MCS 201 UNIT – 1 PROGRAMMING FUNDAMENTALS

CHECK YOUR PROGRESS - 1

Question - 1;

Differentiate between flowchart and algorithm.

Answer - 1;

Aspect	**Algorithm**	**Flowchart**
Definition	A step-by-step written procedure to solve a problem.	A graphical representation of an algorithm.
Form	Written using pseudocode or simple English.	Drawn using symbols like ovals, rectangles, diamonds, etc.
Symbol Usage	No special symbols are used.	Uses standard symbols to show logic and flow.
Understanding	Good for planning the steps of a program.	Better for visualizing and explaining logic.
Use	Helps in logic building before coding.	Helps in visually explaining program flow.

Question - 2;

Compute and print the sum of a set of data values.

Answer - 2;

Answer (Algorithm Steps):

1. Start
2. Set the sum of the data values and the count to zero.
3. As long as data values exist, add the next value to sum and increase the count by 1.
4. Compute the average (sum ÷ count).
5. Display the average
6. Stop

Question - 3;

Write the following steps are suggested to facilitate the problem-solving process using computer.

Answer - 3;

a) Define the problem

b) Formulate a mathematical model

c) Develop an algorithm

d) Design the flowchart

e) Code the same using a programming language

f) Test the program

Question - 4;

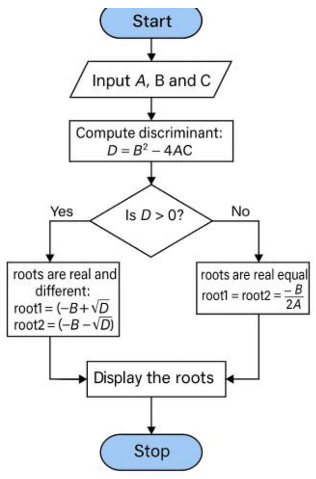
Draw an algorithm and flowchart to calculate the roots of quadratic equation Ax^2 + Bx + C = 0.

Answer - 4;

Algorithm to Calculate Roots of Quadratic Equation:

1. Start
2. Input values A, B, and C
3. Compute discriminant: $D = B^2 - 4AC$
4. If $D > 0$
   → roots are real and different:
   $root1 = (-B + \sqrt{D}) / (2A)$
   $root2 = (-B - \sqrt{D}) / (2A)$
5. If $D = 0$
   → roots are real and equal:
   $root1 = root2 = -B / (2A)$
6. If $D < 0$
   → roots are complex
7. Display the roots
8. Stop

FLOWCHART



MCS 201 UNIT – 1 PROGRAMMING FUNDAMENTALS

Check your progress - 2

Question - 1;

“A Program written in Low Level Language is faster.” Why?

Answer - 1;

A program written in **Low Level Language** is faster because it is **machine-dependent** and **closer to the hardware**. It does **not need conversion** through a compiler or interpreter like high-level languages.

So, it **executes directly** on the machine, making it **faster and more efficient**.

Question - 2;

What is the difference between High Level Language and Low-Level Language?

Answer - 2;

Low level languages express algorithms in the form of numeric or mnemonic codes, whereas **High Level Languages** express algorithms using concise, precise, and unambiguous notation.

Low level languages are **machine dependent**, while High level languages are **machine independent**.

Low level languages are **difficult to program and to learn**, while High level languages are **easy to program and learn**.

Examples of High-level languages are FORTRAN and Pascal.

Examples of Low-level languages are machine language and assembly language

Question - 3;

Why is C referred to as a middle-level language?

Answer - 3;

C is called a **middle-level language** because it combines features of:

- **High-level languages** (easy syntax, structured programming)
- **Low-level languages** (can access hardware, use memory addresses)

It allows both **system-level programming** (like OS, drivers) and **application programming**, making it **powerful and flexible**.

MCS 201 UNIT – 1 PROGRAMMING FUNDAMENTALS

CHECK YOUR PROGRESS – 3

Question – 1

What is the basic unit of a C program?

Solution: The basic unit of a C program is a **C function.**

Question - 2

What is the basic unit of a C program?

Solution: The basic unit of a C program is a **C function.**

Question - 3

indicate the syntax errors in the following program code:

Solution: Based on the errors listed in the solutions, the syntax errors in the (missing) program code were likely: a) # **not present with include**. This suggests a line like include <stdio.h> instead of the correct #include <stdio.h>.

b) **{brackets should be present instead of [] bracket**. This indicates that square brackets [were used where curly braces {} are required to define code blocks for functions, loops, or conditional statements in C (as seen in C code examples provided, e.g., ().

Check Your Progress 4

Question 1

What is the extension of an executable file?

Solution –

The extension of an executable file is **.exe.**

Question 2

What is the need for linking a compiled file?

Solution –

The need for linking a compiled file is that the C program contains many C pre-defined functions present in the C library. These functions **need to be linked with the C program for execution**; else the C program may give a linker error indicating that the function is not present.

Question 3

How do you correct the logical errors in the program?

MCS 201 UNIT – 1 PROGRAMMING FUNDAMENTALS

Solution –

Logical errors can be corrected through **debugging or self-checking**. Debugging is described as the process of removing errors from the program. This means manually checking the program step by step and verifying the results at each step. Debugging can be made easier by a tracer provided in the Turbo C environment. Logical errors are considered the most difficult to correct.