Chapter No. 5: Data Representation

Q1. What is meant by 'data' and 'information'? Differentiate 'data' and 'information' with example.

Ans. **Data:** A Collection of raw facts and figures is called data. It is not in a meaningful form. e.g. the sentence 'being into Pakistan came 14th August on.' is data as it is not meaningful.

Information: Arranged, processed or meaningful form of data is called information. e.g. the above sentence can be arranged as 'Pakistan came into being on 14th August.' is in meaningful form which conveys meaning.

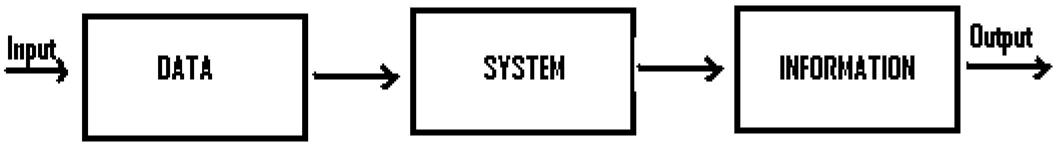


Fig: Data is converted into information by the computer.

Q2. What are different types of data? Explain with example.

Ans. **Types of Data:** Data can be of 'Numeric', 'Alphabetic', 'Alpha-numeric', and 'Multimedia' type.

i. **Numeric Data**: Numeric data consists of numbers such as 6,7, 6.5, 4, -23 etc. It consists of integer and real data.
   a. **Integer Data**: Integer data consists of positive or negative whole numbers without decimal point e.g. +3, -2, +67 etc.
   b. **Real Data**: Real data consists of numbers which may be fractions such as 15.4, 6.7 etc.

ii. **Alphabetic Data:** Alphabetic data consists of combination of capital or small English alphabets e.g. Islamabad, Aero Asia etc.

iii. **Alphanumeric Data:** Alphanumeric data consists of a combination of alphabets and numbers e.g. F-16, I/10-3 etc.

iv. **Multimedia Data:** It includes images, sounds, videos etc.

Q3. What are different number systems? Briefly explain.

Ans. **Number Systems:** There are four common number systems; decimal, binary, octal and hexadecimal.

1. Decimal Number System: It consists of ten digits from 0 to 9 and their combinations. Its base is 10.

2. Binary Number System: It is the most important number system for digital computers. It consists of only two digits; 0 and 1 and their combination. The base of binary number system is 2 as it has only two digits.

3. Octal Number System: It consists of eight digits from 0 to 7 and their combination. The base of octal number system is 8. Each octal digit can be represented as a group of three binary digits.

Octal Number	Binary Number		
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

4. Hexadecimal Number System: It is a number with sixteen digits from 0 to 15(F) or their combination. The base of hexadecimal number system is 16. One hexadecimal digit can be represented as a group of four binary digits.

Hexadecimal Number	Decimal Number	Binary Number			
0	0	0	0	0	0
1	1	0	0	0	1
2	2	0	0	1	0
3	3	0	0	1	1
4	4	0	1	0	0
5	5	0	1	0	1
6	6	0	1	1	0
7	7	0	1	1	1
8	8	1	0	0	0
9	9	1	0	0	1
A	10	1	0	1	0
B	11	1	0	1	1
C	12	1	1	0	0
D	13	1	1	0	1
E	14	1	1	1	0
F	15	1	1	1	1

Q4. What is the importance of machine language or binary language in computers.

Ans. <u>Importance of Binary Numbers and Machine Language in Computers</u>: Digital Computers are built from electronic circuits with only two states of switches; ON and OFF. These two states are represented by 0 (OFF) and 1 (ON). Therefore, binary language is used in computers as a mother language or machine language.

Q5. What is meant by code?

Ans. <u>Code</u>: A single **b**inary dig**it** is called a bit. Information in computers is represented as group of bits. Each group of bits is called a code.

Q6. What is meant by coding in computers?

Ans. <u>Coding</u>: Digital Computers understand binary language only. Therefore, all numeric and non-numeric data is converted into binary form. The process of converting numeric and non-numeric data into binary form is called coding.

Q7. What are different codes used in computers? Explain.

Ans. <u>Codes used in computers</u>:

1. **Binary Coded Decimal (BCD):** In Binary Coded Decimal (BCD) code, each decimal digit is represented as a group of four binary digits.
   e.g. 9 4 8 = 1001 0100 1000

2. **The ACSII (American Standard Code for Information Interchange) Code:** ACSII is a seven bit code to handle alphanumeric data and standardize I/O devices such as keyboards, monitors, printers etc. An 8-bit ASCII code is also available.

3. **EBCDIC (Extended Binary Coded Decimal Interchange Code):** It is also an 8-bit code which can handle 256 different characters. It is mainly used by IBM computers.

Q 5.06 Describe the importance of binary number system.

Ans. Look Q4 of this chapter.

Q 5.07 Convert the following numbers into their binary equivalent.
Solution.

(i) 47

2	47	
2	23	1
2	11	1
2	5	1
2	2	1
2	1	0
	0	1

= (101111) Ans.

(ii) 167

2	167	
2	83	1
2	41	1
2	20	1
2	10	0
2	5	0
2	2	1
2	1	0
	0	1

=(10100111) Ans.

Chapter No. 6: Boolean Algebra

Q1. What is meant by Boolean Algebra?

