Unit 1

History of C:

C is a general-purpose programming language that was created in the early 1970s by **Dennis Ritchie** at Bell Labs. It was developed as an evolution of the B programming language and was designed to be a system programming language for the Unix operating system. C gained popularity due to its simplicity, efficiency, and portability. Over time, it became the foundation for many other programming languages and played a significant role in the development of modern operating systems.

Your paragraph text

Advantages of Structured Programming in C:

Structured programming is a programming paradigm that emphasizes the use of structured control flow constructs like loops and conditionals to improve the clarity and efficiency of code. Advantages of structured programming in C include:

1. Readability: Structured code is easier to read and understand, making it less error-prone and more maintainable.
2. Modularity: It encourages modular code design, which involves breaking down a program into smaller, reusable functions or modules.
3. Reusability: Code in structured programming can often be reused in different parts of a program or in other projects, reducing redundancy and saving time.
4. Debugging: Structured programs are easier to debug since the control flow is well-organized.
5. Maintainability: Changes and updates to structured code are more manageable because of its modular nature.

Files Used in C:

In C programming, different types of files are used:

1. Source Files (.c): These files contain the actual C source code. They are text files that programmers write and edit.

2. Header Files (.h): Header files contain function prototypes, macro definitions, and data structure declarations that are shared across multiple source files. They are included using `#include` directives in source files.

3. Object Files (.o or .obj): Object files are the result of compiling source files. They contain machine code and data in a format that is not yet executable. Multiple object files are linked together to create an executable program.

4. Binary Executable Files: These are the final output files generated after linking one or more object files. They contain the machine code that can be executed directly by the computer program

MCQ forpractice:

1. Compiler converts the .c file into
   a) class file b) obj file c) lib file d) batch file
2. Which of the following are not standard header files in C?
   a) stdio.h b) stdlib.h c) conio.h d) None of the above

Characteristics of C:

C has several key characteristics:

a) Procedural: C follows a procedural programming paradigm, where programs are organized as functions or procedures that manipulate data.
b) Structured: It supports structured programming, promoting the use of functions and control structures like loops and conditionals.
c) Portable: C code is relatively portable across different platforms and compilers, making it suitable for systems programming.
d) Efficient: C provides low-level access to memory and hardware, allowing for efficient code optimization.
e) Extensible: C supports the creation of libraries and user-defined data types, enabling code reuse and extensibility.
f) Middle-Level Language: It combines high-level abstractions with low-level memory manipulation, making it suitable for both system-level and application-level programming.
g) Preprocessor: C includes a preprocessor that allows macros and conditional compilation, enhancing code flexibility.
h) Pointer Support: C has robust pointer support, enabling advanced memory manipulation and data structures.
i) Community and Libraries: It has a strong developer community and a vast collection of libraries, making it easier to find solutions to common programming problems.

Describe different types of storage class in C

Automatic:

- Place of Storage: Stack
- Scope: Local to the block
- Default Value: Garbage value
- Lifetime: Limited to the block execution

Register:

- Place of Storage: CPU register (if possible)
- Scope: Local to the block
- Default Value: Garbage value
- Lifetime: Limited to the block execution

Static:

- Place of Storage: Data segment
- Scope: Local to the block or global (depending on context)
- Default Value: Zero (0)
- Lifetime: Throughout program execution.

Extern:

- Place of Storage: Data segment
- Scope: Global (defined in another file)
- Default Value: Zero (0)
- Lifetime: Throughout program execution

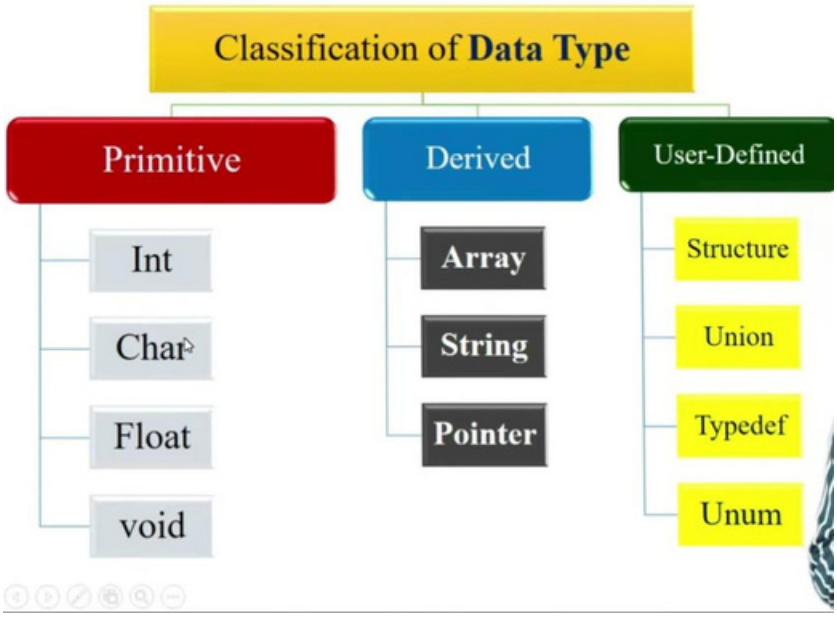
Tokens in C

In C, a token refers to the smallest individual unit of source code that has a specific meaning. C language has several types of tokens, which are used to build programs. Here are some common C tokens:

- **Keywords**: Keywords are reserved words with special meanings in C, such as int, if, while, and return. They cannot be used as identifiers (variable or function names).
- **Identifiers**: Identifiers are names used for variables, functions, and other program elements. They must start with a letter or underscore and can contain letters, digits, and underscores.
- **Constants**: Constants are fixed values that do not change during program execution. These include integer constants (e.g., 42), floating-point constants (e.g., 3.14), and character constants (e.g., 'A').
- **String Literals**: String literals are sequences of characters enclosed in double quotes (e.g., `"Hello, World"`).
- **Operators**: Operators perform various operations, like addition (+), subtraction (-), multiplication (*), and more.
- **Punctuation Symbols**: These include punctuation symbols like semicolons (;), commas (,), and parentheses (()) used to structure the code.

