**2)A compiler** is a special program that translates a programming language's source code into machine code, bytecode or another programming language

**Source code** is a group of instructions a programmer writes using computer programming languages

**Object code** generally refers to the output, a compiled file, which is produced when the Source Code is compiled with a C compiler

**linker** is a computer system program that takes one or more object file and combines them into a single executable file, library file, or another object file.

4)Compiler:Store machine language as

machine code on the disk

Interpreter:Not saving machine code at all.

Compiler; Compiled code run faster

Interpreter;Interpreted code run slower

Compiler ;Generates output program which can

be run independently from the original program.

Interpreter;Do not generate output program. So they evaluatethe source program at every time during execution.

Compiler; Takes an entire program

Interpreter;Takes a single line of code

Compiler;Generate intermediate machine code

Interpreter:Never generate any intermediate machine code

Compiler;Display all errors after, compilation,

all at the same time.

Interpreter;Displays all errors of each line one by one.

3)1. Preprocessing

A preprocessor is used in the compilation process's initial phase. The source code is processed by the preprocessor at this phase prior to compilation.Expansion of macros `#{ directives, such as {#include}, are managed in this phase. Comments are also removed.

2. compilation

The preprocessed code is compiled following preprocessing. The preprocessed code is converted into platform-specific assembly code by the C compiler during compilation.

3.assembling

In this step, syntax and semantic errors are examined. An object file is generated by the assembler from the assembly code it got from the compiler if there are no errors.

4.Linking

The object code generated in the assembly stage is made up of machine instructions that the processor understands, but some parts of the program are out of order or missing. To produce an executable program, the existing pieces must be rearranged and the missing pieces must be completed. This process is called bonding.

Long programs are broken into files and converted to object code separately and the linker will link them and make them ready for execution.This is separate compilation.

The executable file is loaded into memory by the operating system . When the programme runs, it asks the user for two integers, adds them, and shows the final result.

5)

1. Increment and decrement operators

These are (++) and (-) resp

1. Bitwise operators

Bitwise AND: converts the two operands into binary and performs conjunctive operation bit by bit.

Bitwise OR: converts the two operands into binary and performs disjunctive operation bit by bit.

Bitwise LEFT SHIFT:

Bitwise RIGHT SHIFT:

Bitwise XOR: converts both operands into binary and performs xor operation bit by bit

Bitwise ONE’S COMPLEMENT: returns the complementary form of the operand.

1. Assignment operators

=assignment

+=addition assignment

/=division assignment

\*=multiplication assignment

-=subtraction assignment

%=modulus assignment

1. Logical operators

Unary operators

Binary operators

1. Relational operators

<lesser than

>greater than

!=not equal to

==equals to

<=lesser than equal to

>=greater than equal to

1. Special operators
2. Conditional operators
3. Arithmetic Operators

Include:Addition,subtraction,division,multiplication,modulus