Ans. Boolean Algebra:-

Boolean Algebra is a branch of knowledge which uses algebraic notation to express logical relationships. It is also called two-valued-algebra or switching algebra because it uses only two states ON/OFF, TRUE/FALSE or 0/1. The founder of Boolean Algebra was George Boole (1806-1876).

Q2. What are the elements of Boolean Algebra?

Ans. Elements of Boolean Algebra:-

In Boolean Algebra, an expression is a combination of variables, constants and logical operators.

1) Boolean Constants:

In Boolean Algebra a set of constants has only two elements 0 or 1.

2) Boolean Variables:

In Boolean Algebra, variables are represented by capital or small English alphabets e.g. A, B, C ….. a, b, c …. etc.

Boolean variables can assume only two values 0 or 1.

3) Logical Operators:

In Boolean Algebra, there are three basic operators; AND, OR, NOT.

i) **NOT Operation**:

NOT operation is a unary operation i.e. it operates only one input. It negates the input. If the input is zero, the output will be one or if the input is one the output will be zero.

Truth Table:

X	NOT(X)
0	1
1	0

ii) **OR Operation:**

OR operation means logical addition. It is represented by a plus sign. It operates two inputs. The output is TRUE(1), if one of the inputs is TRUE(1).

Truth Table:

A	B	A OR B
0	0	0
1	0	1
1	1	1
0	1	1

iii) **AND Operation:**

AND operation means logical multiplication. It is represented with or without a dot between two variables. It operates two inputs. The output is TRUE (1), if both of the inputs are TRUE(1).

Truth Table:

A	B	A AND B
0	0	0
1	0	0
1	1	1
0	1	0

Q3. What is meant by operator precedence?

Ans. The order in which logical operations are evaluated is called 'operator precedence'.

Operator Precedence:

i) Expressions are evaluated from left to right.
ii) Parentheses are evaluated first.
iii) NOT operations are performed after parenthesis.
iv) AND operations are performed after NOT.
v) OR operations are performed at the end.

Q4. Describe different laws of Boolean Algebra.

Ans. Laws of Boolean Algebra:

Axiom 1(Existence of identity element)

a) A + 0 = A
b) A.1 = A

Axiom 2 (Commutative Law)

a) A + B = B + A
b) A . B = B . A

Axiom 3 (Associative Law)

a) A + (B + C) = (A+ B) + C
b) A . (B.C) = (A.B).C

Axiom 4 (Distributive Law)

a) A . (B + C) = A.B + A.C
b) A+(B.C) = (A + B) . (A + C)

Axiom 5 (Existence of Inverse)

a) A + A’ = 1
b) A . A’ = 0

Q5. What is a truth table?

Ans. Truth Table:

A truth table is a table which shows the result of a Boolean expression for all possible input combinations.

Truth Table of NOT:

X	NOT(X)
0	1
1	0

Truth Table of OR:

A	B	A OR B
0	0	0
1	0	1
1	1	1
0	1	1

Truth Table of AND:

A	B	A AND B
0	0	0
1	0	0
1	1	1
0	1	0

Q6. State and prove the Idempotent Law.

Ans. Statement and Proof of Idempotent Law:

a) A + A = A
b) A . A = A

a) A + A = A

Proof:

L.H.S. = A + A
= (A + A) . 1
= (A + A) . (A + A)
= A + (A.A)
= A + 0
= A = R.H.S …………Hence proved.

b) A . A = A

Proof:

L.H.S = A . A
= A . A + o

= A . A + A. A
= A . (A + A)
= A . 1
= A = R.H.S….………Hence proved.

Q7. State and prove the Absorption Law.

Ans. <u>Statement and Proof of Absorption Law:</u>

c) A + (A . B) = A
d) A . (A+B) = A

a) A + (A . B) = A

Proof:

L.H.S. = A + (A.B)
= (A.1) + (A.B)
= A (1 + B)
= A .1
= A = R.H.S …………Hence proved.

b) A . (A + B) = A

Proof:

L.H.S = A .(A + B)
= (A +0). (A + B)
= A + (0.B)
= A + 0
= A = R.H.S….………Hence proved.

Q8. State and prove the Involution Law.

Ans. <u>Statement and Proof of Involution Law:</u>

<u>Involution Law:-</u> A’’ = A

<u>Proof:</u>

This can be proved by ‘perfect induction method’

A	A’	A’’
0	1	0
1	0	1

Q9. State and prove the ‘Demorgan’s Law’.

Ans. <u>Statement and Proof of Demorgan’s Law:</u>

a) (A + B)’ = A’ . B’
b) (A . B)’ = A’ + B’

a) (A + B)’ = A’.B’

Proof:

L.H.S = (A + B)’
= A’ +’ B’
= A’ . B’ = R.H.S ……….Hence proved

b) (A . B)’ = A’ + B’

Proof:

L.H.S = (A . B)’
= A’ .’ B’
= A’ + B’ = R.H.S……….Hence proved

Q 6.05 What is Boolean Algebra?

Ans. Look Q1

Q 6.06 What do you understand by logical operations AND and OR?
Look Q3 (3-ii & 3-iii)

Q 6.07 Find the values of Boolean Expressions.

i) XY + XY’ when X=1 and Y=0

= 1.0 + 1.0’
= 0 + 1.1
= 0 + 1
= 1 Ans.

ii) (X + Y) . (XY) when X=1 and Y=0

= (1 + 0).(1.0)

= (1).(0)

= 0 Ans.

iii) (X+Y’) . (X’ + Y) when X=1 and Y=1

= (1+1’).(1’+1)

= (1+0)(0+1)

= 1 . 1

= 1 Ans.