- **Comments**: Comments are not processed by the compiler but provide information for programmers. C supports single-line comments (//) and multi-line comments (/* ... */).
- **Preprocessor Directives**: Preprocessor directives begin with a # symbol and are used to instruct the preprocessor to perform tasks like including header files (#include) or defining constants (`#define`).
- **Whitespace**: Whitespace includes spaces, tabs, and newlines used for formatting and separating code.

Data Types used in C and their Sizes



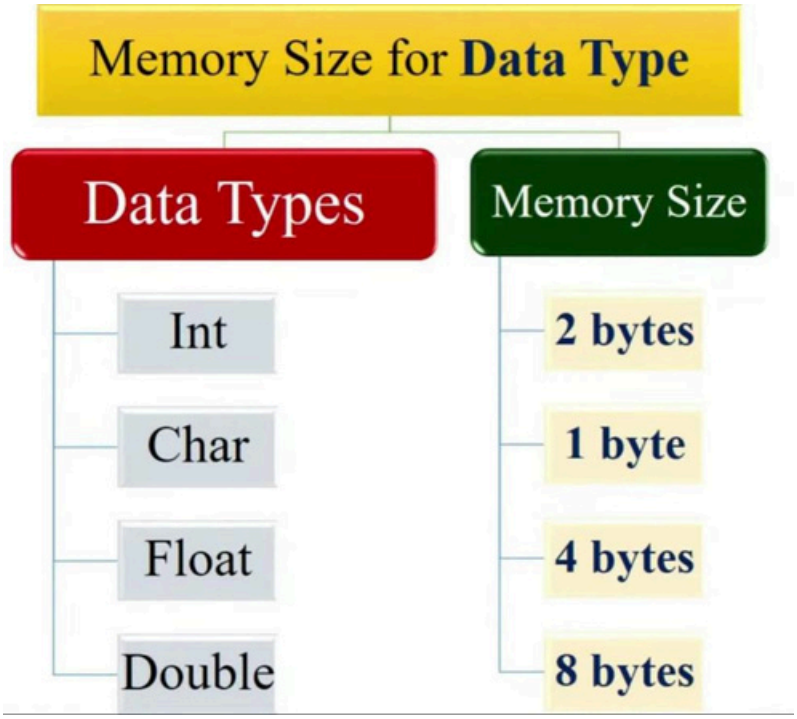
MCQ for Practice:

1. What is the size of the int data type (in bytes) in C?
   a) 4 b) 8 c) 1 d) None of these
2. Which is User-Defined data type?
   a)Array b) int c) Structure d) Float

What is Type Castinggive example: Type casting, also known as type conversion, is the process of changing the data type of a value or an
expression from one data type to another. This is often done to ensure that the data can be used in a specific context

- ❑ In C, there are two main types of type conversion:

i)Implicit Type Conversion (Coercion): It is also known as automatic type conversion; this occurs when the C compiler automatically converts one data type to another without the programmer's intervention.



Example:

```
int x = 5;
float y = 3.14;
float result = x + y; // Implicit type conversion of 'x' to float
```

ii) **Explicit Type Casting**: Explicit type casting is a manual operation where the programmer explicitly specifies the desired data type for a value or expression.

Example:

```
float f = 3.14;
int i = (int)f; // Explicitly converting 'f' to an integer
```

Unit 2: Decision control and looping statement

In the C programming language, there are three primary types of control statements:

1. Decision-making statements:
    - if statement
    - if-else statement
    - switch statement
2. Iteration or looping statements:

    for loop
    while loop
    do-while loop

3. Jump statements:

break statement
continue statement
goto statement
return statement

Decision-making statements:

1. What is if statement?

Decision-making statement that is used to execute a block of code based on the value of the given expression. If the given condition is true then the statements inside the body of “if” will be executed.

Flowchart of if Statement

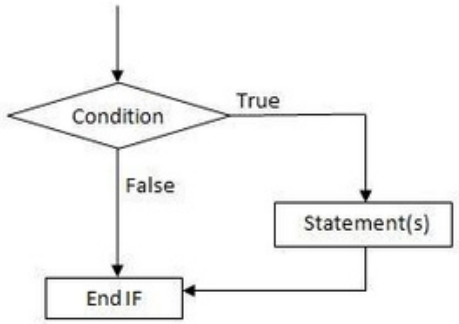


fig: Flowchart for if statement

Example:

```
if(condition) {
   // Code to execute if the condition is true
}
 else {
//Code to execute if the condition is false
}
```

2. What isswitch statement?

A switchstatement allows a program to make decisions based on the value of a variable by choosingdifferent actions for different variable values.

