2) Define:

a) Compiler : A compiler is a software tool that translates high-level programming languages (like C) into machine code or an intermediate code.

b) Source code : Refers to the human-readable code written by a programmer in a high-level programming language, such as C.A source code file typically have extensions like “.c” in C programing

C) Object code : It is a machine-readable binary code that is not directly human-readable but is executable by a computer's CPU.

d) Linkers : are tools that combine multiple object files generated by the compiler into a single executable program.

4) Differences between compiler and interpreter.

1. The compiler translates the entire source code into machine code before execution while an interpreter translates and execute the source code line by line.
2. Compiler produces a faster-running code because the entire code is translated before execution while the Interpreter is typically slower as it executes code line by line.
3. The complier produces a code that is self dependant /standalone executable files while Interpreter requires the interpreter to be present on the system where the code is executed.
4. Generally compiled programs consumes less memory during execution because the entire program is translated into machine code before running while the interpreted programs may consume more memory as it needs to interprete and execute code on fly.
5. Compiler often has a separate debugging phase because debugging is done on executable codes while in interpreter allows for easier debugging since errors can be identified and corrected during the execution of program.
6. Compiled programs are generally less portable while in interpreter source code is more portable since it can be executed on any system with appropriate interpreter

5) Main categories of operators available in C programming and specific operator

1. **Arithmetic Operators:**

Example : +, -, %, \*, /,

1. **Relational Operators:**

**Example : ==(equal to), !=(not equal to), <(less than), >(greater than), <=(less than or equal to, >=(greater than or equal to) .**

1. **Logical Operators:**

**Example:** && (Logical AND), || (Logical OR) and ! (Logical NOT).

1. **Assignment Operators:**

**Example**= (Assignment), += (Add and assign), -=, \*= (Multiply and assign), /=, and %=.

1. **Increment and Decrement Operators:**

**Example :** ++ (Increment) , -- (Decrement).

1. **Bitwise Operators:**

**Example:** & (Bitwise AND), | (Bitwise OR), ^ (Bitwise XOR), ~ (Bitwise NOT), << (Left shift), and >> (Right shift).

1. **Conditional (Ternary) Operator:**

Example : ?: (Conditional operator).It is often used as a shorthand for an if-else statement.

1. **Bitwise Shift Operators:**

Example: << (Left shift), >> (Right shift).

1. **Unary Operators:**

Example: - (Unary minus), + (Unary plus) , ! (Logical NOT), and ~ (Bitwise NOT).

3)

1. **Preprocessing:**

The preprocessor takes care of directives , including the contents of the standard stdio.h your code.The preprocessed code is generated and saved in an intermediate file.

1. **Compilation:**

The compiler translates the preprocessed code into assembly code .It checks for syntax errors, semantic errors, and generates object code.

1. **Assembly (optional):**

Some compilers may include an optional step to translate the generated assembly code into machine code.

1. **Linking:**

The linker combines the object code generated from your program with the object code of any library functions. It resolves references between different parts of your code and libraries, ensuring everything is connected.

1. **Executable Output:**

The final result is an executable file that can be run on your computer.This executable file contains machine code that the computer's CPU can understand.

Example of a program.

include <stdio.h>

int main() {

int num1, num2, sum;

printf("Enter first number: ");

scanf("%d", &num1);

printf("Enter second number: ");

scanf("%d", &num2);

sum = num1 + num2;

printf("Sum: %d\n", sum);

return 0;

}