Q 6.08 State and prove the two basic Demorgan’s theorem. Find out the compliments of the following Boolean expressions.

Ans. <u>Statement and Proof of Demorgan’s Theorems:</u>

Look Q9

<u>Compliments:</u>

i) XY + XY’

= (XY + XY’)’

= (X.Y)’ +’ (XY’)’

= (X’ + Y’) . (X’+Y’’)

= (X’ + Y’).(X’+Y) Ans.

ii) (X + Y). (XY)

= [(X + Y).(XY)]’

= (X + Y)’ + (X.Y)’

= (X’.Y’) + (X’+Y’) Ans.

iii) (X + Y’) . (X’ + Y)

= [(X + Y’).(X’+Y)]’

= (X+Y’)’ + (X’ + Y)’

= (X’.Y’’) + (X’’.Y’)

= X’.Y + X.Y’ Ans.

Q 6.09 What is a truth table? Construct a truth table for AND and NOT of AND operation for the three variables X, Y and Z.

Ans. <u>Truth Table:</u>

Look Q5

<u>Truth Table of AND and NOT of AND Operation</u>

X	Y	Z	X.Y.Z (AND)	(X.Y.Z)’ (NOT of AND)
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Q 6.10 State and prove the following laws:
Ans. Look Q6, Q7, Q8

Q 6.11 Construct a truth table for the following Boolean expressions.
Ans. I) XY + X’Z + YZ

X	Y	Z	X’	XY	X’Z	YZ	XY+X’Z+YZ
0	0	0	1	0	0	0	0
0	0	1	1	0	1	0	1
0	1	0	1	0	0	0	0
0	1	1	1	0	1	1	1
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	0	1	0	0	1
1	1	1	0	1	0	1	1

II) (X+Y) . (X.Y)

X	Y	(X + Y)	XY	(X+Y).(X.Y)
0	0	0	0	0
0	1	1	0	0
1	1	1	1	1
1	0	1	0	0

III) XY’ + XZ + YZ’

X	Y	Z	Y’	Z’	XY’	XZ	YZ’	XY’+XZ+YZ’
0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	0	1	0	0	1	1
0	1	1	0	0	0	0	0	0
1	0	0	1	1	1	0	0	1
1	0	1	1	0	1	1	0	1
1	1	0	0	1	0	0	1	1
1	1	1	0	0	0	1	0	1

Q 6.12 Simplify the following Boolean expressions.

Ans. I) A’.C + A’.B + AB’C + BC

= A’C(B+B’) + A’B(C+C’) + AB’C + BC(A+A’)
= A’BC+A’B’C + A’BC+A’BC’ + AB’C+ ABC +A’BC

= A’BC +A’B’C+A’BC’+AB’C+ABC
= A’BC+A’BC’ +A’B’C+AB’C +ABC
= A’B(C+C’) + B’C(A+A’) + ABC
= A’B .1 + B’C.1 +ABC
= A’B + B’C +ABC Ans.

II) XY’Z + XY’Z’ +X’Y’Z+ X’Y’Z’
= XY’Z + X’Y’Z + XY’Z’+X’Y’Z’
= Y’Z(X+X’) + Y’Z’(X+X’)
= Y’Z.1 + Y’Z’.1
= Y’Z + Y’Z’
= Y’(Z+Z’)
= Y’.1
= Y’ Ans.

III) (A + B + C).(A + B’ + C’).(A + B + C’).(A + B’ + C)

Let T_1= (A+B) , T_2= (A+B’). So, we ca rewrite the expression as under:-

= (T1+C).(T2+C’).(T1+C’).(T2+C)

= [(T1+C).(T1+C’)].[(T2+C’).(T2+C)]

= [T1+(C.C’)].[T2+(C.C’)]

= [T1+0].[T2+0]

= (T1).(T2)

Replacing the values of T1 and T2.

= (A+B).(A+B’) Ans.

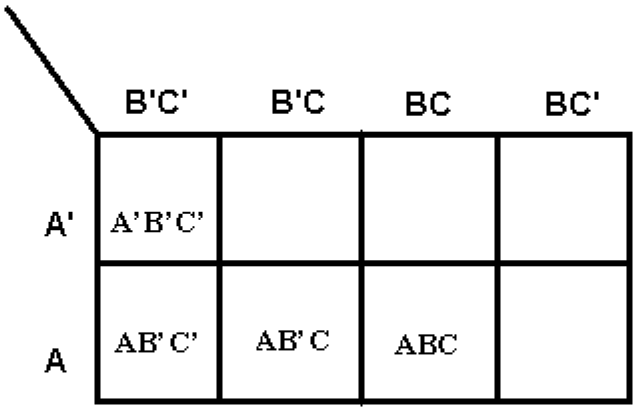
Q 6.13 Simplify the following using Karanaugh map.

Ans.