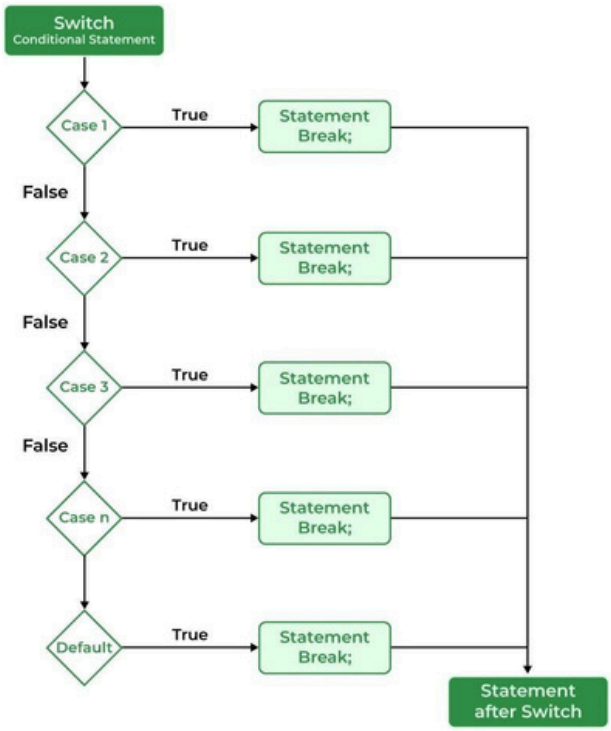
Example:

```
switch (expression) {
  case value1:
    // Code to execute when 'expression' equals 'value1'
    break;

  case value2:
    // Code to execute when 'expression' equals 'value2'
    break;

  // Additional cases can be defined here

  default:
    // Code to execute when 'expression' doesn't match any case
}
```



Looping statements:

Exam Tips:Studythedifferences of each loop like **while loop vs. Do while loop and while loop vs. for loop**

1. What is while loop?

A “while loop” is an entry control loop that allows you to repeatedly execute a block of code as long as a specified condition remains true.

❖The basic structure of a while loop in C:

```
While (condition) {
// Code to execute while the condition is true
}
```

2. Do-while loop?

A “do-while loop” is an exit control loop that ensures a block of code is executed at least once, and then it repeatedly executes the same block of code as long as a specified condition remains true.

❖The basic structure:

```
do {
```

```
    // Code to execute
} while (condition);
```

In a "do-while loop," the condition is checked at the end (after the first iteration). This guarantees that the code block is executed at least once, even if the condition is false from the start.

3. What is For loop?

It allows you to repeatedly execute a block of code for a specific number of iterations.

❑ Structure:

```
(initialization; condition; update) {
  // Code to execute during each iteration
}
```

MCQ tips: Which of the following is an exit-controlled loop?

a) while loop b) for loop c) do-while loop d) None

This type of MCQ often seen in previous year question.

Jump statement:

1. **Break**:weoftencomeacross situations where we want to jump out of a loop instantly without goingbacktotheconditions test. The keyword “break” allows us to do this.

Example:

```
for(inti=0;i<5;i++) {
  if (i == 3) {
    break;//Exitsthe loop when i = 3
  }
  printf("%d ", i);
}
```

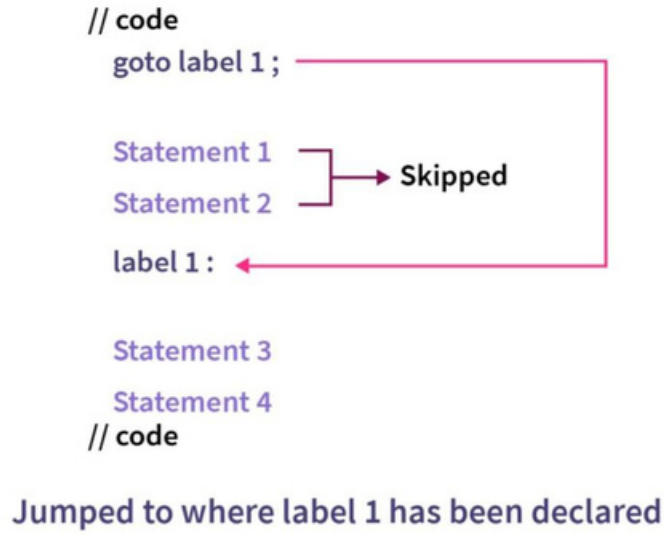
Output: 012

2. **Continue**:Thecontinue statement is used to skip the rest of the code inside a loop for the currentiterationandproceed to the next iteration.

```
for(inti=0;i<5;i++) {
  if (i == 2) {
    continue;//Skips the rest of the loop for i = 2
  }
  printf("%d ", i);
}
```

Output: 0 1 3 4

3. **Go to statement: Goto statement in C is a jump statement that is used to jump from one part of the code to any other part of the code in C. Goto statement helps in altering the normal flow of the program according to our needs.



Advantages of goto Statement in C

- Goto is a jump statement that can alter the normal flow of execution of code. Using the goto statement, you can not only jump to a part of the code below the current flow but also a part above the current flow.
- This also enables goto statements to initiate loops in the program, without using for or while in the code.
- We can also use goto statement when a certain condition is satisfied, and skip some lines of code altogether by moving to another part of the program.
- Goto statements can be helpful when you want to break out of nested loops. Using one goto statement, you can break out of all loops, instead of using multiple break statements.

Disadvantage:

1. Unstructured Code: The use of "goto" can lead to unstructured and hard-to-maintain code. It can make the program's flow difficult to follow, as it allows for non-linear and unpredictable jumps.

