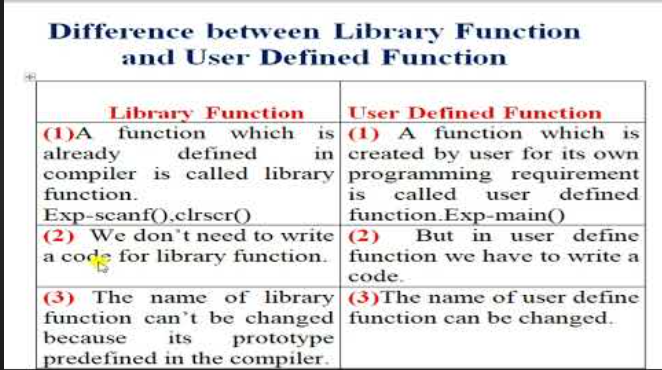
1. **Define the formal and actual arguments with example?**

Arguments which are mentioned in the function call is known as the actual argument. Example: factorial(10); // 10 is actual argument.

Arguments which are mentioned in the definition of the function is called formal arguments.

Example: int factorial(int n) // n is formal argument  
{  
 // write logic here  
}

1. **Difference between predefined and user defined function?**

****

1. **What are the three function elements. Give Syntax with example.**

Every function has the following elements:

1. Function Declaration (Function Prototype) – Tells the compiler about the name of the function, its return type ,and arguments list.
2. Function Definition - Function definition contains executable block of statements.

Syntax:

return\_type function\_name( argument list )

{

:

:

}

1. Function Call - A function can be called simply by using name of the function followed by a list of Actual parameters (Arguments).

Syntax: function\_name( argument list )

1. **Is main () is library function or user defined function? Justify.**

The main () function is a User-defined function.

The main characteristic of the User Defined Function's is that the function definition is entirely given by the programmer.

The code inside the main () function is absolutely different from the other. If it was a built-in function then the source code in the main () function would have remained the same in all the programs.

1. **What are the different ways to pass argument to a function?**

There are two ways by which we can pass the parameters to the functions:

**1. Call by value**

Here the values of the variables are passed by the calling function to the called function.

If any value of the parameter in the called function has to be modified the change will be reflected only in the called function.

**2. Call by reference**

Here, the address of the variables are passed by the calling function to the called function.

The address which is used inside the function is used to access the actual argument used in the call.

If there are any changes made in the parameters, they affect the actual arguments.

1. **What is recursion and what is recursive function?**

Recursion is the process of calling a function itself repeatedly until a particular condition is met.

A function that calls itself directly or indirectly is called a recursive function and such kind of function calls are called recursive calls.

Example:

int Sum(int n)

{

int res = n + Sum(n - 1);

return res;

}

1. **Define pointers with syntax and why it is necessary?**

A pointer is defined as a derived data type that can store the address of other C variables or a memory location.

**Syntax: datatype \* ptr;**

* It is used to access and manipulate the data stored in that memory location.
* It provides dynamic memory allocation
* It is used to implement complex data structures like linked list and trees.

1. **What are \* and & operators means?**
2. **The indirection operator/Dereference operator (\*):** The indirection/ dereference operator is a unary operator that returns the value of the variable present at the given address.

**Syntax: \*ptr= value at address contained in ptr;**

1. **The Address-of operator** (&): It is a unary operator that returns the memory address of its operand which means it stores the address of the variable. It is spelled as the address of the variable.  
   **Syntax: int \*ptr = &x; // the variable ptr stores the address of the variable x.**
2. **What are the valid arithmetic operations that can be performed over pointers.**

* Increment/Decrement of a Pointer
* Addition of integer to a pointer
* Subtraction of integer to a pointer
* Subtracting two pointers of the same type
* Comparison of pointers

1. **What are illegal pointer operations?**

1. Adding two pointer can’t be done

2. Adding float, double or char to a pointer can’t be done.

3. Assigning a non pointer value to a pointer variable can’t be done.

4. Multiplication/subtraction can’t be applied to a pointer

5. bitwise operators can’t be applied to a pointer

1. **Define Null pointer and void pointer with syntax.**

**NULL pointer:**

* A NULL pointer is a special pointer that does not point anywhere.
* It does not hold any address.

|  |  |
| --- | --- |
| **Syntax** | **Example** |
| datatype\* variable\_name=0; (Or)  datatype\* variable\_name=NULL; | int\* a=0; (Or)  int\* a=NULL; |

**VOID pointer**:

* Void pointer is a general pointer that can point to any type of data.
* The pointer used to point different data types is called void data type.

|  |  |
| --- | --- |
| **Syntax** | **Example** |
| void\* variable\_name; | void\* a; |

1. **What do you mean by array of pointers. Give example.**

A pointer array or array of pointers is a homogeneous collection of pointer variables that are references to a memory location.

It is generally used in when we want to point at multiple memory locations of a similar data type in our program or when we want to store multiple strings in an array.

**Syntax: pointer\_type \*array\_name [array\_size];**

* pointer\_type: Type of data the pointer is pointing to.
* array\_name: Name of the array of pointers.
* array\_size: Size of the array of pointers.

1. **What are the advantages and disadvantages of using Pointers.**

**Advantages:**

**** Provides direct access to the memory locations.

 Used to implement data structures like linked list and tree.

 Pointers can be used to return multiple values from a function via arguments.

 It allows dynamic memory allocation.

 It increases the execution speed and reduces the program execution time.

**Disadvantages:**

**** Memory corruption can occur if an incorrect value is provided to pointers.

 Pointers are a little bit complex to understand.

 Pointers are majorly responsible for memory leaks in C.

 Pointers are comparatively slower than variables in C.

 Uninitialized pointers might cause a segmentation fault.

**PART-B**

1. Explain the parameter passing methods in functions with suitable example.
2. Classify functions based on arguments & return values. Supplement your answer with suitable example programs.
3. What is meant by recursion? Explain the concept by implementing binary search using recursion.
4. Explain in detail about the built-in string functions and math functions using suitable example.
5. What do you mean by pointers. Write in detail about the types of pointers, operators used in pointers, Advantages and disadvantages of pointers.
6. With suitable program explain in detail about pointer arithmetic.
7. Write a program to find the factorial of a given number using recursion