I) AB’C + ABC + B’C’

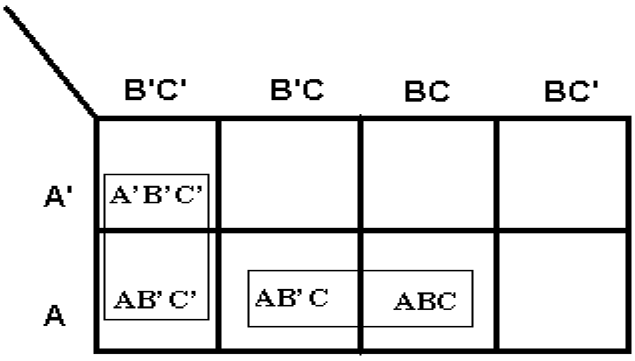
Solution:-

Step1: Draw a K’map of three variables and place values in relevant blocks.



	B'C'	B'C	BC	BC'
A'	A’B’C’			
A	AB’C’	AB’C	ABC	

Step2: Make groups of adjacent blocks (a group consists of 2 or 4 blocks)



	B'C'	B'C	BC	BC'
A'	A’B’C’			
A	AB’C’	AB’C	ABC	

Group1:
A'B'C'
AB'C'

=B'C'

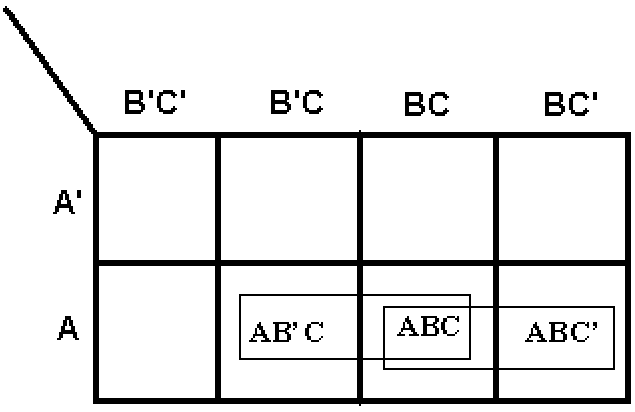
Group2:
AB'C
ABC

= AC

= B'C' + AC

II) ABC + AB’C + ABC’

Solution:



	B'C'	B'C	BC	BC'
A'				
A		AB’C	ABC	ABC’

Group1:
AB'C
ABC

=AC

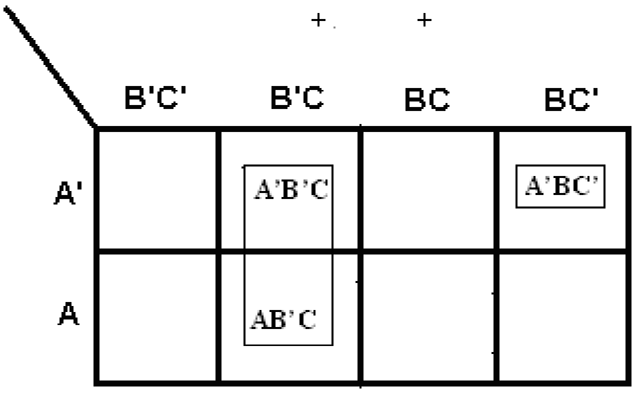
Group2:
ABC
ABC'

=AB

= AC+AB

III) AB’C + A’BC’ + A’B’C

Solution:



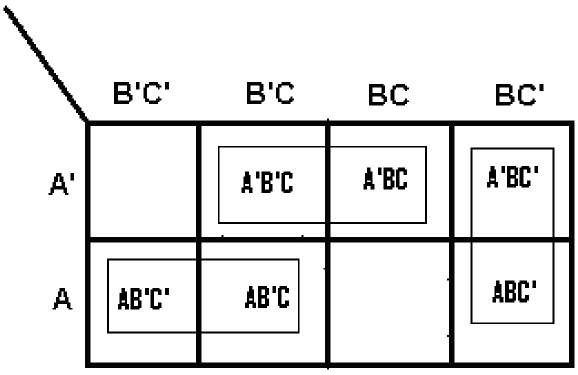
	B'C'	B'C	BC	BC'
A'		A'B'C		A'BC'
A		AB'C		

Group1
A'B'C
AB'C
=B'C

Group2
A'BC'

IV) AB' + A'C + BC'

Solution:



	B'C'	B'C	BC	BC'
A'		A'B'C	A'BC	A'BC'
A	AB'C'	AB'C		ABC'

Group1
A'B'C
A'BC
=A'C

Group2
A'BC'
ABC'
=BC'

Group3
AB'C'
AB'C
=AB'

=A'C + AB' + BC'

Chapter No. 9: Problem Solving

Q1. What is problem solving?

Ans. **Problem Solving:**

Problem solving means computer programming. So the process of developing software using software engineering disciplines is called ‘problem solving’. There are two crucial steps in problem solving:

- To ensure that correct problem in being solved.
- To develop a correct logic to solve the problem.

Q2. What are different steps involved in problem solving?

Ans. **Steps involved in problem solving:**

Computer programming is a very difficult job. Quality programs can only be developed if there is proper planning. There are many phases(steps) in problem solving:

1. **Problem Identification Stage**

   The first stage in program development is to ‘identify the problem’. It includes two steps:

   a) **Defining the problem:** This is the first step in program preparations. Errors made here cannot be corrected in the later stages. In this step, the objectives and other important factors of the problem are written. Following steps are taken:
      - Problem is studied carefully.
      - Is the solution possible.
      - Identify inputs and outputs.
      - Note important requirements of the problem.

   b) **Analyzing the problem:** In this step, problem is analyzed and the following questions are answered:
      - How to solve the problem?
      - Can it be solved on computer?
      - What are inputs and outputs?
      - How many solutions are possible?
      - Which one is the simplest and efficient solution?

