

1. Exp following terms

• Data : Information provided by user to computer.

• Output : Information in another form is presented by the computer after performing a process on it.

• Hardware : Hardware is the machinery itself. It is made up of the physical parts or devices of the computer system like the electronic integrated circuits (ICs), magnetic storage media and other mechanical devices like input devices, output devices etc.

• Input device : They are hardware components that allow users to interact with computers by providing data and instructions. Eg:- keyboards; mice, scanners, cameras, joysticks & microphones.

• Input : the information in one form which is presented to the computer.

• Program : The set of instructions given to the computer to perform various operations.

• Software : It is the set of instructions or programs that tell a computer what to do. It's the intangible part of a computer system that controls its functions.

• Output device : They are hardware components that display or present information from a computer to the user. Eg:- Monitors, printers, speakers & headphones.

• Programming language : A programming language can be defined as a vocabulary and set of grammatical rules for instructing the computer to perform specific tasks. Each programming language has a unique set of characters, keywords and the syntax for organizing programming instructions. Eg :- BASIC, C, C++, Python, Java, SQL.

2. Types of programming language.

- A/q - There are 2 major types,
1. Low level languages
 2. High level languages.

→ Low level languages are machine oriented and require extensive knowledge of computer hardware architecture and its configuration.

It is further divided into → Machine language and
→ Assembly language.

Machine language is the only language that is directly understood by the computer without any translator program. The instructions are called machine instructions and written as strings of 1's & 0's.

Eg:- 1011 0001 11101

Assembly language is a set of mnemonics (symbolic keywords) used to represent machine codes.

Mnemonics are usually combination of words like

ADD, SUB & LOAD etc. In order to execute the program written in assembly language, a translator is required to be called as Assembler.

High level languages. They are simple languages that use English and mathematical symbols like +, -, %, /, for its program construction. In high level languages, they're easier to learn than low-level languages. Compiler is a translator program which converts a program in high level language in to machine language. These are problem oriented languages.

Eg:- COBOL, BASIC, C, C++ etc., FORTRAN.

3. Exp Program development.

- A/q - The various steps involved in Program development are,
1. Defining or Analyzing the program
 2. Design(Algorithm)
 3. Coding

4. Documenting the Program
5. Compiling and running the Program
6. Testing & Debugging
7. Maintenance

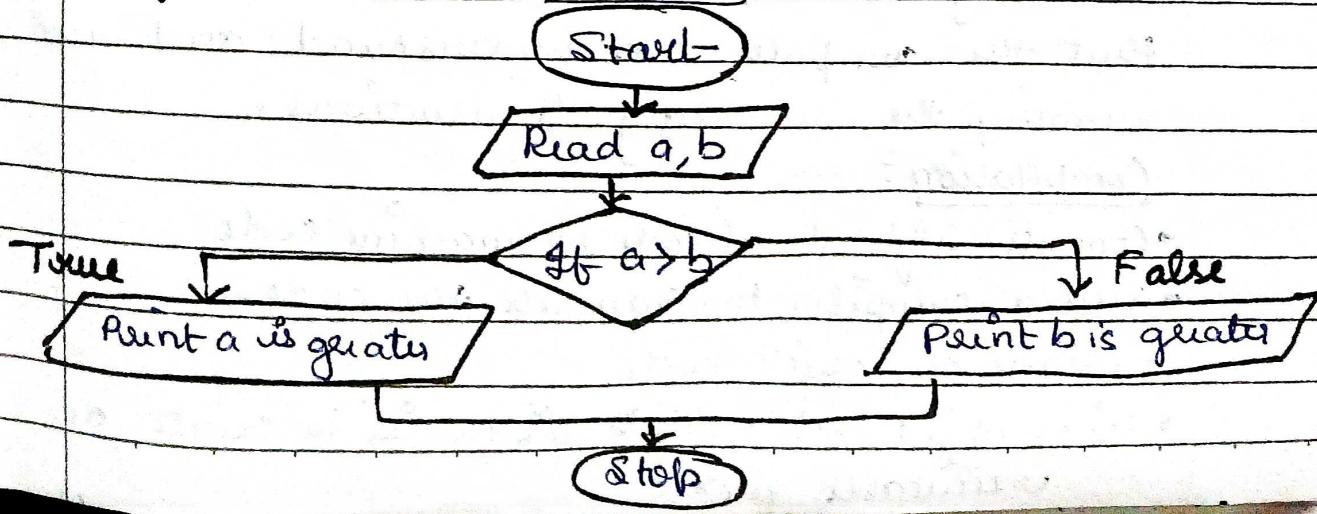
1. → Defining or Analyzing: Clearly understand the problem you want to solve or the task you want to automate.
2. → Design (Algorithm): Plan a step-by-step approach to solve the problem using logical instructions.
3. → Coding: Write the algorithm using a programming language that the computer can understand.
4. → Documenting the program: Exp what the code does and how it works for future reference
5. → Compiling & running: Convert the code into a machine-readable format and execute it.
6. → Testing & Debugging: Check if the program works as expected & fix any errors.
7. → Maintenance: Update the program to address changes or improve its performance over time.