2. Readability: Code with with"goto" statements can be challenging to read and understand. It becomes harder to trace the logic and identify the flow of the program

3. Risk of Errors: "goto" statements can introduce the risk of logical errors and unintended program behavior, especially when not used carefully.

4. Inefficiency: Some compilers or interpreters may have difficulty optimizing code that uses "goto" statements, potentially leading to less efficient programs.

5. Alternative Approaches: In most cases, it's possible to achieve the same results with clearer and more maintainable code by using structured programming constructs, such as loops, conditionals, and functions.

Important Question:

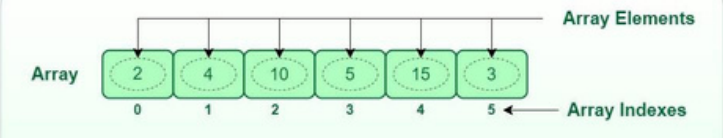
1. What is theuse of goto statement? Give suitable example.

2. Why go tostatement is ignored in coding?

Unit 3

****What is Array:**

An Array is a variable that can store multiple values of the same data type. These values are stored in contiguous memory locations, which means they are placed right next to each other in memory.



This allows for efficient and direct access to individual elements within the array based on their index or position.

- The syntax for declaring an array
  data_type array_name[array_size];
  e.g : int myArray[3];
- The elements of this array are:
  int myArray[0], int myArray[1], int myArray[2];
- Here's how we can initialize an array
  Int myArray[5] = {10, 20, 30, 40, 50};

Characteristic of array:

1. Homogeneous Data: An array is a collection of similar elements.
2. Fixed Size: The size of an array is usually defined at the time of declaration and remains constant.
3. Contiguous Memory: Array elements are stored in adjacent memory locations for efficient access.
4. Zero-Based Indexing: Elements are accessed using indices, with the first element at index 0.
5. Initialization: Arrays can be initialized with specific values during declaration.
6. Direct Access: Elements can be accessed directly using their indices, making arrays efficient for random access.
7. The type of the array may be of any type. It might be int, char, float, etc. But always the index number will be an int.

Types of Array:

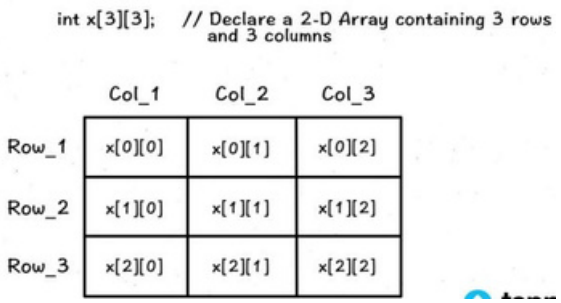
1. One Dimension
2. Two Dimensional
3. Multi Dimensional

What is One Dimensional: A One-Dimensional Array is the simplest form of an Array in which the elements are stored linearly and can be accessed individually by specifying the index value of each element stored in the array.

- Syntax
  data_type array_name[array_size];
- **Initialization**
  we can initialize the element of array in the same way on the ordinary variable.
  The Syntax of initilization of the array is
  data_type array_name[array_size] = {element1, element2, ..., elementN};
  e.g: int myArray[4] = {1, 2, 3, 4};
  e.i. – int myArray[0] = 1;
  int myArray[1] = 2;
  int myArray[2] = 3;
  int myArray[3] = 4;
- **Accessing array elements**
  We can access individual elements in a one-dimensional array by specifying the index of the element we want to access. Array indexes start at 0 for the first element.
  Here's an example of how to access array elements:

  IntmyArray[5] = {10, 20, 30, 40, 50};
  IntfirstElement = myArray[0]; // Access the first element (index 0)

What isTwo-Dimensional Array: A 2D array, is a collection of data elements arranged in a grid-like structurewith rows and columns. Each element in the array is referred to as a cell and can be accessed by its rowand column indexes.



- **Syntax:** **data_type array_name[row_size][column_size];**
- Initialization:

```
int myArray[3][3] = {
   {1,  2,  3},
   {4,  5,  6},
   {7, 8, 9} };
```

What is Multi-dimensional: A multi-dimensional array is an array with more than one dimension, typically represented as a grid of values.

- **Syntax:** **data_type array_name[dim1][dim2][dim3]...[dimN];**
- **Example:** **int my3DArray[2][3][4];**

Structure And Union

What is structure: when we want to represent Collection of data item of different types in a single name then we cannot use array, we use structure.

- **Declaration:**

```
struct structure_name {
   data_type member1;
   data_type member2;
   // ... add more members as needed
};
```

Here's a step-by-step process for initializing values in a structure:

1. Define the Structure:

- Begin by defining the structure's blueprint. Define the structure with its members and their data types.

```
Struct Point {
    Int x;
    Int y;
};
```

2. Create a Structure Variable:

- Declare a variable of the structure type to represent an instance of the structure:

```
Struct Point myPoint;
```

3. Initialize the Structure Members:

- Assign values to the structure's members. You can do this either when declaring the structure variable or afterward.

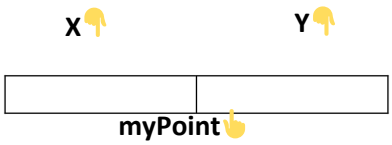
- **Option1:Initializing when Declaring:**
  Providethe values within curly braces `{}` in the order of member declaration.

  ```
  StructPoint myPoint = {3, 4};
  ```

- **Option2:Initializing Members Individually:**
  Assignvalues to each member separately after declaring the structure variable:

  ```
  myPoint.x = 3;
  myPoint.y = 4;
  ```

MemoryAllocationDiagram:



Examtips:Towriteinexam we can follow this pattern:

- **Declaration of structure**
- **Declaration of structure variable**
- **Initialisation of structure valuable**

Declaration of string variable:

In C, there are several main ways to declare & initialize a string variable:

- **Character Arrays:**

Declaring a string as a character array is one of the most common methods.