2. **Planning Stage**

   This phase includes the following jobs:
   - Developing flowcharts.
   - Developing algorithms.
   - Writing pseudo codes.

3. **Coding Stage**

   The process of writing computer programs in a computer language is called ‘coding’. Flowcharts, algorithms and pseudo codes are converted into computer programs in a specific language.

4. **Testing and Debugging Stage**

   There is always a chance of many errors in the program after coding it. So, it is necessary to test the program. In this step, the program is executed and tested with many inputs. In case of errors, an arror report is made.

   Afterwards, these errors are corrected. The process of removing errors from a program is called debugging.

5. **Implementation and Documentation Stage**

   In this stage, the program is delivered to the client and it is implemented on the computers. Documentation is also made which tells how to use the software.

Q3. What is an algorithms? What are its characteristics?

Ans. **Algorithm:** An algorithm is a step by step procedure to solve any problem.

Characteristics of an Algorithm:

- Every step should be precise and clear.
- Each step should be performed finite number of times.
- There should be some output.

Q4. What is a flowchart? Explain different symbols used in a flowchart?

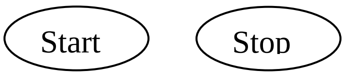
Ans. **<u>Flowchart:</u>**

A flowchart is a pictorial or graphical representation of an algorithm. A flowchart shows the flow of control of an algorithms or program.

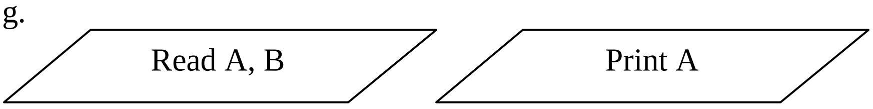
<u>Symbols used in flowcharting:</u>

a) **Flow line (→ ↑):** It is a line with an arrow head. It shows the direction of flow of control. It connects different flowcharting symbols.

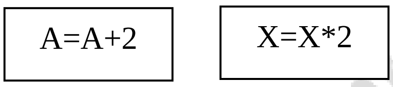
b) **Terminal Symbol(⬭):** It is an oval shape. It is used to begin or end a flowchart.
e.g.



c) **Input/ Output Symbol (▱):** A parallelogram represents the Input / Output operations in a flowchart.
e.g.



d) **Processing Symbol (▭):** A rectangular represents processing in a flowchart.
e.g.



e) **Decision Symbol (◇):** A diamond represents decision in a flowchart. It has one incoming flow line and two outgoing flow lines. It is labeled with a condition. Each outgoing flow line is labeled with 'yes' or 'no'. If the condition inside the decision box is true then the control is transferred to the arrow labeled 'yes'. Otherwise the control is transferred to the arrow labeled 'no'.

f) **Off-page Connector Symbol ( ):** If the flowchart consists of more than one page then connector symbol is used. Control is transferred from one connector to the other labeled with the same number.

Q5. What are different types of errors? Explain each of them with example.

Ans. The errors in a program are of generally three types:-

a) **<u>Logical Errors</u>**:-

Logical errors are normally due to the wrong use of formula or wrong use of symbol. A computer cannot detect such errors. These errors are detected by the user by executing the program and giving test data.

Example: As we know that:

Speed= distance / time

but if a programmer writes:

speed = distance x time

there exists a logical error.

b) <u>Syntax Errors</u>:-

Syntax errors are due to the wrong use of programming language. This includes incorrect punctuation, incorrect word sequence or misuse of terms.

Example: Consider the expression

C=a+/B in this expression there exists a syntax error.

c) **<u>Execution Errors / Run-time Errors:-</u>**

Execution errors arise due to the software or hardware limitation. Execution errors are difficult to locate. They occur at the run time or after execution.

Q6. What is meant by documentation? Discuss its importance. What are different types of documentation?

Ans. <u>Documentation:</u>

A program is useless for the user if it is not well-documented. Documentation is a document which explains the software and its use. It also works as a catalog. Documentation is necessary for the proper use of any software.

Types of Documentation: There are two types of documentation:-

a) User Documentation:- It is designed for the users of the program. It may include the following:-

- Discussion of the problem.
- How the problem is solved.
- Whats the output.
- Required data.
- List of problems that may be encountered by the user.

b) Technical Documentation:- It helps the operators to execute the program. It is also used by the analysts if there is a need of improvement in the program. It includes flowcharts, structure charts, coding forms and other materials that were used during design.

Q 9.05 Explain the importance of problem definition in computer with some suitable example.

Ans. **Importance of Problem Definition:** This is the first step in program preparations. Errors made here cannot be corrected in the later stages. In this step, the objectives and other important factors of the problem are written. So, this is one of the most important steps in program preparation.

Example: Suppose, you are given a result card of 30 students of class X. Find out the number of students place in grade A, B or C respectively.

Solution:

Problem Definition:
"Finding out the number of students placed in A, B or C grades."

Q 9.06 Describe various steps necessary to solve a problem.

Ans. Look Q2

Q 9.07 What is an algorithm? What are the characteristics necessary for a set of instructions to qualify as an algorithm?

Ans. Look Q3

Q 9.08 Write the algorithm for the conversion of temperature from celcius scale to Fahrenheit scale.

