EXPERIMENT NO. 1

Title: Write C Program to understand input and output function

a. Find the circumference of circle
b. Find the sum and average of 5 marks of subjects.

Objectives:

1. To understand input and output statements in c.

Theory:

printf( ) and its Purpose-

- C does not contain any instruction to display output on the screen.
- All output to screen is achieved using readymade library functions. One such function is printf( ).
- Once the computatation is done its result needs to be displayed on the screen. We have used printf( ) to do so.
- For us to be able to use the printf( ) function, it is necessary to use #include <stdio.h> at the beginning of the program. #include is a preprocessor directive

The general form of printf( ) function is,

printf ("<format string>", <list of variables>) ;

<format string> can contain,

%f for printing real values
%d for printing integer values
%c for printing character values

In addition to format specifiers like %f, %d and %c, the format string may also contain any other characters. These characters are printed as they are when printf( ) is executed.

Scanf()

To make the program general, the program itself should ask the user to enter the values for computation through the keyboard during execution. This can be achieved using a function called scanf( ). This function is a counter-part of the printf( ) function. printf( ) outputs the values to the screen whereas scanf() receives them from the keyboard.

```
scanf("%d", &a);
```

Here, use of ampersand (&) before the variables in the scanf( ) function is a must. & is an 'Address of' operator. It gives the location number (address) used by the variable in memory. When we say &a, we are telling scanf( ) at which memory location should it store the value supplied by the user from the keyboard.

Program:

Output:

Conclusion:

EXPERIMENT NO.2

Title: Write C Program to understand Operators

a. Display the output by applying arithmetic and logical operators on two numbers.
b. Find the sum of 3 digits number using arithmetic operators.

Objectives:

2. To understand and apply operators in c program.

Theory:

Arithmetic operators:
Arithmetic Operators are the type of operators in C that are used to perform mathematical operations in a C program. They can be used in programs to define expressions and mathematical formulas.
Arithmetic Operators in C:
The C arithmetic operators are the symbols that are used to perform mathematical operations on operands. There are a total of 9 arithmetic operators in C to provide the basic arithmetic operations such as addition, subtraction, multiplication, etc.

Types of Arithmetic Operators in C:
The C Arithmetic Operators are of two types based on the number of operands they work. These are as follows:

1. Binary Arithmetic Operators
2. Unary Arithmetic Operators

1. Binary Arithmetic Operators in C
The C binary arithmetic operators operate or work on two operands. C provides **5** Binary Arithmetic Operators for performing arithmetic functions which are as follows:

Operator	Name of the Operator	Arithmetic Operation	Syntax
+	Addition	Add two operands.	x + y
–	Subtraction	Subtract the second operand from the first operand.	x – y
*	Multiplication	Multiply two operands.	x * y
/	Division	Divide the first operand by the second operand.	x / y
%	Modulus	Calculate the remainder when the first operand is divided by the second operand.	x % y

2. Unary Arithmetic Operators in C

The unary arithmetic operators operate or work with a single operand. In C, we have two unary arithmetic operators which are as follows:

Operator	Symbol	Operation	Implementation
Decrement Operator	--	Decreases the integer value of the variable by one.	–h or h–
Increment Operator	++	Increases the integer value of the variable by one.	++h or h++

Logical operators:

Logical operators in C are used to combine multiple conditions/constraints. Logical Operators returns either 0 or 1, it depends on whether the expression result is true or false. In C programming for decision-making, we use logical operators.

We have 3 logical operators in the C language:

- Logical AND (&&)
- Logical OR (||)
- Logical NOT (!)

Program:

Output:

Conclusion:

EXPERIMENT NO.3

Title: Use Control statements to write C program
a. Check Whether a Number is Even or Odd or positive or negative.
b. Check whether a character is vowel or consonant using switch case.

Objectives:

1.To understand and apply control statements in c program.