Example: **char myString[] = "Hello, World!"; // Declares and initializes a character array with a string.**

- **String Literals:**

We can directly assign a string literal to a pointer variable.

Example: **char *myString = “Hello, World!”; // Assigns a string literal to a pointer.**

String handling function from standard library #include<string.h>

strlen(char *str)	Returns the length of string
strcpy(char *s1, char *s2)	Copies the contents of string s2 to string s1
strcat(char *s1, char *s2)	Concat s1 string with s2 and stores the result in s1
strcmp(char *s1, char *s2)	Compares the two strings
strncpy(char *s2, char *s1, int len)	Copy substring of size len starting from s1 character pointer into s2.

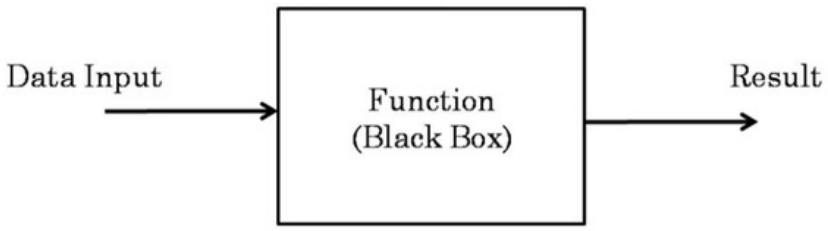
Important Question:

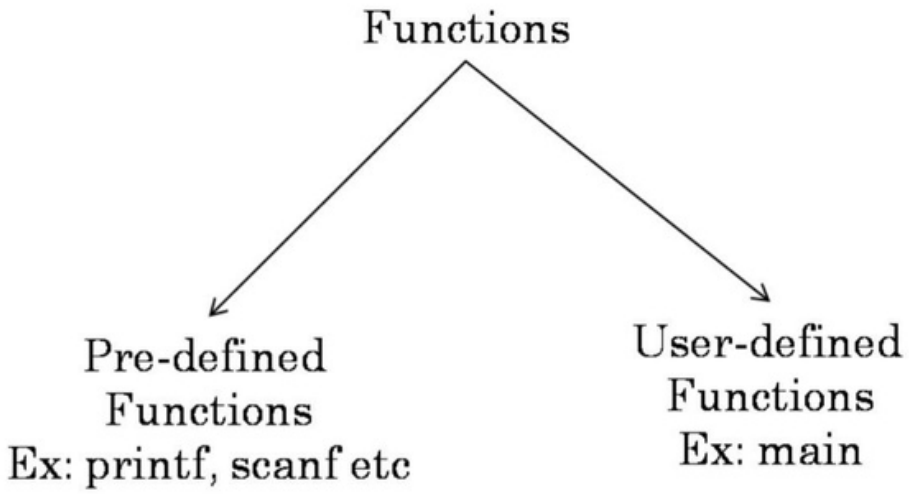
1. **Explain the following string operations using in C (1) stremp () (ii) strcat()**
2. **Explain the function strncpy()**

Unit 4

A function in C is a set of statements that when called perform some specific task. It is the basic building block of a C program.

• It is used to break down and organize a program into manageable parts for better readability and reusability.





1. Predefined function: A predefined function or library function is a function which is already written by another developer. The users generally use these library functions in their own programs for performing the desiredtask.
2. User-Defined Function: A user defined functionis a function which is declared and definedby the userhimself.

Advantage of function

1. Modularity: Functions allow you to break down a complex program into smaller, more manageable modules. Each function can focus on a specific task, making the code easier to understand and maintain.
2. Reusability: Functions can be reused in different parts of a program or even in other programs. This saves time and effort by avoiding redundant code.
3. Readability: Well-named functions make the code more readable and self-explanatory. You can understand the purpose of a function by its name without needing to know the implementation details.

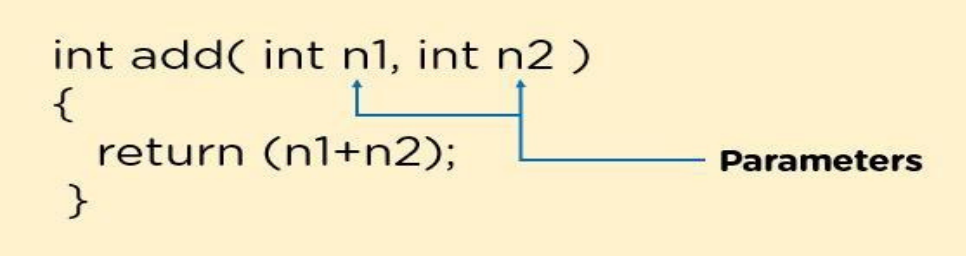
4. Testing: Functions make it easier to test individual parts of a program in isolation. You can test and debug functions independently, which simplifies the debugging process.
5. Scalability: Functions make it easier to extend and modify a program without affecting the entire codebase.
6. Abstraction: Functions provide an abstraction layer, allowing you to hide the implementation details and focus on how to use the function rather than how it works internally.
7. Collaboration: Functions enable multiple programmers to work on different parts of a program concurrently, as long as they adhere to the function interfaces.