Ans. **Algorithm for the Conversion of Temperature from Celcius Scale to Fahrenheit Scale**

```
Algorithm farenheit-to-celcius
Input:      Temperature in Celcius scale.
Output:     Temperature in Fahrenheit scale.

Step 1:     Begin
Step 2:     Read temperature in Celcius scale in variable 'C'
Step 3:     Calculate the temperature in Fahrenheit scale
            F=(9/5 x C)+32
Step 4:     Output the temperature in Fahrenheit scale from variable
            'F'.
Step 5.     Stop
```

Q 9.09 What is a flowchart? What are its advantages?

Ans. **Flowchart:**

A flowchart is a pictorial or graphical representation of an algorithm. A flowchart shows the flow of control of an algorithms or program.

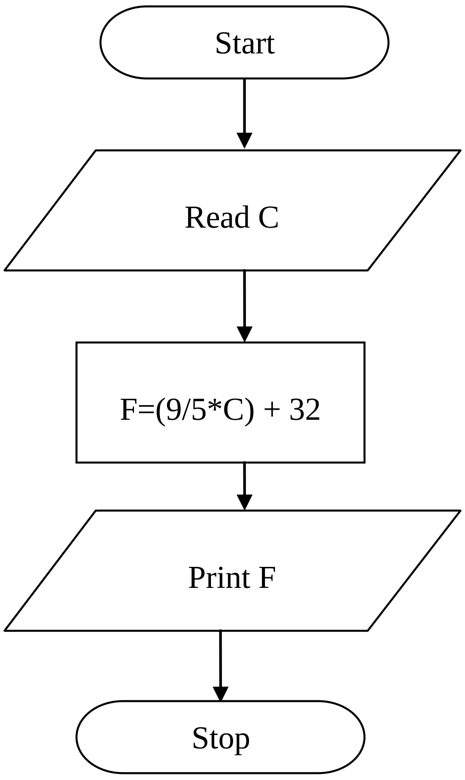
Advantages of a Flowchart:

1. It demonstrates clearly the logical flow of the computer program.
2. It can help a person to review the program.
3. It expresses the basic structure of the program.

Q 9.10 Describe four symbols used in a flowchart.

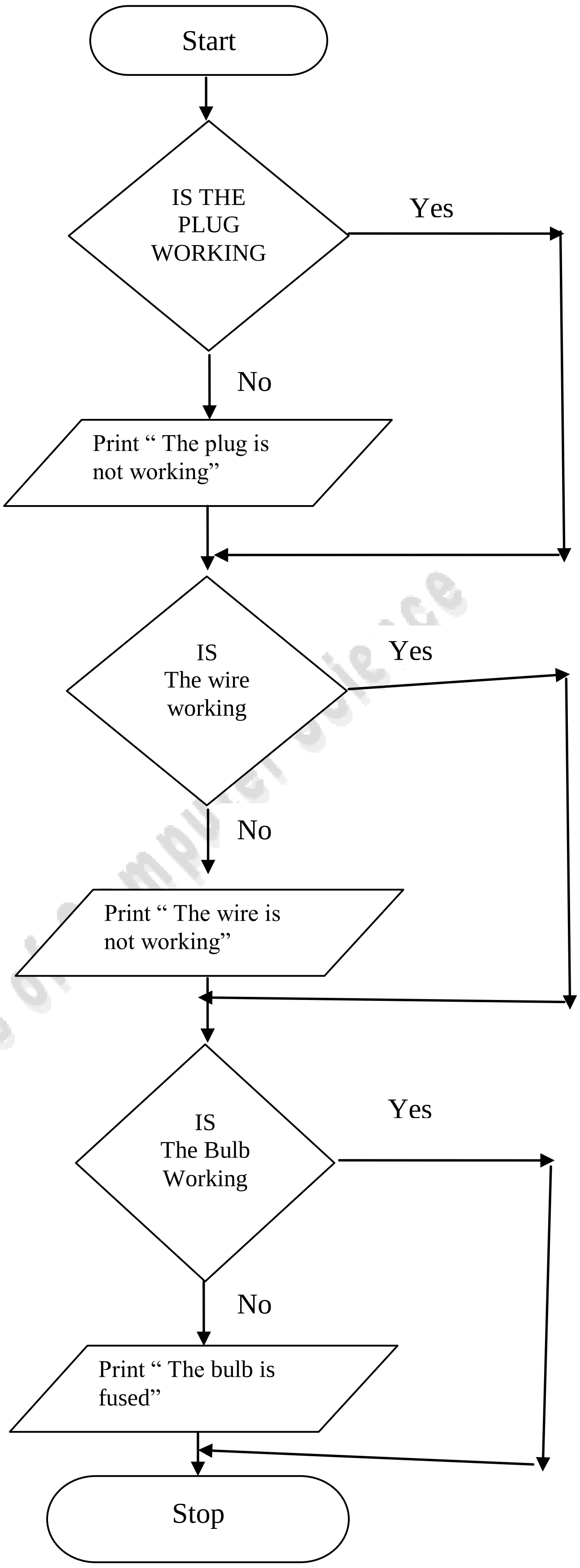
Ans. Look Q4 (Symbols used in a flowchart)

Q 9.11 Convert the above algorithm of question 9.08 into a flowchart.