4. What is algorithm? Why is it used. with eg.

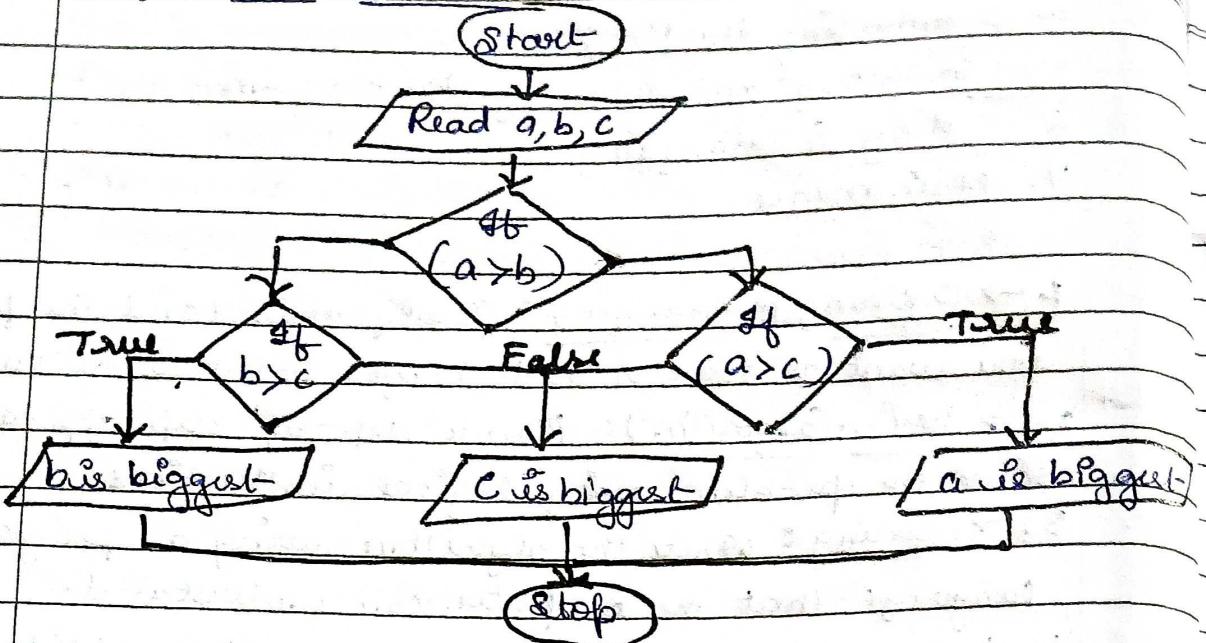
Ans - A set of finite rules or instructions to be followed in calculations or other problem-solving operations. They are used to solve problems and perform tasks efficiently.

Eg:- a search engine uses an algorithm to find the best results for your query.

Largest of 2 numbers Flowchart-



largest of 3 numbers



5. Structure of C program. is defined by set of rules called protocol, to be followed by programmer while writing C program. All C programs are having sections/parts which are mentioned here,
1. Documentation section
 2. Link section
 3. Definition section
 4. Global declaration section
 5. Function prototype declaration section
 6. Main function
 7. User defined function definition section.