Calling function and Called function

1. Calling function: A calling function is a part of the code that triggers another function (known as the called function) with passing the needed data. This allows the called function to perform a specific task.

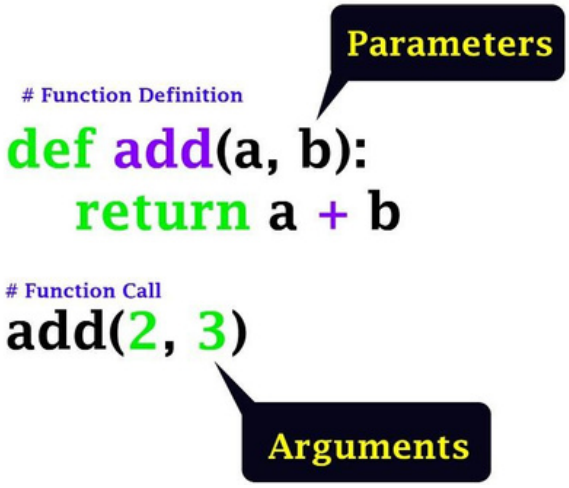
2. Called function: A called function accepts data from the calling part and perform tasks as specified within its code (function body). If there are return statements in the function, it can provide results back to the calling function.

Formal Parameters and Actual Parameters



1. The arguments or parameters which are used in calling function are call actual argument.

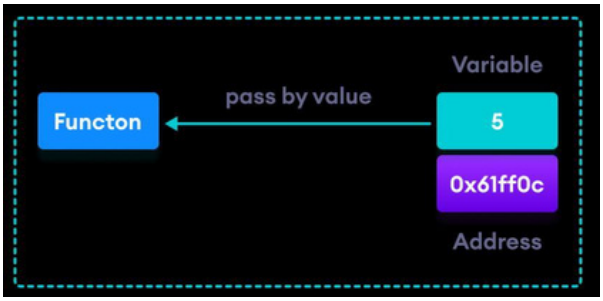
2. The arguments or parameter which are used in Called function is called formal argument



Call by value and Call by reference

There are two methods to pass the data into the function in C language, i.e., call by value and call by reference.

1. CallbyValue: In this method, a copy of the actual argument's **value** is passed to the function.
   This means any modifications made to the formal argument (parameter) within the functiondo not affect the original value of the actual argument outside the function.



Example:

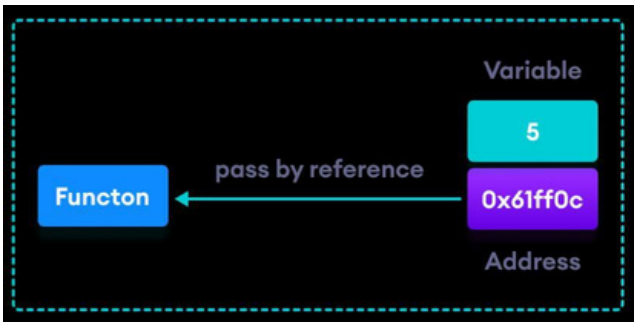
```
#include <stdio.h>
void modifyValue(int x) {
   x = x + 5;
}
int main() {
   int num = 10;
```

```
 printf("Before function call: num = %d\n", num);
   modifyValue(num);
   printf("After function call: num = %d\n", num);
   return0;
}
```

Thevalueof num in the main function remains unchanged before and after calling the function because“call by value” passes a copy of the variable, and modifications inside the function don’t affecttheoriginal variable.

2. CallbyReference: In this method, a copy of the actual argument’s **address** is passed to the function.

   This means if there is any change in the values inside the function, it reflects that changeinthe actual values.



Example:

```
#include <stdio.h>
void modifyValue(int *x) {
   *x = *x + 5; // Modifying the original value of x using a pointer
}
int main() {
   int num = 10;

   printf("Before function call: num = %d\n", num);

   modifyValue(&num); // Calling the function with a reference to num

   printf("After function call: num = %d\n", num);

   return 0;

}
```

In this example, the modifyValue function takes an argument x by reference (using a pointer). It directly modifies the original value of num by adding 5 to it.

Classification of Functions

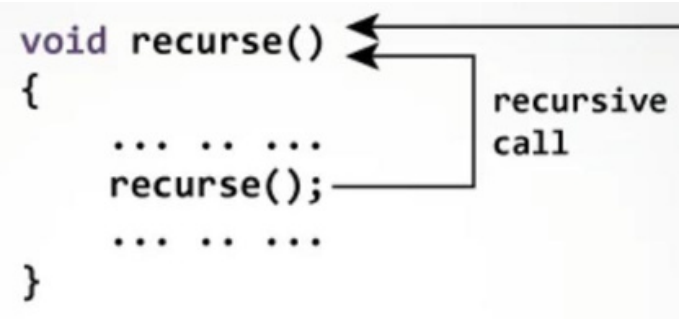
Based on theparametersandreturnvalues,functionscanbecategorized into four types. They are:

1. Function without arguments and without return value: A function that performs a task but doesn't take any inputs and doesn't return a value.
2. Function without arguments and with return value: A function that doesn't require inputs but provides a result as its output.
3. Function with arguments and with return value: A function that takes inputs, processes them, and returns a value as a result.
4. Function with arguments and without return value: A function that takes inputs, performs a task, but doesn't provide a return value.

