1.create a c program to store and manage student details.

#include <stdio.h>

struct Student {

char regNumber[20];

char firstName[50];

char lastName[50];

float programming;

float networking;

float accounting;

float computation;

float operatingSystem;

};

int main() {

struct Student student;

// Input student details

printf("Enter registration number: ");

scanf("%s", student.regNumber);

printf("Enter first name: ");

scanf("%s", student.firstName);

printf("Enter last name: ");

scanf("%s", student.lastName);

printf("Enter programming marks: ");

scanf("%f", &student.programming);

printf("Enter networking marks: ");

scanf("%f", &student.networking);

printf("Enter accounting marks: ");

scanf("%f", &student.accounting);

printf("Enter computation marks: ");

scanf("%f", &student.computation);

printf("Enter operating system marks: ");

scanf("%f", &student.operatingSystem);

// Display student details

printf("\nStudent Details:\n");

printf("Registration Number: %s\n", student.regNumber);

printf("First Name: %s\n", student.firstName);

printf("Last Name: %s\n", student.lastName);

printf("Programming Marks: %.2f\n", student.programming);

printf("Networking Marks: %.

2f\n", student.networking);

printf("Accounting Marks: %.2f\n", student.accounting);

printf("Computation Marks: %.2f\n", student.computation);

printf("Operating System Marks: %.2f\n", student.operatingSystem);

return 02.define the following terms

a.compiler;

A specialized program that that translates source codes into machine code

b.source code

A high level programming language used to write a set of instructions

c.object code

output produced by a compiler after translating the source code of a program

d.linkers

A utility that combines multiple objects files generated by a compiler into a single executable programme

3.Using an example ie aprogram to add 2 numbers explain the compilation process of a c program.

#include <stdio.h>

int main() {

int num1, num2, sum;

// Get input from the user

printf("Enter two numbers: ");

scanf("%d %d", &num1, &num2);

// Add the numbers

sum = num1 + num2;

// Display the result

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

return 0;

}

process

1.Editing: You create or modify the source code using a text editor or an integrated development environment (IDE).

2.Preprocessing: The preprocessor is the first step in the compilation process. It handles directives that start with #, such as #include. In this example, #include <stdio.h> tells the preprocessor to include the contents of the stdio.h header file in the program. The preprocessor also handles macro substitutions and conditional compilation.

3.Compilation: The preprocessed code is then compiled into assembly code or an intermediate form by the compiler. This step checks for syntax errors and generates object code.

4.The compiler produces an object file (e.g., main.o) from the source code. However, the object file is not yet executable.

5.Linking: The object file needs to be combined with other necessary object files (e.g., those for library functions like printf), and this process is called linking. The linker resolves references to functions and variables that are defined in other files or libraries.

The final executable file (e.g., a.out or program.exe) is created, and it's ready to be executed.

4.differences btwen a compiler and an interpreter(atleast 6)

a.A compiler processes the entire source code at once and translates it into object code while an interpreter processes the source code line by line during run time

b.compiler execute generally faster as the entire program is already translated into machine code while interpreters execute slower because each line is translated and executed sequentially

c.compiler produces an executable file or binary code that can be run multiple time without recompilation while an interpreter does not produce a separate executable file hence the source code is interpreted each time the program is run

d.In a compiler the program will not run until all errors are resolved while in interpreters the program may execute partially even if there are errors

e.A compiler generally results in a stand alone executable with potentially lower memory overhead during execution while interpreters, the interpreter itself and the interpreted code may consume more memory compared to a compiled program

f.Compiler generates platform specific machine code while an interpreter the source code is interpreted in a platform-independent manner

5.List all categories of operators available in c program under each specify the specifi operator

Arithmetic operators:

+ (Addition)

- (Subtraction)

\* (Multiplication)

/ (Division)

% (Modulus)

Relational Operators:

== (Equal to)

!= (Not equal to)

< (Less than)

> (Greater than)

<= (Less than or equal to)

>= (Greater than or equal to)

Logical Operators:

&& (Logical AND)

|| (Logical OR)

! (Logical NOT)

Assignment Operators:

= (Assignment)

+= (Add and assign)

-= (Subtract and assign)

\*= (Multiply and assign)

/= (Divide and assign)

%= (Modulus and assign)

Increment and Decrement Operators:

++ (Increment)

-- (Decrement)

Bitwise Operators:

& (Bitwise AND)

| (Bitwise OR)

^ (Bitwise XOR)

~ (Bitwise NOT)

<< (Left shift)

>> (Right shift)

Conditional (Ternary) Operator:

? : (Conditional operator)

Miscellaneous Operators:

sizeof (Size of an object)

, (Comma operator)

-> (Member access through pointer)

. (Member access)

& (Address-of)

\* (Pointer dereference)

Special Operators:

sizeof (Size of an object or data type)

& (Address of a variable)

\* (Pointer to a variable)