1. History and Features of C:

1.Who is the father of C language?
Dennis Ritchie is considered the father of the C language.

2. In which year was C language developed?
C was developed in 1972.

3. C was developed at which research center?
C was developed at Bell Labs (Bell Telephone Laboratories).

4. C is a successor of which programming language?
C is a successor of the B language, which was influenced by BCPL.

5. Name two main features of C language.
Two main features are:
Portability
Efficiency and Speed

6. Why is C called a middle-level language?
C is called a middle-level language because it combines the features of high-level languages (like abstraction and structured programming) and low-level languages (like direct memory access and hardware interaction).

7. What are the applications of C language?
Applications include:
Operating systems (like UNIX)
Embedded systems
Compilers and interpreters
System-level programming
Database systems
Game development

8. Why is C called a portable language?
C is called portable because code written in C can be compiled and run on different machines with little or no modification, thanks to its standardization and minimal hardware dependency.

2. Structure of a C Program:

1. What are the main sections of a C program?
A typical C program has the following main sections:
Preprocessor Directives (e.g., #include, #define)
Global Declarations (variables, functions, constants)
main() Function (entry point of the program)
Function Definitions (user-defined or library functions)

2. What is the use of the #include directive?
The #include directive is a preprocessor command that tells the compiler to include the contents of a file (usually a header file) in the program.
For example:

```
#include <stdio.h>  // Includes standard input/output functions
```

3. Why is main() function important in C?
The main() function is the entry point of a C program.
Execution of every C program starts from main(). Without it, the program will not run.

4. Write the general structure of a C program.

```
#include <stdio.h>      // Preprocessor Directive
// Global Declarations
int global_var = 10;
// Function Prototypes
void greet();
int main()              // Main Function
{
   printf("Hello, World!\n");
   greet();           // Function Call
   return 0;
}

// Function Definitions
void greet()
{
   printf("Welcome to C Programming!\n");
}
```

5. What is the difference between header files and source code files?

Header Files (.h)	Source Code Files (.c)
Contain declarations	Contain definitions
Used for function prototypes, macros, constants	Used to implement logic (functions, algorithms)
Example: stdio.h, math.h	Example: main.c, utils.c

6. What is the role of the return 0; statement in main()?

return 0; tells the operating system that the program executed successfully.
A return value of 0 means success
A non-zero value usually means an error or abnormal termination

3. Constants and Variables:

1. What is a variable in C?
A variable in C is a named storage location in memory that can hold a value which can be changed during program execution.
Example:
int age = 25;

2. Define constants in C.
A constant is a value that does not change during the execution of a program. Once defined, it cannot be modified.

3. What is the difference between a variable and a constant?

Variable	Constant
Value can change	Value remains fixed
No keyword required	Requires const or #define
Stored in memory	Stored in memory or replaced at compile time
Example: int x = 5;	Example: const int x = 5;

4. Which keyword is used to define constants in C?
The const keyword is used to define constants.
Also, #define can be used for symbolic constants.
Example:
const float PI = 3.14;
#define MAX 100

5. Give an example of an integer constant.
Example of an integer constant:
100
-45
0

6. What is the difference between symbolic constant and literal constant?

Symbolic Constant	Literal Constant
Named constant defined using #define or const	Actual value written directly in the code
Example: #define PI 3.14	Example: area = 3.14 * r * r;
Easy to change in one place	Hard-coded, must be changed everywhere

7. Can we change the value of a constant during execution?
No, the value of a constant cannot be changed once it is defined.
If you try to modify a constant, the compiler will generate an error.

4. Data Types and Type Conversion:

1. What are the basic data types in C?
The basic data types in C are:

Data Type	Description
int	Integer numbers
float	Floating-point numbers (single precision)
double	Double-precision floating-point numbers
char	Single character
void	Represents absence of value (used for functions with no return type)

2. What is the difference between int and float?

int	float
Used for whole numbers	Used for decimal numbers
Example: int x = 5;	Example: float y = 5.25;
No fractional part	Has fractional part

3. What is the size of char in C?
The size of a char is 1 byte (8 bits).
It can store values from -128 to 127 (signed) or 0 to 255 (unsigned).

