**UNIT I BASICS OF C PROGRAMMING**

**PART-A**

1. **Write about features and applications of C?**

The features are

* 1. Structure oriented and procedural Language
  2. Fast and Efficient
  3. General-Purpose Language
  4. Rich set of built-in Operators
  5. Middle-Level Language
  6. Portable and Easy to Extend

The applications are

1. Operating Systems
2. GUI (Graphical User Interface)
3. Embedded Systems
4. Design of a Compiler
5. Gaming and animation
6. Platforms for new programming languages
7. **Define the rules for writing identifier names in c?**
   1. Identifier names must be unique.
   2. Keywords cannot be used as identifiers.
   3. Identifier has to begin with a letter or underscore \_
   4. It should not contain white space.
   5. Special characters are not allowed.
   6. Identifiers can consist of only letters, digits, or underscore.
   7. Identifiers are case sensitive.
8. **List down the formatted and unformatted I/O functions available?**

The formatted I/O functions are

1. printf()
2. scanf()
3. sprintf()
4. sscanf()

The unformatted I/O functions are

1. getch()
2. getche()
3. getchar()
4. putchar()
5. gets()
6. puts()
7. putch()
8. **Define variable?**

A variable in C refers to the name of a memory location that helps to store data and retrieves it when required. A variable can be used to store any type of data.

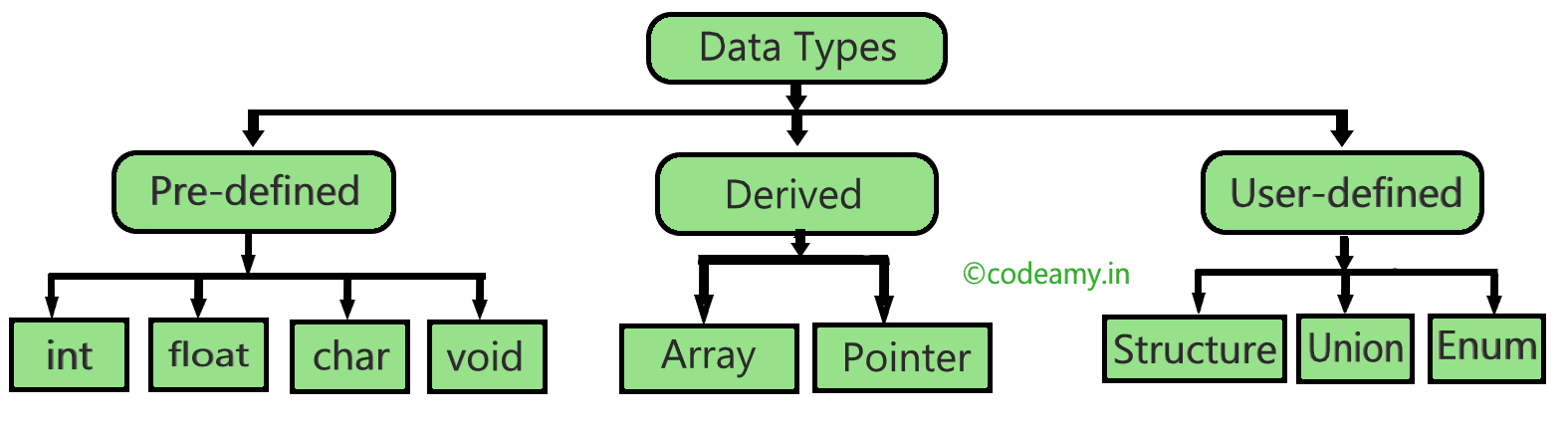
**Syntax: data\_type variable\_name = value;**

Ex: int var=10; // integer variable

char a; // character variable

1. Define Data type and its types?

A data type specifies the type of data that the variable can store like integer, character, floating, double, etc.



1. **Define Operator and Operand? Give example?**

Operators are symbols or keywords that perform specific operations on operands to produce a result. An operand is a term used to describe data values or variables that are operated on by an operator in an expression. Operands can be of various types, including variables, constants, and expressions.

**EX: int c = a + b;**

Here, ‘+’ is the addition operator. ‘a’ and ‘b’ are the operands

1. **How many bytes are occupied by the int, char, float, long int & double?**

int: 4 bytes (32 bits).

