2. Define the following terms as used in C programming

a) Compiler → a program that translates a programming language's source code into machine code, bytecode or another programming language

b) Source code → human readable text that a ~~compute~~ programmer writes to create a computer program.

c) Object code → binary representation of a program that is generated after the source code is compiled. It is the result of compiling C source code, and it's the machine-understandable representation of your program, ready for execution

d) Linkers → it is a program that combines multiple object code files and libraries into a single executable program. They play a significant role in transforming the source code into a runnable application.

4. Explain the differences between a compiler and interpretor. Give at least 6 comparisons.

	Compiler	Interpretor
a)	Translates the entire source code into machine code or an intermediate language all at once for execution	Processes the source code line by line during execution.
(b)	Produces a standalone executable file, which can be run without the need for the original source code	Does not produce a separate executable file as it directly executes the source code, and the original code must be present to run the program.
(c)	Produces faster-running code since it optimizes the entire program before execution.	Slower as it translates and executes code line by line, without the benefit of advanced optimizations.

4. May provide less detailed error messages since it checks the entire code at once as debugging can be more challenging.	Often provides more detailed error messages and allows easier debugging because it stops execution at the first encountered error.
5. Produces machine-specific code, so you may need to recompile the code for each target platform.	Tends to be more portable, as long as an interpreter is available for the target platform.
6. Examples are ; C, C++ and Java	Examples; Python, JavaScript and Ruby

5. List all main categories of operators available in C & specific operators in each category

a) Arithmetic Operators; addition (+), subtraction (-), multiplication (x) division (/), modulus (remainder) (%), increment (++), decrement (--)

b) Relational Operators; equal to (==), not equal to (!=), less than (<) greater than (>), less than or equal to (<=), greater than or equal to (>=)

c) Logical operators; Logical AND (&&), Logical OR (||), logical NOT (!)

d) Assignment Operators; Assignment: =, Addition assignment: += subtraction assignment: -=, Multiplication assignment *=, division assignment: /=, modulus assignment: %=

e) Bitwise Operators: Bitwise AND: &, Bitwise OR: | Bitwise XOR (Exclusive OR): ^, Bitwise NOT (complement): ~ Left shift: << Right shift: >>

f) Conditional (Ternary) operator; Conditional operator (Ternary): ?

g) Unary Operators: Unary Plus :(+), Unary minus (-), Increment (++), Decrement (--), Logical NOT (!), Bitwise NOT (!), Address-of : & Dereference(indirection): *

h) Comma Operator; ~~Comma o~~ (,)
i) Size of Operator ; sizeof

3. Using an example, i.e., a program to add 2 numbers, explain the compilation process of a C program.
Create the C program by writing the source code, which is the human-readable text, which in this case is the two numbers. Before compilation, the source code may go through a pre-processing stage. This stage instructs the preprocessor to include the contents of the standard I/O library in your program. The C compiler is used to compile the source code. It translates the C source code into objectcode. During compilation, the code is checked for syntax errors and type consistency. The compiler generates object code and it includes the instructions and data for your program but is not yet executable The linker combines the object code with the necessary librarycode to create an executableprogram ~~whi~~ and also ~~it~~ resolves references to external functions andensures everything fits together, and can run directly on the target platform.