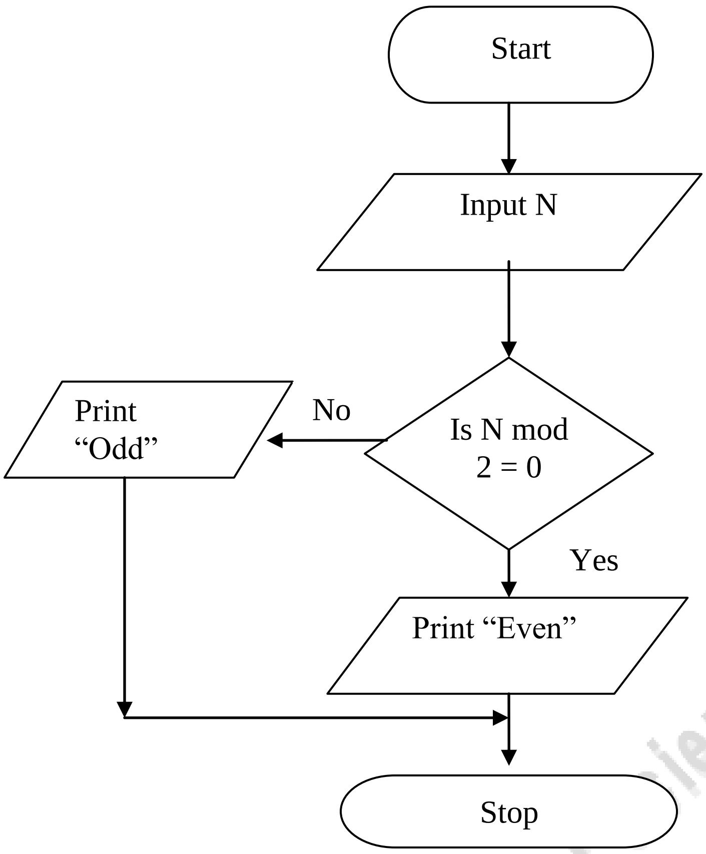
Q 9.12 **Your table lamp is not in working order. Develop a flowchart for locating the problem in it. The trouble may be either in the plug or in the connecting wire or the bulb itself may be fused.**

Ans.


Start
IS THE PLUG WORKING
Yes
No
Print " The plug is not working"
IS The wire working
Yes
No
Print " The wire is not working"
IS The Bulb Working
Yes
No
Print " The bulb is fused"
Stop

Q 9.13 Develop a flowchart for identifying and printing an odd number.

Ans.



Q 9.15 What is meant by a computer program?

Ans. Computer Program:

"A computer program is a set of instructions that tells a computer what to do."

Q 9.16 Mention the advantages of programming in BASIC.

Ans. Advantages of Programming in BASIC:

1. It is a user-friendly language.
2. It is very close to English language.
3. Its syntax is very simple.
4. It can be used for scientific and business applications.
5. It is the best tool for learning the basics of programming.

Q 9.17 What is debugging? Why is t necessary to test or debug a program?

Ans. There is always a chance of many errors in the program after coding it. So, it is necessary to test the program. In this step, the program is executed and tested with many inputs. In case of errors, an error report is made.

Afterwards, these errors are corrected. This is called debugging. It is necessary to debug a program to remove the errors from it.

Q 9.18 Why is it necessary to document a program?

Ans. Importance of Documentation: It is necessary to document a program because documentation best describes what the program should do. Documentation is helpful for the user. Moreover, whenever the program is needed to be improved or rewritten, documentation can be very helpful. It helps a programmer to modify the program.

Chapter No. 10: Data Types Assignment and Input / Output Statements

Q1. What are different features of BASIC programming language?

Ans. **BASIC:** BASIC is a high level language. It was developed in 1964. It quickly became one of the most popular language for general purpose use and scientific applications.

Key Features of BASIC

BASIC has the following features common in all versions:-

1. It is an easy, user-friendly language.
2. It is a general purpose language and can be used for business and scientific applications.
3. It has simple syntax rules and is easy to test and debug
4. It has English-like structure.
5. It is best for students.

Q2. What is Character Set in BASIC language?

Ans. **Character Set in BASIC Language:-**

The character set in BASIC consists of alphabetic characters of English, numerical characters and special characters. It also included numeric, functional, relational and logical operators.

- Alphabetic characters in BASIC are English capital and small letters. (i-e. A – Z, a-z)
- Numeric characters are from 0 to 9
- Special characters include $, #, %, @, /, blank etc.
- Arithmetical operators include +, -, /, * etc.
- Relational operators include =, <, >, <>, <=, >= etc.

Q3. What are reserved words?

Ans. **Reserved Words:**

Certain words have special meaning in BASIC. They are called reserved words. There are about 160 reserved words in BASIC.

Q4. What are different types of entries in BASIC? What is the difference between commands and statements?

Ans. **<u>Types of Entries in BASIC:-</u>**

There are three different types of entries in BASIC. These are:-

- Commands
- Program Statements and
- Data

Commands are executable instructions such as SAVE, LOAD, LIST etc. They are generally executable in 'direct mode'.

Program Statements include REM, LET, PRINT, END etc. They are used in programs in 'indirect mode'. A statement has two parts; a line number and one or more BASIC instruction.

Data in BASIC consist of two types; numeric data and string data. In other words, there are two data types in BASIC.

- **Numeric Data (type)** include numbers e.g. 12, +2.67, -23.4, etc.
- **String Data (type)** include words or text. A string consists of characters, special character or numeric characters or their combination. String data is enclosed in quotation marks.