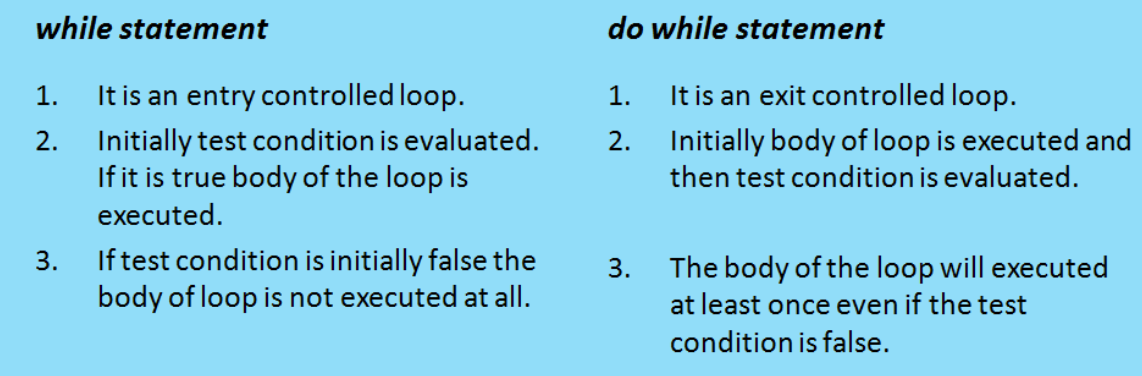
char: 1 byte.

float: 4 bytes (32 bits).

long int: 4 bytes

double: 8 bytes (64 bits).

1. What is the difference between while and do while statement?

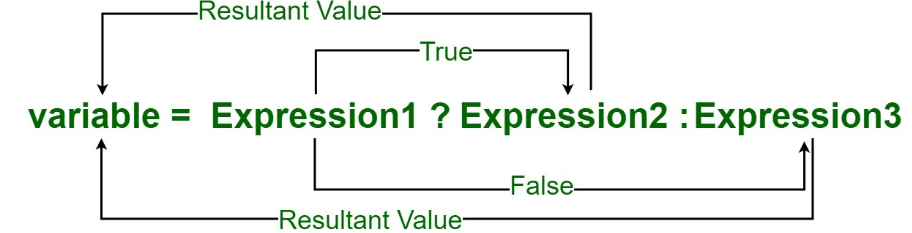


1. What is ternary Operator?

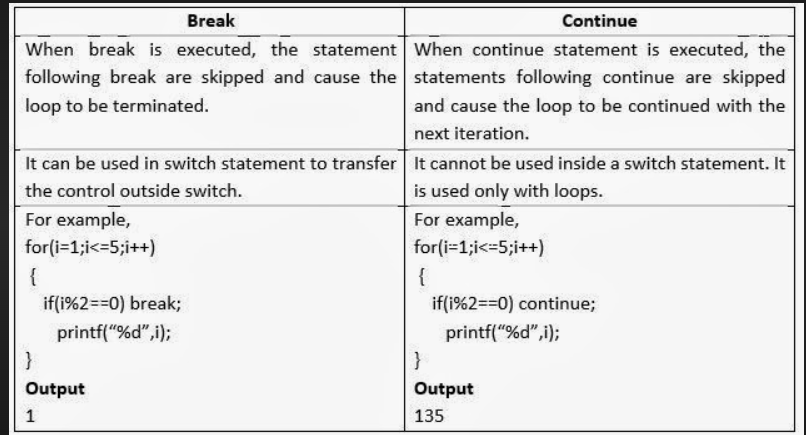
The ternary operator in C is a conditional operator that allows us to perform a simple conditional test and return a value based on the result of that test.

It's often referred to as the "conditional operator" because it provides a simple way to express conditional statements.

**Syntax:**



1. What is the difference between continue and break in C?



**PART-B**

1. Discuss in detail about various operators used in C with an example**.(Refer Notes)**
2. Explain briefly the formatted and unformatted I/O functions.

**Formatted I/O Functions:**

* Formatted I/O functions are used to take various inputs from the user and display multiple outputs to the user. These types of I/O functions can help to display the output to the user in different formats using the format specifiers.
* These I/O supports all data types like int, float, char, and many more.
* The following are the formatted I/O functions
  + printf()
  + scanf()

1. **printf ():** printf () function is used in a C program to display any value like float, integer, character, string, etc on the console screen.