Theory:

1. The if-else

The if-else statement in C is used to perform the operations based on some specific condition. The operations specified in if block are executed if and only if the given condition is true.

```
if(expression)
{
//code to be executed
}
```

There are the following variants of if statement in C language.

- If statement
- If-else statement
- If else-if ladder
- Nested if

2. Switch statement

Switch statement in C is an alternate to if-else-if ladder statement which allows us to execute multiple operations for the different possible values of a single variable called switch variable. Here, We can define various statements in the multiple cases for the different values of a single variable.

```
switch(expression){
case value1:
 //code to be executed;
 break;  //optional
case value2:
 //code to be executed;
 break;  //optional
.....

default:
 code to be executed if all cases are not matched;
}
```

Program:

Output:

Conclusion:

EXPERIMENT NO.4

Title: Apply loop to write C program
a.Display the Armstrong numbers between 1 to 500
b.Find the factorial of any number entered through keyboard.

Objectives:

1. To apply loop in c program.

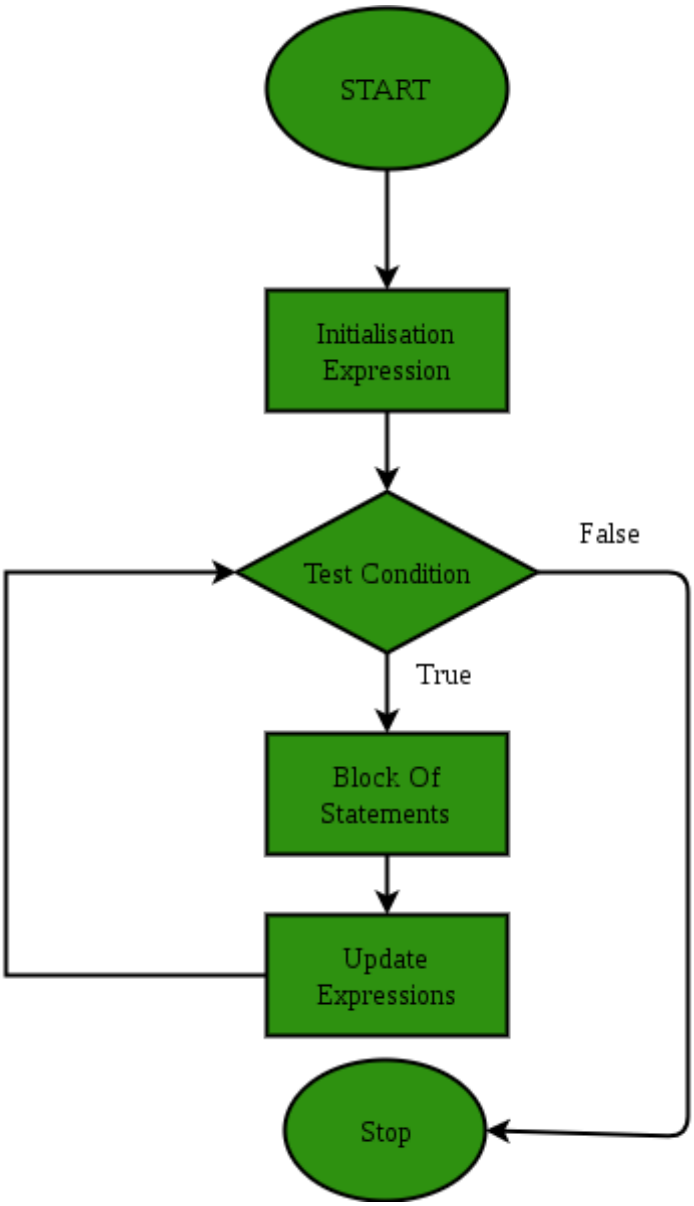
Theory:
Loops in programming are used to repeat a block of code until the specified condition is met. A loop statement allows programmers to execute a statement or group of statements multiple times without repetition of code.