6. Process of Compilation & Execution.

These are 2 key phases in the development and running of computer programs. They involve converting human-readable code into a language that the computer can understand and run running the resulting instructions.

Compilation:

- Converts high-level code to machine code
- Use a compiler to translate the code.
- Creates an object code
- Links object code with libraries to create an executable file.

Execution :

- Runs the machine code instructions
- Loads the executable file into memory
- CPU executes the instructions
- Produces output.

7. Characteristics of C language.

Any → Efficiency: C provides direct access to hardware, making it fast

Portability: C code can be easily run on different systems

Structure: C programs are organized into modules & use pointers for memory access.

Flexibility: C can be used for various tasks and offers precise control over hardware.

Simplicity: C has a straightforward syntax and focuses on essential features.

Limitations: C requires manual memory management and lacks some built-in features.

It's a compiled language: This means it needs to be translated into a form the computer can understand before it can run.

It's a popular for system programming: C is often used to write operating systems & other low-level software.

It's efficient: C programs run quickly & use less computer resources.

It's versatile: C can be used for many different kinds of projects, from games to databases.

Variables: Containers for data

Data types: Tell what kind of data is in a variable.

Functions: Smaller programs within a larger program.

Statements: Commands that tell the computer to do something.

Expressions: Combinations of variables, operators, and values.

8.

Ans

What is C token? Explain different types of tokens
→ C tokens are the smallest units in a C program.
• They form the building blocks of the language.
• It describes category of words used in C program.

They are 6 types :

1. Identifier (x in $\text{int } x$)

2. Key words (int, while)

3. Constants constant char letter = n

4. Strings

5. Operators

6. Special symbols.

→ Keywords • Predefined words in C

• Have specific function

• Cannot be used as variable names

• C language supports 32 keywords.

→ Identifiers • Are names given to program elements like variables, functions and arrays.

→ Constants are like variables, but their values cannot be changed after they're set.

• Used for fixed values.

→ Operators: Used for performing certain operations.

Four categories 1. Unary operators

2. Binary operators

3. Ternary operators

4. Special operators

→ Strings: Is a finite sequence of characters • In a C program a string are enclosed with double quotes.

Eg:- "Your name".

9. Rules for constructing a valid Identifier.

- Ans -
- 1. The first character should be a letter or underscore
 - 2. The succeeding character might be either letter or digits.
 - 3. Punctuations and special symbols are not allowed.
 - 4. It should not be a reserved word or keyword.
 - 5. Identifiers are case sensitive (for eg:- Hello & hello are 2 different words in a case-sensitive system)
- [Example just for understanding, no need to write]

10. Different types of valid identifier name.

Ans - Examples :- salary, num, sum, sum
int_num, avg, salary, num
rakesh (underline)

11. Different types of constants.

- Ans -
- ① Integer constants :- Must have digits without decimal point and be either negative or positive.
 - ② Real or floating point constants :- Must have digits with decimal point and can be either negative or positive.
 - ③ Octal and Hexadecimal constants :- These are written in different bases.
 - Numbers starting with 0 are written in Octal.
 - Numbers starting with 0x are written in Hexadecimal.
 - ④ Character constants :- These constants are single characters enclosed in quotes - single quotes.
 - ⑤ String constants :- These constants are single or several characters enclosed in double quote.
 - ⑥ Backslash character constants :- c provides backslash constants to perform specific actions.

Eg:- ' \n ' for newline, ' \t ' for tab space etc.

12. What is Data type? [To specify size & type.]

Simple & sweet

Ans - A data type tells the computer what kind of data a variable can hold and also tells how to store data and work with it.

<u>Data type</u>	<u>Keyword</u>	<u>Size (in bytes)</u>	<u>Range</u>
Integer	int	2	+8276
Real (floating point)	float	4	+3.4E+10
Double precision real	double	8	+1.7E+30
Character	char	1	+127

13. What is data type modifier & except diff data type modifier

Ans. - The storage size and range of values for basic data types can be changed using the following modifiers:

1. Signed : Indicates a signed value (+ve or -ve)
2. Unsigned : Indicates an unsigned value (only +ve)
3. Long : Increases the storage size and range of values
4. Short : Decreases the storage size and range of values.