**Syntax 1: printf (“Format Specifier”, var1, var2, …., varn);**

**Syntax 2: printf (“Enter the text which you want to display”);**

**Example: #include <stdio.h>**

**int main()**

**{**

**printf("This is a string");**

**return 0;**

**}**

**Output: This is a string**

1. **scanf ():** scanf () function is used in the C program for reading any value from the keyboard by the user, which can be of any data type like integer, float, character, string, and many more.

In scanf () function we use & (address-of operator) which is used to store the variable value on the memory location of that variable.

**Syntax: scanf(“Format Specifier”, &var1, &var2, …., &varn);**

**Example: int main()**

**{**

**int num1;**

**printf("Enter a integer number: ");**

**scanf("%d", &num1);**

**printf("You have entered %d", num1);**

**return 0;**

**}**

**Output**

**Enter an integer number: You have entered 10**

**Unformatted Input/Output functions**

* Unformatted I/O functions are used only for character data type or character array/string and cannot be used for any other datatype.
* These functions are used to read single input from the user at the console and it allows to display the value at the console.
* These functions are called unformatted I/O functions because we cannot use format specifiers in these functions and hence, cannot format these functions according to our needs.

The following are the unformatted I/O functions.

* getch()
* getche()
* getchar()
* putchar()
* gets()
* puts()
* putch()

1. **getch()**: getch() function reads a single character from the keyboard by the user but doesn’t display that character on the console screen and immediately returned without pressing enter key. getch() is also used for hold the screen.

**Syntax: getch(); or variable-name = getch();**

**Example: #include <conio.h>   
#include <stdio.h>   
int main()   
{   
 printf("Enter any character: ");   
 getch();   
 return 0;   
}**

**Output: Enter any character:**

1. **getche():** getche() function reads a single character from the keyboard by the user and displays it on the console screen and immediately returns without pressing the enter key.

**Syntax: getche(); or variable\_name = getche();**

**Example:**

**#include <conio.h>**

**#include <stdio.h>**

**int main()**

**{**

**printf("Enter any character: ");**

**getche();**

**return 0;**

**}**

**Output:Enter any character: g**

1. **getchar():** The getchar() function is used to read only a first single character from the keyboard whether multiple characters is typed by the user and this function reads one character at one time until and unless the enter key is pressed.

**Syntax: Variable-name = getchar();**

**Example:**

**#include <conio.h>**

**#include <stdio.h>**

**int main()**

**{**

**char ch;**

**printf("Enter the character: ");**

**ch = getchar();**

**printf("%c", ch);**

**return 0;**

**}**

**Output: Enter the character: a**

**a**

1. **putchar():**The putchar() function is used to display a single character at a time by passing that character directly to it or by passing a variable that has already stored a character.

**Syntax: putchar(variable\_name);**

**Example:**

**#include <conio.h>**

**#include <stdio.h>**

**int main()**

**{**

**char ch;**

**printf("Enter any character: ");**

**ch = getchar();**

**putchar(ch);**

**return 0;**

**}**

**Output: Enter any character: Z**

**Z**

1. **gets():** gets() function reads a group of characters or strings from the keyboard by the user and these characters get stored in a character array.

**Syntax: char str[length of string in number];   
gets(str);**

**Example:**

**#include <conio.h>**

**#include <stdio.h>**

**int main()**

**{**

**char name[50];**

**printf("Please enter some texts: ");**

**gets(name);**

**printf("You have entered: %s", name);**

**return 0;**

**}**

**Output: Please enter some texts: geeks for geeks**

**You have entered: geeks for geeks**

1. **puts():**In C programming puts() function is used to display a group of characters or strings which is already stored in a character array.

**Syntax: puts(identifier\_name );**

**Example:**

**#include <stdio.h>**

**int main()**

**{**

**char name[50];**

**printf("Enter your text: ");**

**gets(name);**

**printf("Your text is: ");**

**puts(name);**

**return 0;**

**}**

**Output: Enter your text: CyberSecurity**

**Your text is: CyberSecurity**

1. **putch():**putch() function is used to display a single character which is given by the user and that character prints at the current cursor location.