Recursion

A functionis said tobe recursive, if a function calls itself within the function's definition.

Using recursion we can solve a complex problem in a small piece of code by breaking down the problem.



In this example we can see a function called recurse is calling itself in its definition.

a) Discuss how the recursion use the memory stack:

Recursion uses the memory stack, also known as the call stack, to keep track of function calls. Each time a function is called, a new stack frame is created and pushed onto the call stack. This stack frame contains information such as local variables, the return address, and other necessary information for the function's execution.

b) Types of Recursion in C

There are two types of recursion in the C language

- Direct Recursion

Direct recursion in C occurs when a function calls itself directly from inside. Such functions are also called direct recursive functions.

Example:

```
function_01()
{
   //some code
   function_01();
   //some code}
```

- Indirect Recursion

Indirect recursion in C occurs when a function calls another function and if this function calls the first function again. Such functions are also called indirect recursive functions. Example

```
function_01(){
   //some code
   function_02();
}
function_02(){
    //some code
  function_01();
}
```

In this example the function_01() executes and calls function_02(). After calling now, function_02 executes where inside it there is a call for function_01, which is the first calling function.

3. Advantages:

• The code becomes shorter and reduces the unnecessary calling to functions. • Useful for solving formula-based problems and complex algorithms. • Useful in Graph and Tree traversal as they are inherently recursive. • Recursion helps to divide the problem into sub-problems and then solve them,

essentially divide and conquer.

4. Disadvantages:

• The code becomes hard to understand and analyze.

• A lot of memory is used to hold the copies of recursive functions in the memory.

• Time and Space complexity is increased.

• Recursion is generally slower than iteration.

Applications of Recursion:

• Mathematical functions:

Recursion can be used to compute many mathematical functions, such as the factorial function, the Fibonacci sequence, and the greatest common divisor.

• Searching Algorithms: Recursive techniques are applied in searching algorithms

like binary search, making it an efficient algorithm for large datasets.

• Sorting Algorithms: Some sorting algorithms, such as quicksort and mergesort, use recursion as a key component of their design. These recursive sorting algorithms are known for their efficiency

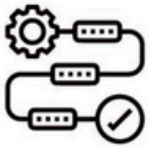
• Tree and graph traversal algorithms:

Recursion is often used to traverse trees and graphs, such as depth-first search and breadth-first search.

• Dynamic programming:

Many dynamic programming algorithms use recursion, such as the knapsack problem and the longest common subsequence problem.

Difference between recursion and iteration:

RECURSION	VS	ITERATION
		
In recursion, function calls itself directly or indirectly, thus achieving repetition		Iteration specifically uses a repetition structure with conditional loop
Recursion stops when the base case is reached		Iteration function gets terminated when the loop condition fails
Recursive function stores all the steps in a memory stack		Iteration doesn't use stack memory
Iteration will continue to modify the counter until the loop continuation condition fails		Recursion continues to generate simplified versions of the original problem until the base case is reached
Iterative code tends to be bigger in size		Recursive code is comparatively smaller
Stack overflow error occurs which may crash the system, if the base case is not defined		There will be an infinite loop if the control variable doesn't reach the termination value
Recursion is slower in execution		Iteration is comparatively faster
Best to use if problems can be divided into smaller subproblems that are similar to the original problem		Best to use if problems can be divided into smaller, repeating steps

Important Question

1. Explain the concept of recursive function with an example?
2. Differentiate between call by value and call by reference.
3. Write a program in C to add two numbers using pointers and function.
4. Differentiate between recursion and iteration.
5. What Is parameters in function.

The programming in C notes ends here.

Here are some tips to score well on your exam:

-

Data Structures

APC Ray Polytechnic // CST 3rd Sem. // 1st Internal Assessment // OCT - 2023

Full Time: 45 mins. ***Full Marks: 20***

Answer Q. 1 and any three from the rest

1. Define *linear* and *non-linear* data structure with example. "Algorithm has a close connection with data structure" – Explain. **3 + 2**
2. "The *stack* may be viewed as an *ADT*." – Justify. Explain the operations of *push* and *pop* on a contiguous stack with suitable examples. **2 + 3**
3. Write the algorithms for *insertion* and *deletion* operations on a contiguous linear queue. **2 x 2.5**
4. What is *associativity* of an operator? Convert the following *infix* expression into its equivalent *postfix* form making use of a stack: **1 + 4**

   A + (B * C – (D / E – F) * G) / H
5. Write short notes on any *two* of the following: **2 x 2.5**

   i) *ADT*

   ii) *Circular queue*

   iii) *Polish notation*

APC//1st Internal//OCT 2023//DCST & DEIE//Programming in C// F.M – 20//Time : 45 MIN

Answer any Four **5X4=20**

1. Discuss different operators in C with example.
2. "goto is generally avoided" – Discuss. Differentiate between break and continue statement.
3. Differentiate between the following-
   a) while and do-while loop
   b) while loop and for loop
4. Discuss switch statement with an example.
5. What do you mean by array? Write a program to add two one dimensional arrays.
6. Write a program to check whether a year is leap year or not

---

A.P.C Ray Polytechnic

Subject: SL(Python) 1st Internal F.M:20 Time:45 minutes

Answer any four Questions 4*5=20

1. Define Python. Explain the different feature of Python.
2. Differentiate between List and Dictionary in Python with example.
3. Explain break, continue and pass statement with example.
4. Define range functionwith example.
5. Write down the program using python