14. What is variable?

Ans. - Variables are named storage locations that hold data. They can change during program execution. Constants cannot change.

15. Rules for naming C variables.

- Ans. -
- ① Start with a letter or underscore.
 - ② Can contain digits, letters and underscores.
 - ③ Cannot be keywords.
 - ④ Are case-sensitive.
 - ⑤ No special symbols except underscores.

Eg:- • sum, height, _value. are valid examples.
 • 1.height, Height, my_name are invalid examples.

16. How to declare and initialize variable.

Ans. - Variable Declaration:

- Defining the type and name of a variable.

- No memory is allocated

Variable Initialization:

- Assigns a value to a variable at declaration.

TYPE	SYNTAX
Variable declaration	$\Rightarrow \text{data_type variable_name};$ Eg:- int x, y, z;
Variable Initialization	$\Rightarrow \text{data_type variable_name} = \text{value};$ Eg:- int x = 50;

Q7. How to define a symbolic constant in C.

Ans - Symbolic Constants are defined using the #define preprocessor directive. Such preprocessor statements are placed at the beginning of the program and are not part of the C program. They begin with # symbol and do not end with a semicolon.

They replace a variable name with a constant throughout the program.

Syntax : #define CLASS "I Section"

define symbolic_name value

Q8. What is operator? with 4 categories

Ans - Operators in a C programs are used to perform certain operators. There are four categories of operators

1. Unary operators: Works with one operand
2. Binary operators: Works with two operands
3. Ternary operators: Works with three operands
4. Special operators: Used in special cases.

Pre-Increment/decrement: Increment/decrement the operand before using its value.

Post-Increment / decrement: Increment/decrement the operand after using its value.

Bitwise Operators: Perform operations on individual bits of operands (Eg:- AND, OR, XOR).

19. Exp. detailed about Unary operation?

Ans. - Unary operators act on a single operand performing a specific operation on it.

(1) About Pre & Post increment/decrement

(2) Unary minus (-) :- It is used for subtracting, and to change the sign of operand to minus (-).

$$\text{Eg: } x = 10 \Rightarrow -x = -10$$

(3) Logical Not (!) :- It negates the operand if its true then it returns false.

It reverses the truth value of a proposition.

If Input is true, the output is false.

If Input is false, the output is true.

(4) Bit wise Complement (\sim) :-

It is used to change 1 to 0 and 0 to 1.

$$\text{Eg: } \sim 8 = 7$$

20. What is expression?

Ans. - Expressions are basically operators acting on operands.

There are rules which involve precedence (order of operations) and associativity (grouping of operands).

Q. Write a note on Precedence & Associativity.

Ans. - Precedence : The order in which different operators are used to evaluate an expression is called precedence of operators or hierarchy of operators. The operators are evaluated from highest precedence to least precedence.

Associativity : whenever two or more operators having the same precedence or priority occurs in the expression then the direction order chosen (left to right or right to left) to evaluate an expression is called associativity of operators.

21. What is type conversion? and explain its 2 types.

Ans. - The process of converting one type of data type to another, say change int to float or float to int.

is known as data type conversion or type casting.

Data type conversion can be done in 2 ways:

1. Implicit type conversion

Eg:- $\text{int } x = 5;$

2. Explicit type conversion

$\text{double } y = x;$

Eg:- $\text{Double } z = 3.14;$

$\text{int } w = (\text{int}) z;$

① Implicit type Conversion: is when the compiler automatically changes data types in an expression. It's also called "automatic type conversion". The compiler follows a specific hierarchy to determine the new type.

② Explicit type conversion: is when the user manually tells the program to change data types in an expression. This is done using a cast operator.

Q2. What is decision making statement & exp 2 types.

Ans. — Generally C program execute its statements sequentially. But in order to solve problems we may have some situations where we have to change the order of executing the statements based on whether some conditions have met or not.

So, controlling the execution of statements based on certain condition or decision is called decision making and branching.