**Syntax: putch(variable\_name);**

**Example:**

**#include <conio.h>**

**#include <stdio.h>**

**int main()**

**{**

**char ch;**

**printf("Enter any character:\n ");**

**ch = getch();**

**printf("\nEntered character is: ");**

**putch(ch);**

**return 0;**

**}**

**Output: Enter any character:**

**Entered character is: d**

1. Write about types of Decision-making statements and discuss with example. **.(Refer Notes)**
2. Explain a structure of a C program. What are the advantages and applications of C language? **.(Refer Notes)**
3. Write a program for the following: **(Refer Notes)**

a) To check whether a number is prime or not.

b) To find the digits of a number.

c) To find the sum of first n integers.

d) To check whether a given number is a Armstrong number or not.

**UNIT II ARRAYS AND STRINGS**

**PART-A**

1. **Define an Array and its Types with an Example.**

An array is a collection of similar data elements stored at contiguous memory locations. It can be used to store the collection of primitive data types such as int, char, float, etc., and also derived and user-defined data types such as pointers, structures, etc.

**Syntax of Array Declaration:** data\_type array\_name [size];

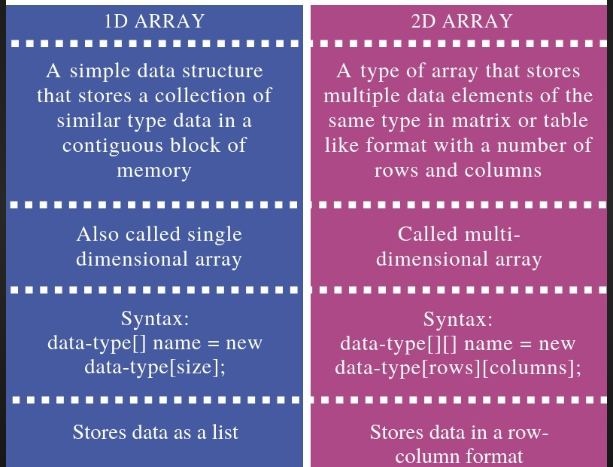
**Types of Arrays in C:**

There are two types of arrays based on the number of dimensions it has.

One Dimensional Arrays (1D Array)

Multidimensional Arrays (2D Array)

1. **Write the Difference between one dimensional and two-dimensional array.**

****

1. **Write the Syntax for declaring two dimensional arrays**.

Syntax: data\_type array\_name[row\_size][column\_size];

**data\_type:** This represents the data type of the elements that will be stored in the array, such as int, float, or any other valid C data type.

**array\_name:** This is the name of the array.

**Example: int matrix[3][4];** // Declares a 3x4 integer array named "matrix"

1. **Why don’t we use the & symbol while reading string through scanf?**

Strings in C are represented as arrays of characters. The name of the string itself points to the base address of the character array. So there is no need to use & in scanf function.

**Example:**

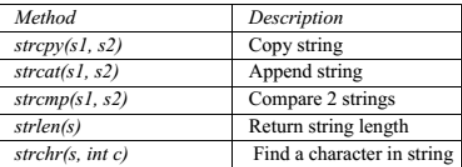
char name[50];

printf("Enter your name: ");

scanf("%s", name); // No & symbol before 'name'

In this code, the name array is already a pointer to the first character of the string, so the function knows that it's dealing with a string and reads characters into the array accordingly.

1. List out four predefined functions for string manipulation.



1. Write syntax for string declaration with an example

A String in C programming is a sequence of characters terminated with a null character ‘\0’. The C String is stored as an array of characters.

**Syntax: char string\_name[size]; Or**

**char string\_name[size] = "string";**

char c[] = "abcd";

char c[50] = "abcd";

char c[] = {'a', 'b', 'c', 'd', '\0'};

char c[5] = {'a', 'b', 'c', 'd', '\0'};

1. **What is sorting? and why it is necessary?**

Sorting in c is the processing of arranging the data in ascending and descending order. There are several types of sorting in data structures, namely – bubble sort, insertion sort, selection sort, bucket sort, heap sort, quick sort, radix sort, etc.

By sorting data, it is easier to search through it quickly and easily. The simplest example of sorting is a dictionary.

1. **What is linear search? Write its advantages and disadvantages**

Linear search, also known as a sequential search, is a straightforward searching algorithm that is used to find a specific element within a list or array.

It works by examining each element in the list one by one until a match is found.