Loop Type	Description
for loop	first Initializes, then condition check, then executes the body and at last, the update is done.
while loop	first Initializes, then condition checks, and then executes the body, and updating can be inside the body.
do-while loop	do-while first executes the body and then the condition check is done.

Syntax:

```
for (initialize expression; test expression; update expression)
{
   //
   // body of for loop
   //
}
```

for loop Equivalent Flow Diagram



Program:

Output:

Conclusion:

EXPERIMENT NO.5

Title: Write C Program using 1-D Array

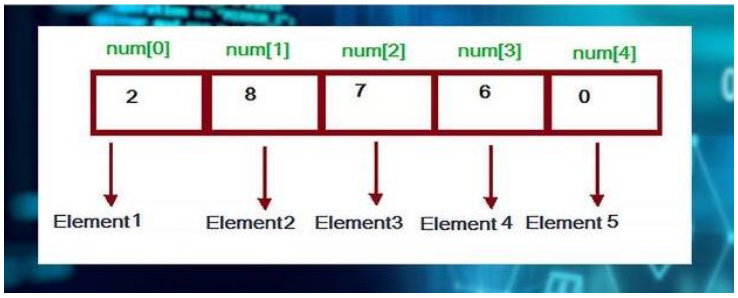
a. Find the largest and smallest number of an array.
b. Arrange an array in descending order

Objectives:

1.To understand and apply 1-D in c program.

Theory:

Arrays are a fundamental concept in programming, and they come in different dimensions. A one dimensional array in C contains a series of elements of the same datatype. The elements are zero-indexed, meaning the first element is in position or index 0, and the last element is in position or index (array size) - 1.



Syntax :

dataType arrayName[array Size];

- data Type specifies the data type of the array. It can be any valid data type in C programming language, such as int, float, char, double, etc.
- array Name is the name of the array, which is used to refer to the array in the program.
- array Size specifies the number of elements in the array. It must be a positive integer value.

Program:

Output:

Conclusion:

EXPERIMENT NO.6

Title: Write C Program using 2-D Array
a. Find Transpose of a Matrix
b. Calculate Addition of All Elements in the matrix

Objectives:

1.To understand and apply 2-D in c program.

Theory:

a 2D array, is a collection of data elements arranged in a grid-like structure with rows and columns. Each element in the array is referred to as a cell and can be accessed by its row and column indices/indexes.The two-dimensional array can be defined as an array of arrays. The 2D array is organized as matrices which can be represented as the collection of rows and columns. However, 2D arrays are created to implement a relational database lookalike data structure. It provides ease of holding the bulk of data at once which can be passed to any number of functions wherever required.

Syntax:
data_type array_name[rows][columns];

Initialization of 2D Array in C:

In the 1D array, we don't need to specify the size of the array if the declaration and initialization are being done simultaneously. However, this will not work with 2D arrays. We will have to define at least the second dimension of the array. The two-dimensional array can be declared and defined in the following way.

```
int arr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};
```

Program:

Output:

Conclusion:

EXPERIMENT NO.7

Title: Write Function in C to

a. Find the Power of a Number.
b. Find the year is leap or not.

Objectives:

1.To understand function in c program.

Theory:

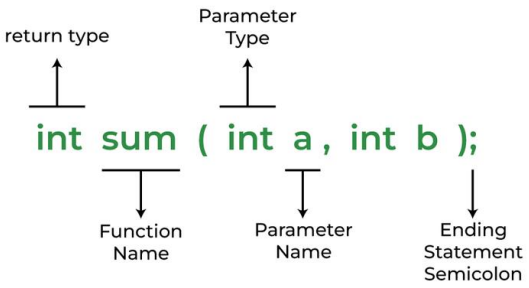
A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. Functions are used to perform certain actions, and they are important for reusing code: Define the code once, and use it many times.