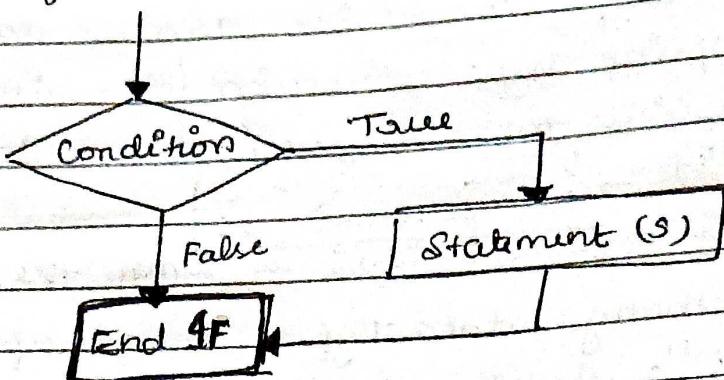
There are of 2 types:-

1) Conditional Branching control statement:

- Simple if statement
- if else statement
- else if ladder
- nested if
- switch statement

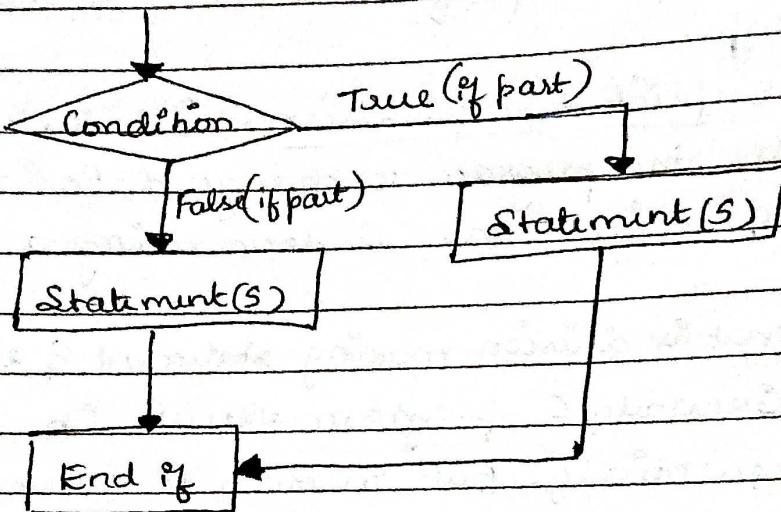
1

Flowchart of simple If



2

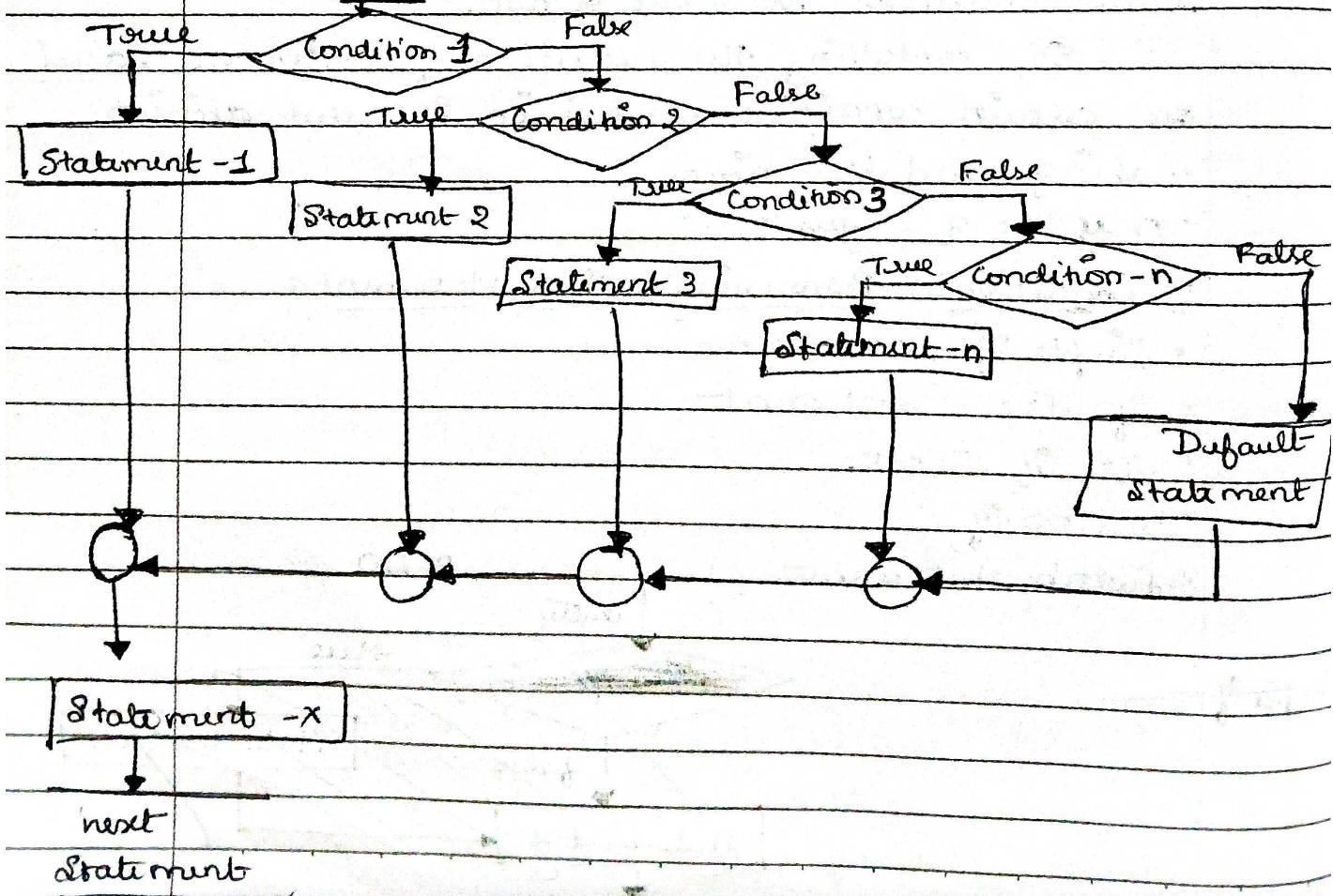
If else



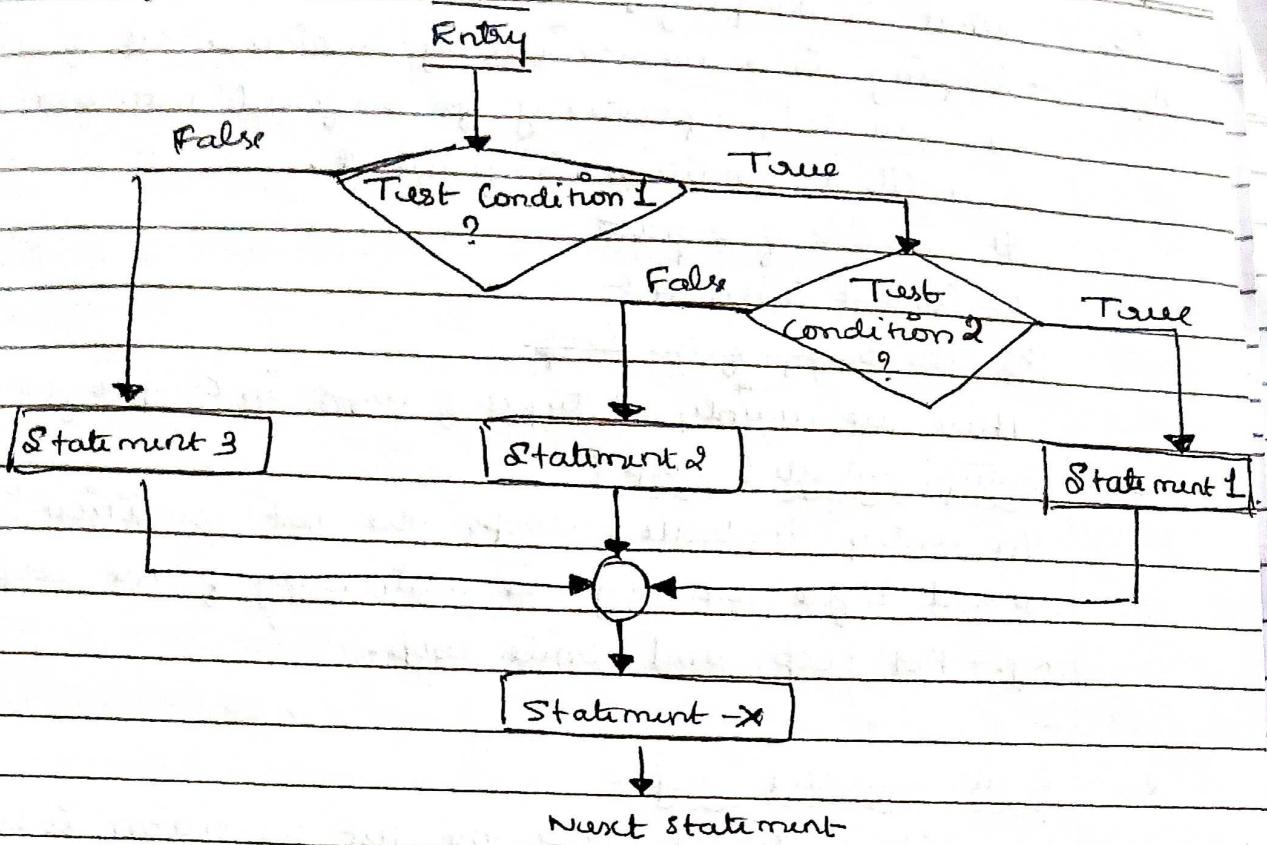
3

else If ladder

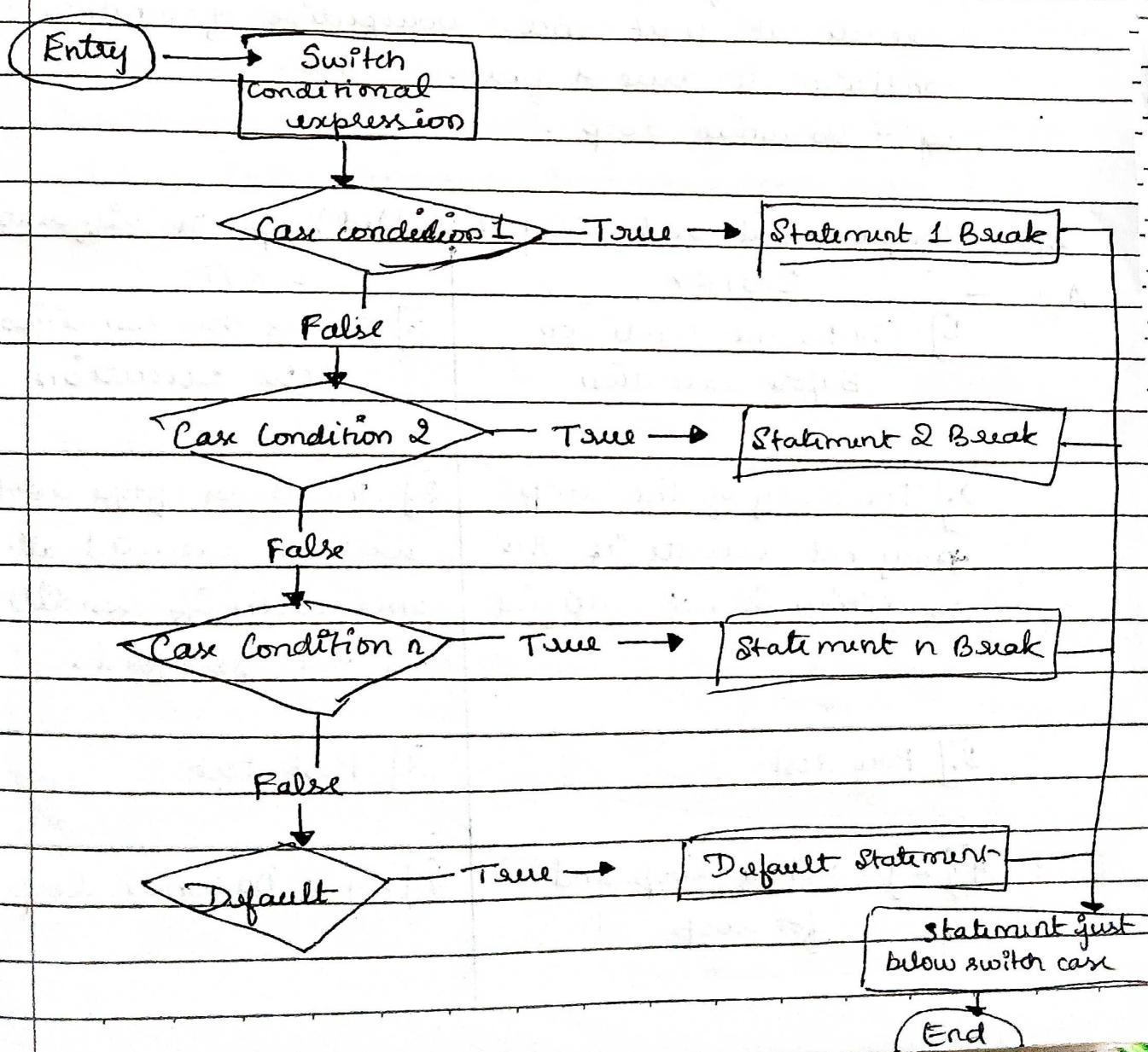
ENTRY



(A) nested if



(5) Switch statement.



29]

Unconditional branching statements:

- Goto statement : which allows the flow of execution of statements to be altered without any conditional expressions. It transfers the control from one point to another.

Syntax : goto label;

23. What is looping?

Ans - Looping is a mechanism by which a set of instructions are executed repeatedly for a fixed number of times or until a condition is fulfilled.

It consists of 2 parts

(1) Control Statement