4. What is the range of int in C (16-bit compiler)?
On a 16-bit compiler, int is usually 2 bytes, so the range is:
Signed int: -32,768 to 32,767
Unsigned int: 0 to 65,535

5. Define type conversion in C.
Type conversion is the process of converting one data type into another.
There are two types:
Implicit type conversion (automatic by compiler)
Explicit type conversion (manual, by the programmer)

6. What is the difference between implicit and explicit type conversion?

Implicit Type Conversion	Explicit Type Conversion
Done automatically by the compiler	Done manually by the programmer
No data loss (usually)	May lead to data loss
Example: int x = 10; float y = x;	Example: float x = 5.6; int y = (int)x;

7. What is type casting? Give an example.
Type casting is a type of explicit type conversion where you manually convert a variable from one data type to another.
Example:

```
float x = 5.75;
int y = (int)x;   // y becomes 5, decimal part is lost
```

8. What is the difference between signed and unsigned integers?

Signed Integer	Unsigned Integer
Can store both positive and negative values	Can store only positive values
Example: int a = -10;	Example: unsigned int a = 10;
Range is smaller	Range is larger

5. Operators and Expressions:

1. What is an operator in C?
An operator is a symbol that tells the compiler to perform a specific operation on one or more operands (variables or values).
Example: +, -, *, /, ==, etc.

2. List the types of operators in C.
The main types of operators in C are:
Arithmetic Operators (+, -, *, /, %)
Relational Operators (==, !=, >, <, >=, <=)
Logical Operators (&&, ||, !)
Assignment Operators (=, +=, -=, etc.)
Increment/Decrement Operators (++, --)
Bitwise Operators (&, |, ^, ~, <<, >>)
Conditional (Ternary) Operator (? :)
Comma Operator (,)
Sizeof Operator (sizeof)
Pointer Operators (*, &)

3. What is the difference between = and ==?

= (Assignment Operator)	== (Equality Operator)
Assigns a value to a variable	Compares two values for equality
Example: x = 5;	Example: x == 5 (returns true if x is 5)

4. Explain the difference between pre-increment (++i) and post-increment (i++).

Pre-increment (++i)	Post-increment (i++)
Increments the value before use	Increments the value after use
Example: x = ++i; (i is incremented first)	Example: x = i++; (x gets i's old value, then i is incremented)

5. What is the use of the modulus (%) operator?
The % operator returns the remainder of an integer division.
Example:
7 % 3 = 1

6. What is operator precedence in C?
Operator precedence defines the order in which operators are evaluated in an expression.
For example:
int x = 5 + 3 * 2; // Result is 11, because * has higher precedence than +
Operators with higher precedence are evaluated before operators with lower precedence.
Use parentheses () to control evaluation order.

7. What is the difference between logical AND (&&) and bitwise AND (&)?

Logical AND (&&)	Bitwise AND (&)
Used with boolean (true/false) values	Used with binary (bit-level) values
Returns true if both conditions are true	Performs AND on each bit of the operands
Example: (a > 0 && b > 0)	Example: 5 & 3 → 101 & 011 = 001 (1)

8. What is an expression in C? Give an example.

An expression is a combination of variables, constants, and operators that produces a value.
Example:

```
int result = (a + b) * c;
```

6. Input and Output Functions:

1. What is the difference between printf() and scanf()?

printf()	scanf()
Used to display output	Used to take input from the user
Sends data from program to screen	Takes data from user to program
Example: printf("Hello");	Example: scanf("%d", &x);

2. Why do we use format specifiers in C?

Format specifiers are used in printf() and scanf() to tell the compiler what type of data is being input or output.
They match variables to data types.
Example:
%d → int
%f → float
%c → char

3. What is the format specifier for float?

The format specifier for float is %f.

4. Which header file is required for printf() and scanf()?

The required header file is:
#include <stdio.h>

5. Write the syntax of scanf() function.

General syntax:
scanf("format specifier", &variable);
example:
int age;
scanf("%d", &age);

6. How can we take a single character as input in C?

You can use scanf() with %c:
char ch;
scanf("%c", &ch);
Or use getchar():
char ch;
ch = getchar();

7. How can we print multiple values using printf()?

List multiple format specifiers and variables:
int a = 10, b = 20;
printf("a = %d, b = %d\n", a, b);

8. What happens if you don’t use & in scanf()?

If you don’t use &, scanf() will not know where to store the input, because it needs the address of the variable.

This can lead to garbage values or a runtime error (crash).

Example (Wrong):

```
int x;
scanf("%d", x);  // Incorrect
```

Example (Correct):

```
int x;
scanf("%d", &x);  // Correct
```