**Algorithm:**

Start from the beginning of the list.

Compare the target element with each element in the list, in order.

If a match is found, return the index (position) of the element. If the entire list is searched and no match is found, return a sentinel value (e.g., -1) to indicate that the element is not in the list.

**Advantages of Linear Search:**

* Simplicity
* Applicability
* No pre-processing

1. **What is binary search? Write its advantages and disadvantages**

Binary search is a search algorithm that is based on the principle of divide-and-conquer strategy and is used to efficiently locate a target element by repeatedly dividing the search space in half until the desired element is found.

**Algorithm:**

Start with the entire sorted array.

Compare the target element with the middle element of the array.

If the target element matches the middle element, the search is successful.

If the target element is less than the middle element, continue the search in the left half of the array.

If the target element is greater than the middle element, continue the search in the right half of the array.

**Advantages:**

* Efficiency
* Applicability
* Optimization:

1. **How can you assign one array to another array?**

**#include <stdio.h>**

**int main()**

**{**

**int a[5] = { 3, 6, 9, 2, 5 }, n = 5;**

**int b[n], i;**

**// copying elements from one array to another**

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

**{**

**b[i] = a[i];**

**}**

**printf("\nThe copied array is :");**

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

**{**

**printf("%d ", b[i]);**

**}**

**return 0;**

**}**

**Output**

**The copied array is :3 6 9 2 5**

PART-B

1. **Explain string handling functions strcat(), strcpy() & strcmp()with syntax and** example.

**1. strcpy() function in C Language**

strcpy() is used store a copy one string to another string variable

Syntax : strcpy(Str2,Str1);

Str2 is the string variable to store the value.

Str1 is the string value to be stored in string variable Str2.

**Program to demonstrate the use of strcpy() function in C.**

#include<stdio.h>

#include<string.h>

int main()

{

char name[20];

strcpy(name,”Rahul”);

printf("name=%s",name);

return(0);

}

Output

name=Rahul

**2. strcat() function in C Language**

strcat() function is used to combine values of two string variables .

Syntax : strcat(Str2,Str1);

Str2 is the string variable whose value will be combined with another string value.

Str1 is the string value to be combined with string variable Str2.

**Program to demonstrate the use of strcat() function in C.**

#include<stdio.h>

#include<string.h>

int main()

{

char name[30]=”Rahul”;

strcat(name,” Singh”);

printf("name=%s",name);

return(0);

}

Output

name=RahulSingh

**3. strlen() function in C Language**

strlen() function is used to count and return number of characters in a string value.

Syntax : strlen(Str);

Str is the string value whose length we want to find.

**Program to demonstrate the use of strlen() function in C.**

#include<stdio.h>

#include<string.h>

int main()

{

char name[30]="Rahul";

int num;

num=strlen(name);

printf("\nNumber of characters=%d",num);

return(0);

}

Output

Number of characters=5

**4. strrev() function in C Language**

strrev() function is used to reverse the characters stored in a string value.

Syntax: strrev(Str);

Str is the string variable whose value we want to reverse.

**Program to demonstrate the use of strrev() function in C.**

#include<stdio.h>

#include<string.h>

int main()

{

char name[20]=”Rahul”;

strrev(name);

printf("name=%s",name);

return(0);

}

Output

name=luhaR

**5. strcmp() function in C Language**

strcmp() function is used to compare two string values. 0 is returned if the values of strings being compared are same.

Syntax: strcmp(Str1,Str2);

Here, Str1 and Str2 are string variables or constants which we want to compare.

**Program to demonstrate the use of strcmp() function in C.**

#include<stdio.h>

#include<string.h>

int main()

{

char name[35];

strcpy(name,”Sumit”);

if(strcmp(studentname,”Amit”)==0)

cout<<”\nWelcome Amit”;

else

cout<<”Wrong User”;

return(0);

}

**Output**

**Wrong User**

1. Writes notes on arrays and its types with example**.(Refer Notes)**
2. Explain about linear search and binary search in detail. .(**Refer Notes)**
3. What is meant by sorting. Write a C program to implement selection sort. **.(Refer Notes)**
4. Write a c program **.(Refer Notes)**
5. To print an array in a reverse order.
6. To check the given string is palindrome or not.
7. To perform matrix addition
8. To count no of occurrences of a particular character in a string.