2. Define the following terms as used in c programming.

1. Compiler: a compiler is a software tool that is used to translate c code into machine Code or executable code that can be run on a computer.
2. Source code: this is a text based program written by a programmer in c programming language.It is also the human readable representation of a program's logic and functionality.
3. Object code: Refers to the immediate code generated by the compiler during compilation process.it is not the final executable code that can run directly on a computer, instead it's an essential step in process of creating an executable program.
4. Linkers:It is a software tool responsible for combining multiple object files and libraries into a single executable program.

3.using an example of a program to add two numbers explain the compilation process of a c program.

* Writing the source code; first write the c program code.use codeblocks to create a.c file. Then start writing the code.
* # include <studio.h>

Int main( ) {

Int num1,num2,sum;

Print f("enter two numbers:");

Scan f("%d%d",&num1 ,&num2);

Sum=num1+num2;

Printed("sum:% d/n", sum);

Return 0;

}

The program prompts the user to enter two numbers,adds them, and prints the sum.

* Compilation: once written we will have to compile it you can use a c compiler like gcc for c programming.open your terminal and run the following command to compile code: gcc-o-add numbers.c

Gcc compiler

-oadd- numbers: this is option specific output file name (the executable file).we should name it maths.

-oadd- numbers.c: the name of your c source code file.

* Execution:after compilation you will have an executable file named"add number" you can run the program from command by typing:

./add numbers

The program will execute prompt you enter two numbers then display the sum.

4. Differences between compiler and interpreter.

They both convert high level programming code into a lower level format that can be executed by a computer.

1. Translation process;

* Compiler; A compiler translates the entire source code of a program into machine Code of a program or an intermediate code all at once.It generates an executable file, and this file can be run independently of the source code.
* Interpreter; Translates the source code line by line or statement by statement while program is being executed.It doesn't produce a separate executable file;the source code is executed directly.

2. Execution.

* Compiler; After compilation the program runs independently of the source code. This means that the source code and the executable program are separate entities.
* Interpreter; The source code is required to be present during execution, and the program is executed directly from source code.

3. Error Handling.

* Compiler; Often catches many errors before the program is executed. This means you may encounter errors only after you have finished writing your entire program.
* Interpreter; Errors are typically detected and reported as they occur during execution.This makes it easier to detect errors and fix them.

4. Execution speed.

* Compiler; compiled programs run faster since entire code is translated into machine Code before execution.
* Interpreter; Interpreted programs are slower since they translate and execute code line by line.

5. Portability.

* Compiler; The same compiled executable program code can be run on a different computer without the need for source code or compiler, as long as the target computer has compatible architecture.
* Interpreter; Programs may not be as portable, as they rely on interpreter being available on target system to execute the source code.

6. Examples.

* Compiler; c,c++ and rust are languages being compiled.
* Interpreter; Python,ruby,and javascript are examples of languages being interpreted

5.list all categories of specific operators available in c programming.

* Arithmetic operators.- they are of two types;
  + Unary operators: operate and work with a single operand.they include. Increment(++) and decrement (-) operators.
  + Binary operators: operators that operate with two operands. They include;
    - +(Addition)
    - -(subtraction)
    - \*(Multiplication)
    - /(Division)
    - %modulus,gives the remainder of division.
* Relational operators:
  + ==(Equal to)
  + !=(not equal to)
  + <(Less than)
  + >(Greater than)
  + <=(Less than or equal to)
  + >=(Greater than or equal to)
* Assignment operators:
  + =(Assignment)
  + +=(Addition assignment)
  + -=(subtraction assignment)
* Logical operators
  + &&(Logical AND)
  + ||(Logical OR)
  + !(logical NOT)
* Assignment operators
  + \*=(Multiplication assignment)
  + /=(Division assignment)
  + %=(modulus assignment)
  + &=(Bitwise and assignment)
  + |=(Bitwise or assignment)
  + ^=(bitwise XOR assignment)
  + <<=(Left shift assignment)
  + >>=(Left shift assignment)
* Bitwise operators
  + &(Bitwise AND)
  + |(Bitwise OR)
  + ^(bitwise XOR)
  + ~(bitwise NOT)
  + <<(Left shift)
  + >>(Right shift)
* Conditional (ternary) operator
  + ?:(conditional operator,used for decision making.