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 that provides modularity and code reusability. The programming statements of a function are enclosed within **{ } braces**, having certain meanings and performing certain operations

Syntax of Functions in C
The syntax of function can be divided into 3 aspects:

1. Function Declaration
2. Function Definition
3. Function Calls

return_type **name_of_the_function** (*parameter_1*, *parameter_2*);



Program:

Output:

Conclusion:

EXPERIMENT NO.8

Title: Write C Program using Pointer
a. Access Elements of an Array Using Pointer
b. Swap Numbers in Cyclic Order Using Call by Reference.

Objectives:

1.To understand pointer in c program.

Theory: A pointer is defined as a derived data type that can store the address of other C variables or a memory location. We can access and manipulate the data stored in that memory location using pointers.

Syntax:

datatype * ptr;

where
- ptr is the name of the pointer.
- datatype is the type of data it is pointing to.

The use of pointers can be divided into three steps:
1. Pointer Declaration
2. Pointer Initialization
3. Dereferencing

1. Pointer Declaration
In pointer declaration, we only declare the pointer but do not initialize it. To declare a pointer, we use the (*) dereference operator before its name.
Example
int *ptr;

The pointer declared here will point to some random memory address as it is not initialized. Such pointers are called wild pointers.

2. Pointer Initialization
Pointer initialization is the process where we assign some initial value to the pointer variable. We generally use the (&) addressof operator to get the memory address of a variable and then store it in the pointer variable.
Example
int var = 10;
int * ptr;
ptr = &var;

3. Dereferencing
Dereferencing a pointer is the process of accessing the value stored in the memory address specified in the pointer. We use the same (*) dereferencing operator that we used in the pointer declaration.

Program:

Output:

Conclusion:

EXPERIMENT NO.9

Title: Write C Program on String

a. Find the Length of a String Using Library Function.

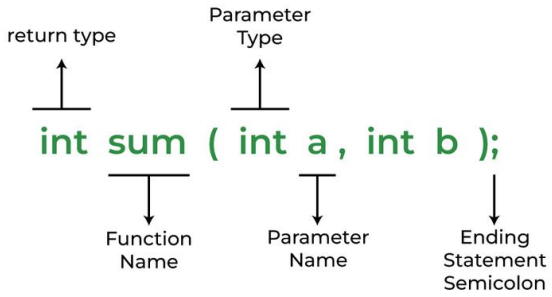
b. Copy String without using strcpy ().

Objectives:

1.To understand string in c program.

Theory:

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. The difference between a character array and a C string is the string is terminated with a unique character ‘\0’.



C String Declaration Syntax

char string_name[size];

In the above syntax str_name is any name given to the string variable and size is used to define the length of the string, i.e the number of characters strings will store.

There is an extra terminating character which is the Null character (‘\0’) used to indicate the termination of a string that differs strings from normal character arrays.

Program:

Output:

Conclusion:

EXPERIMENT NO.10

Title: Write C Program on Structure

a. Calculate Size of Structure using Size of Operator

b. Create a structure book, which consists book_name, author_name and price. Accept the information of books and display it.

Objectives:

1.To understand structure in c program.

Theory:

Structure in C is a user-defined data type. It is used to bind two or more similar or different data types or data structures together into a single type. The structure is created using the struct keyword, and a structure variable is created using the struct keyword and the structure tag name.

C Structure Declaration

We have to declare structure in C before using it in our program. In structure declaration, we specify its member variables along with their datatype. We can use the struct keyword to declare the structure in C using the following syntax:

Syntax:

```
struct structure_name {
   data_type member_name1;
   data_type member_name1;
   ....
   ....
};
```

The above syntax is also called a structure template or structure prototype and no memory is allocated to the structure in the declaration.

The C dot (.) operator is used for direct member selection via the name of variables of type struct and union. Also known as the direct member access operator, it is a binary operator that helps us to extract the value of members of the structures and unions.

Syntax of Dot Operator

variable_name.member;
variable_name: An instance of a structure or a union.
member: member associated with the created structure or union.

Program:

Output:

Conclusion: