Tab 2

## **1. History and Features of C**

**Q1. Who is the father of C language?** 👉 Dennis Ritchie

**Q2. In which year was C language developed?** 👉 1972

**Q3. C was developed at which research center?** 👉 AT&T Bell Laboratories, USA

**Q4. C is a successor of which programming language?** 👉 B Language (developed by Ken Thompson)

**Q5. Name two main features of C language.** 👉 Portability and Efficiency

**Q6. Why is C called a middle-level language?** 👉 Because it combines features of both **low-level (machine/assembly)** and **high-level (user-friendly)** languages.

**Q7. What are the applications of C language?** 👉 Operating systems, embedded systems, compilers, databases, game development.

**Q8. Why is C called a portable language?** 👉 Because C programs can run on different machines with little or no modification.

## **2. Structure of a C Program**

**Q1. What are the main sections of a C program?** 👉 Documentation, Preprocessor directives, Global declarations, main() function, and user-defined functions.

**Q2. What is the use of the #include directive?** 👉 To include header files (like stdio.h, conio.h) in a program.

**Q3. Why is main() function important in C?** 👉 Execution of every C program starts from the main() function.

**Q4. Write the general structure of a C program.**

#include <stdio.h>

int main() {

// code

return 0;

}

**Q5. What is the difference between header files and source code files?** 👉 **Header files (.h):** Contain function declarations & macros.  
 👉 **Source code files (.c):** Contain actual program logic/definitions.

**Q6. What is the role of the return 0; statement in main()?** 👉 It indicates successful termination of the program.

## **3. Constants and Variables**

**Q1. What is a variable in C?** 👉 A variable is a named memory location used to store data that can change during program execution.

**Q2. Define constants in C.** 👉 Constants are fixed values that do not change during program execution.

**Q3. What is the difference between a variable and a constant?** 👉 Variable value can change, but constant value remains fixed.

**Q4. Which keyword is used to define constants in C?** 👉 const or #define

**Q5. Give an example of an integer constant.** 👉 10 or -25

**Q6. What is the difference between symbolic constant and literal constant?** 👉 **Symbolic constant:** Defined using #define (e.g., #define PI 3.14).  
 👉 **Literal constant:** Directly written in code (e.g., 3.14).

**Q7. Can we change the value of a constant during execution?** 👉 No ❌

## **4. Data Types and Type Conversion**

**Q1. What are the basic data types in C?** 👉 int, float, char, double, void

**Q2. What is the difference between int and float?** 👉 int stores whole numbers, float stores numbers with decimal points.

**Q3. What is the size of char in C?** 👉 1 byte

**Q4. What is the range of int in C (16-bit compiler)?** 👉 -32,768 to +32,767

**Q5. Define type conversion in C.** 👉 Converting one data type into another.

**Q6. What is the difference between implicit and explicit type conversion?** 👉 **Implicit (type promotion):** Done automatically by compiler.  
 👉 **Explicit (type casting):** Done manually by programmer.

**Q7. What is type casting? Give an example.** 👉 Converting one data type into another manually.  
 Example:

float a = (float)5/2; // Result = 2.5

**Q8. What is the difference between signed and unsigned integers?** 👉 **Signed int:** Can store both positive and negative values.  
 👉 **Unsigned int:** Can store only positive values but with a larger range.

Would you like me to also prepare this in a **well-formatted PDF/notes style** (so you can use for quick revision before exams)?

##### **You said:**

In easy way that i paste in google docs

###### **ChatGPT said:**

Got it 👍  
 Here’s the same content rewritten in **super easy Q–A style** so you can **directly copy-paste into Google Docs** without editing.

# **Basic C Language Questions & Answers**

## **1. History and Features of C**

**Q1. Who is the father of C language?** 👉 Dennis Ritchie

**Q2. In which year was C language developed?** 👉 1972

**Q3. C was developed at which research center?** 👉 AT&T Bell Laboratories, USA

**Q4. C is a successor of which programming language?** 👉 B Language

**Q5. Name two main features of C language.** 👉 Portability, Efficiency

**Q6. Why is C called a middle-level language?** 👉 Because it has both low-level (machine) and high-level (user-friendly) features.

**Q7. What are the applications of C language?** 👉 Operating systems, Embedded systems, Compilers, Databases, Games

**Q8. Why is C called a portable language?** 👉 Because C programs can run on different computers with little or no change.

## **2. Structure of a C Program**

**Q1. What are the main sections of a C program?** 👉 Documentation, Preprocessor directives, Global declarations, main() function, User-defined functions

**Q2. What is the use of the #include directive?** 👉 To include header files (like stdio.h).

**Q3. Why is main() function important in C?** 👉 Program execution starts from main().

**Q4. Write the general structure of a C program.**

#include <stdio.h>

int main() {

// code

return 0;

}

**Q5. Difference between header files and source code files?** 👉 Header files = function declarations,  
 👉 Source files = program logic.

**Q6. What is the role of return 0; in main()?** 👉 It shows program ended successfully.

## **3. Constants and Variables**

**Q1. What is a variable in C?** 👉 A variable is a named memory location to store data.

**Q2. Define constants in C.** 👉 Constants are fixed values that don’t change.

**Q3. Difference between variable and constant?** 👉 Variable = value can change, Constant = value cannot change.

**Q4. Which keyword is used to define constants?** 👉 const or #define

**Q5. Example of an integer constant?** 👉 10, -25

**Q6. Difference between symbolic constant and literal constant?** 👉 Symbolic constant = defined using #define (PI = 3.14)  
 👉 Literal constant = directly written (3.14)

**Q7. Can we change the value of a constant during execution?** 👉 No

## **4. Data Types and Type Conversion**

**Q1. Basic data types in C?** 👉 int, float, char, double, void

**Q2. Difference between int and float?** 👉 int = whole numbers, float = decimal numbers

**Q3. Size of char in C?** 👉 1 byte

**Q4. Range of int in 16-bit compiler?** 👉 -32,768 to +32,767

**Q5. Define type conversion in C.** 👉 Changing one data type into another.

**Q6. Difference between implicit and explicit conversion?** 👉 Implicit = automatic by compiler  
 👉 Explicit = manual using type casting

**Q7. What is type casting? Example?** 👉 Forcing one type into another manually.  
 👉 Example:

float a = (float)5/2; // Result = 2.5

**Q8. Difference between signed and unsigned int?** 👉 Signed int = positive + negative numbers  
 👉 Unsigned int = only positive numbers, larger range

# **5. Operators and Expressions**

**Q1. What is an operator in C?** 👉 An operator is a symbol used to perform an operation on data.

**Q2. List the types of operators in C.** 👉 Arithmetic, Relational, Logical, Assignment, Increment/Decrement, Bitwise, Conditional, Special

**Q3. Difference between = and == ?** 👉 = is assignment operator (x = 5),  
 👉 == is comparison operator (x == 5).

**Q4. Difference between pre-increment (++i) and post-increment (i++)?** 👉 **++i**: increases value first, then uses it.  
 👉 **i++**: uses value first, then increases it.

**Q5. Use of modulus (%) operator?** 👉 Gives remainder of division. Example: 10 % 3 = 1.

**Q6. What is operator precedence in C?** 👉 The order in which operators are evaluated in an expression.

**Q7. Difference between logical AND (&&) and bitwise AND (&)?** 👉 **&&**: Works on conditions (true/false).  
 👉 **&**: Works bit by bit on binary numbers.

**Q8. What is an expression in C? Give an example.** 👉 Expression = combination of variables, constants, operators.  
 👉 Example: a + b \* 2

# **6. Input and Output Functions**

**Q1. Difference between printf() and scanf()?** 👉 printf() = output (display values)  
 👉 scanf() = input (take values from user)

**Q2. Why do we use format specifiers in C?** 👉 To tell the compiler the type of data (e.g., %d for int).

**Q3. Format specifier for float?** 👉 %f

**Q4. Which header file is required for printf() and scanf()?** 👉 #include <stdio.h>

**Q5. Syntax of scanf() function?** 👉 scanf("format specifier", &variable);

**Q6. How can we take a single character as input in C?** 👉 Using scanf("%c", &ch);

**Q7. How can we print multiple values using printf()?** 👉 Example: printf("Name: %s, Age: %d", name, age);

**Q8. What happens if you don’t use & in scanf()?** 👉 It will not store the value in variable → leads to error/wrong result.

# **7. Programs**

**Q1. Print your name, course, and university.**

#include <stdio.h>

int main() {

printf("Name: Rishabh Vashisth Tiwari\n");

printf("Course: B.Tech\n");

printf("University: XYZ University\n");

return 0;

}

**Q2. Print "C is a powerful language" five times.**

#include <stdio.h>

int main() {

for(int i=0; i<5; i++) {

printf("C is a powerful language\n");

}

return 0;

}

**Q3. Demonstrate portability (run on different compilers).**

#include <stdio.h>

int main() {

printf("This program runs on any C compiler!\n");

return 0;

}

**Q4. Display “Welcome to C Programming” using user-defined function.**

#include <stdio.h>

void message() {

printf("Welcome to C Programming\n");

}

int main() {

message();

return 0;

}

**Q5. Two functions: square and cube.**

#include <stdio.h>

int square(int n) { return n\*n; }

int cube(int n) { return n\*n\*n; }

int main() {

int num = 3;

printf("Square: %d\n", square(num));

printf("Cube: %d\n", cube(num));

return 0;

}

**Q6. Function to print roll number and name.**

#include <stdio.h>

void details() {

printf("Roll No: 101\n");

printf("Name: Rishabh\n");

}

int main() {

details();

return 0;

}

**Q7. Declare and print int, float, and char.**

#include <stdio.h>

int main() {

int a = 10;

float b = 3.14;

char c = 'A';

printf("Integer: %d\n", a);

printf("Float: %f\n", b);

printf("Character: %c\n", c);

return 0;

}

**Q8. Demonstrate const keyword.**

#include <stdio.h>

int main() {

const int x = 10;

printf("Constant value: %d\n", x);

return 0;

}

**Q9. Symbolic constant for Pi, calculate area of circle.**

#include <stdio.h>

#define PI 3.1416

int main() {

float r = 5;

float area = PI \* r \* r;

printf("Area of circle: %f\n", area);

return 0;

}

**Q10. Input an int and float, print sum.**

#include <stdio.h>

int main() {

int a; float b;

printf("Enter int and float: ");

scanf("%d %f", &a, &b);

printf("Sum = %f\n", a + b);

return 0;

}

**Q11. Demonstrate implicit type conversion.**

#include <stdio.h>

int main() {

int a = 5;

float b = 2.5;

float result = a + b; // int automatically converted to float

printf("Result = %f\n", result);

return 0;

}

**Q12. Demonstrate explicit type casting (float → int).**

#include <stdio.h>

int main() {

float x = 5.75;

int y = (int)x;

printf("Float: %f\nInt: %d\n", x, y);

return 0;

}

**Q13. Find ASCII value of a character.**

#include <stdio.h>

int main() {

char ch;

printf("Enter a character: ");

scanf("%c", &ch);

printf("ASCII value: %d\n", ch);

return 0;

}

**Q14. Perform all arithmetic operations.**

#include <stdio.h>

int main() {

int a = 10, b = 3;

printf("Addition: %d\n", a+b);

printf("Subtraction: %d\n", a-b);

printf("Multiplication: %d\n", a\*b);

printf("Division: %d\n", a/b);

printf("Modulus: %d\n", a%b);

return 0;

}

**Q15. Check whether a number is even or odd (using modulus).**

#include <stdio.h>

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if(num % 2 == 0)

printf("Even\n");

else

printf("Odd\n");

return 0;

}

**Q16. Demonstrate difference between pre-increment and post-increment.**

#include <stdio.h>

int main() {

int a = 5, b, c;

b = ++a; // pre-increment

printf("After pre-increment: a = %d, b = %d\n", a, b);

a = 5; // reset

c = a++;

printf("After post-increment: a = %d, c = %d\n", a, c);

return 0;

}

**Q17. Calculate simple interest (SI = P × R × T / 100).**

#include <stdio.h>

int main() {

float P, R, T, SI;

printf("Enter Principal, Rate, Time: ");

scanf("%f %f %f", &P, &R, &T);

SI = (P \* R \* T) / 100;

printf("Simple Interest = %f\n", SI);

return 0;

}

**Q18. Evaluate expression: (a + b) \* (c - d).**

#include <stdio.h>

int main() {

int a, b, c, d, result;

printf("Enter values of a, b, c, d: ");

scanf("%d %d %d %d", &a, &b, &c, &d);

result = (a + b) \* (c - d);

printf("Result = %d\n", result);

return 0;

}

**Q19. Input two integers and print sum, difference, product, and quotient.**

#include <stdio.h>

int main() {

int a, b;

printf("Enter two integers: ");

scanf("%d %d", &a, &b);

printf("Sum = %d\n", a+b);

printf("Difference = %d\n", a-b);

printf("Product = %d\n", a\*b);

printf("Quotient = %d\n", a/b);

return 0;

}

**Q19. Input name, age, and percentage of a student, and print formatted output.**

#include <stdio.h>

int main() {

char name[50];

int age;

float percentage;

printf("Enter Name: ");

scanf("%s", name);

printf("Enter Age: ");

scanf("%d", &age);

printf("Enter Percentage: ");

scanf("%f", &percentage);

printf("\n--- Student Details ---\n");

printf("Name : %s\n", name);

printf("Age : %d\n", age);

printf("Percentage : %.2f\n", percentage);

return 0;

}

**Q20. Input radius of a circle and print area and circumference.**

#include <stdio.h>

#define PI 3.1416

int main() {

float r, area, circ;

printf("Enter radius: ");

scanf("%f", &r);

area = PI \* r \* r;

circ = 2 \* PI \* r;

printf("Area = %f\n", area);

printf("Circumference = %f\n", circ);

return 0;

}

**Q20. Input temperature in Celsius and convert to Fahrenheit.**

#include <stdio.h>

int main() {

float c, f;

printf("Enter temperature in Celsius: ");

scanf("%f", &c);

f = (c \* 9/5) + 32;

printf("Temperature in Fahrenheit = %f\n", f);

return 0;

}

**Q21. Input marks of 3 subjects and print the average.**

#include <stdio.h>

int main() {

float m1, m2, m3, avg;

printf("Enter marks of 3 subjects: ");

scanf("%f %f %f", &m1, &m2, &m3);

avg = (m1 + m2 + m3) / 3;

printf("Average = %f\n", avg);

return 0;

}