(2) The Body of the loop.

There are mainly 3 types of loops in C programming:

1. Entry controlled loop:

In entry controlled loops the test condition is checked before entering the main body of the loop.

Eg:- For loop and while loop.

2. Exit controlled loop:

In exit controlled loops the test condition is evaluated at the end of the loop body. The loop body will execute at least once, irrespective of whether the condition is true or false.

Eg:- Do-while loop.

24. Compare exit and entry controlled loop. Or Differentiate.

Ans -

ENTRY

EXIT

1.] Checks the condition
Before execution

1.] Checks the condition
After execution

2.] The Body of the loop
may not execute if the
condition is not satisfied

2.] The Body of the loop
will be executed atleast
once even if condition
is not satisfied.

3.] Pre test

3.] Post test

4.] Eg:- While loop and
for loop

4.] Eg:- Do while loop

Q5. Write a note on Jumps in loop using break & continue.

Ans - Unconditional statements are the statements which transfer control or flow of execution unconditionally to another block of statements. These are also called as Jump statements.

→ Break and continue are keywords used to modify the behaviour of loops in programming. They provide ways to terminate or skip iterations within a loop.

Break statement : Exits the innermost loop it's contained within. When a break statement is executed, the loop terminates immediately, and control is transferred to the statement following the loop.

continue statement : Skips the remaining statements in the current iteration of a loop and proceeds to the next iteration.

There are 2 types of Jumping's

① Forward jumping: This occurs when control is transferred to a statement that appears later in the program's execution order. This is commonly used to skip sections of code or to break out of loops.

② Backward jumping: This occurs when control is transferred to a statement that appears earlier in the program's execution order. This is often used to create loops or to implement recursive functions.

Q6. Explain formatted input output statement that is printf and scanf.

Ans - Formatted input function → scanf()

Formatted output function → printf()

→ The scanf function is used to read formatted input from the user. It takes a control string and a list of addresses as arguments. The control string specifies the format of the input data, and the addresses indicate where the data should be stored.

Syntax : scanf ("control string", address-list);
or

scanf ("control string", &var1, &var2, ... (&varn));

→ The printf is a function is used to print formatted output to the screen. It takes a format string and variable values as arguments and displays the values in the specified format.

Syntax : printf ("control string", var-list);
or

printf ("control string", arg1, arg2, ..., argn);