   *

   * *

   * * *

   * * * *
6. Explain while statement with example

1st Internal Assessment

CST, 3rd Semester

Computer system organization

Full Time: 45 mins. ***Full Marks: 20***

Answer any four:

1. Explain different types of instruction. 5
2. Explain zero & one addressing instruction 2.5+2.5=5
3. How floating point represented explain with an example. 5
4. What is micro operations? Explain different types of micro operations. 2+3=5
6. What us burst error? Explain hamming code with a suitable example. 1+4=5
7. explain registertransfer operations 5

Siliguri Government Polytechnic

Computer Science and Technology, 3rd Semester

1st Internal assessment, Computer System Organization, F.M.-20, Time- 45min

1. Explain Booth's algorithm with flowchart 5
2. Perform division of the following numbers using non-restoring division 1100 ÷ 11 5
3. Draw and explain 4-segment pipeline with space-time diagram. 5
4. State the differences between hardwired control and microprogrammed control. 5

Data Structures 1st Internal Assessment October 2023

F.M. 20 Time: 1 hour

Answer the following questions:

1 [CO1].	(a)	Define Array?	1
	(b)	In an integer Array, let the address of the 1st number is 1002. Find the address of the 3rd number.	2
	(c)	How will you formulate the address of the nth term in a character array where the lower bound index is 1.	2
2 [CO2].	(a)	Write down the basic structure of a singly linked list.	2
	(b)	Write a program to create a linked list and insert a new number at the beginning of the list.	3
3 [CO3].	(a)	Define Stack.	1
	(b)	Write any four applications of Stack.	1
	(c)	How insertion and deletion takes place in Stacks.	2
	(d)	How will you justify that a Stack is in overflow condition.	1
4 [CO4]	(a)	Write down the classification of Data Structure with example.	3
	(b)	'Circular linked list is a linear data structure'- Justify your answer.	2

SILIGURI GOVERNMENT POLYTECHNIC
INTERNAL EXAMINATION, 3RD SEMESTER
Algorithms

Time Allowed: 45 mins *Full Marks: 20*

1. a) Write down the algorithm of Insertion Sort. 5
   b) Compute the best-case and worst-case time complexities of Insertion Sort. 6

   or

   a) Write down the algorithm of Merge Sort. 5
   b) Compute the best-case and worst-case time complexities of Insertion Sort. 6

2. a) Define an algorithm. 2
   b) Write down short notes on the asymptotic notations. 7

Siliguri Govt. Polytechnic

Dept- CST, Sem- 3rd, Scripting Languages (Python), Internal Assessment – I, 2023-24 , FM-20, Time-45Min

Answer All Questions

1. What is membership and identity operator in python? What is tuple? What is dictionary? 3+2+2
2. Design the switch-case functionality in python. State ways to modify loops with code example 3+3
3. State difference between module and package. How many ways a module can be imported? State the rules of the order of arguments in function call 2+2+3

Siliguri Government Polytechnic

Computer Science and Technology, 3rd Semester, 1st Internal assessment, F.M.-20

Computer Programming In C

1. C is a structured and procedural programming language, explain. Explain ternary operator with an example 3+3
2. What are the different branching statements available in C, explain briefly. What are the different looping statements available in C, explain briefly. 3+3
3. What is an Array? How to declare and initialize a two-dimensional array. Write a program to insert 20 integers in an array and calculate sum and average of those numbers N natural numbers. 1+2+5

Babasaheb Ambedkar Govt. Polytechnic

FM 20 Programming Language C AY -2023-24 Date: 10.10.2023

Q.1 Answer any two from the followings. **5*2=10**

A. Explain the steps involved during compilation of a C program through a proper flowchart

B. What are the different data types used in C ? Mention their sizes according to system specification.

C. What do you mean by implicit and explicit type-casting ? Discuss with proper examples.

D. Give functional examples of:

1. Formatted input 2. Unformatted input 3.Formatted output 4.Unformatted output

Q.2 Answer any two from the followings. **5*2=10**

A. What do you mean by entry controlled and exit controlled loop structure ? Discuss with proper examples.

B. Write a C program to show obtained grades of given students marks

Marks 81-100. Grade A
Marks 61-80. Grade B
Marks 41-60. Grade C
Marks below 40. Grade F

C. Write a C program to compute factorial of a given number

D. Construct a calculator of basic 4 arithmetic operations between two numbers (addition, subtractor, multiplier, divisor) by a C program using switch case.

Kalna Polytechnic//1st Internal Test 2023//Computer Programming in C

F.M-20// Time:45 Minutes

Answer's Question no. 1 and any two from the remaining.

1. **Answers in one or two sentences only** **1X4=4**

a)Why we avoid to use goto statement in our program?

b) Explain the purpose of scanf() and printf() function.

c) Write the purpose of the header file math.h in c-program.

d) In what cases we use %d,%f and %c in our program.

2. Let us consider a program segment –

int i=25, j=35; 2X4=8

find the output for the following-

a) printf("%d%d", ++i,j++);

b) printf("%d",(i>j)?i:j));

c) i+ = ++I + j++;
printf("%d",i);

d) printf("%d%d" ,i%j , i/j);

3. a) Write the general form of while() and do-while() loop.

b) Differenciate them with suitable example.

c) Write a program to find out GCD&LCM of two given +ve integer no. 2+2+4=8

4. a) How you declare one dimension and two dimension array in a program.

b) Write a program to count the no. of element in a single dimension array

c) Write a program to create a 3X3 matrix. 2+3+3=8

2023/10/13

Thank you