<u>Difference between Commands and Statements:-</u>

The difference between commands and statements is that statements are generally executable in 'in-direct mode' while commands are generally executable in 'direct mode'**.** Commands generally perform program operations.

Q5. What are constants? What are their types? Give examples.

Ans. **<u>Constants:</u>**

"A constant is something which does not change its value during the execution of the program."

Types of Constants:

There are two types of constants:-

a) **Numeric Constants**: Numeric constants in BASIC are positive or negative numbers.
Examples:- 23, -4, 36.56 etc are numeric constants.

b) **String Constants:** A string constants is a set of (maximum 255) characters enclosed in quotation marks.
Examples:- "I love Pakistan", "F-16", "23rd March" etc.

Q6. What are variables? Also explain their types. What are rules for naming variables?

Ans. **Variables:**

"Variables are names used to represent certain quantities. Values of variables can change during the execution of the program."

Examples:- A=23, name$= " Abdul Qadeer", x=2.9 etc. Here A, name$ and x are variables.

Types of Variables: There are two types of variables:-

a) **Numeric Variables:** "Numeric variables are names given to numeric values. Their values change during the execution of the program. A numeric variable can have any numerical value. A numeric variable name must begin with an alphabet. Special characters and numbers are not allowed as variable names in BASIC."

b) **String Variables:** "String variables are names given to string values. They can store numbers, characters and numbers. The name of a string variable is followed by a dollar sign($). Strings are enclosed in double quotation marks".
e.g. 10 LET NAME$="Amir"

Rules for Naming Variable Names:

1. A variable name cannot begin with a digit. e.g. 5num is illegal and num5 is a legal variable name.
2. Blank spaces are not allowed in variable names. e.g. num 1 is illegal and num1 is a legal variable name.
3. Special character are not allowed in variable names. e.g. a$b is an illegal variable name. String variables are followed by a dollar sign. e.g. name$.
4. Reserved Words cannot be used as variable names. e.g. print, cls, etc. cannot be used as variable names.

Q7. What are operators? Explain different types of operators.

Ans. **Operators:** "Operators perform certain operations on values or operands."

Types of Operators: In BASIC, there are four different types of operators.

1. Arithmetic Operators: Arithmetic operators perform numerical operations. They are found in numeric expressions.

		Example in BASIC	
Operation	**Operator**	**Arithmetic**	**BASIC**
Addition	+	5+8	5+8
Subtraction	-	8-3	8-3
Multiplication	*	4x7	4*7
Division	/	14 / 2	14/2
Exponentiation	^	9^2	9^2
Negation	-	-6	-6

Hierarchy of Arithmetic operators:
First Step: Perform all operations inside parenthesis.
Second Step: Evaluate all exponentiation from right to left.
Third Step: Perform all multiplications and divisions from left to right.
Fourth Step: Perform all additions and subtractions from left to right.

2. Relational Operators: Relational operators are used to relate two values. They are found in comparison. The answer of expressions containing relational operators is either TRUE or FALSE.

OPERATOR	RELATION	EXAMPLE
=	Equality	IF a=15 THEN STOP
<>	Inequality	IF a<>15 THEN STOP
<	Less than	IF a<15 THEN STOP
>	Greater than	IF a>15 THEN STOP
<=	Less than or Equal to	IF a<=15 THEN STOP
>=	Greater than or Equal to	IF a>=15 THEN STOP

3. Logical Operators: Logical operators perform logical operations on given set of values. They include AND, OR, NOT, NAND, NOR, XOR etc. The result of logical operators is either true or false.
4. Functional Operators: Functional operators include FIX, SQR, LOG, SIN, COS etc. There is a wide range of functional operators in BASIC.

Q8. What are different modes of operation in BASIC? Explain.
OR
What is the difference between 'direct mode' and 'indirect mode'?

Ans. **Modes of Operation in BASIC:** In GW-BASIC, we can write instruction in two ways. These are called modes of operations. There are two modes of operation in BASIC. They are as under:

a) Direct Mode: In 'direct-mode', program statements are written without line numbers. Each statement is executed after we press <RETURN> or <ENTER> key after writing the command.
Program maintenance operations like opening the program, saving the program, listing the program statements etc are performed in 'direct-mode'.

b) Indirect Mode or Program Mode: BASIC programs are written in indirect or program mode. Each line in this mode begins with a line number.

Q9. Explain the following statements:
a) PRINT statement b) INPUT statement
c) Assignment statement / LET statement
d) READ-DATA statement
e) REM statement

Ans. a) **The PRINT Statement:** PRINT statement displays results of processing, values of variables of text on the monitor. Strings in the PRINT statement are written within quotation marks. Variables and arithmetic expression are written outside the quotation marks.

Syntax:
Line# PRINT <VARIABLES>
Line# PRINT <STRING CONSTANT/ TEXT>
Line# PRINT <ARITHMETIC EXPRESSION>
Line# PRINT <VARIABLE / STRINGS / ARITHMETIC EXPRESSIONS

Example:
```
10 LET A= 10, B=20
15 PRINT A
20 PRINT “Hello Pakistan”
30 PRINT A+B+C
40 PRINT “Square of A is”, A*A
```
Use of Comma and Semicolon in PRINT statement:
Comma in the print statement divides the screen into five zones. If all zones of first line are filled then next data goes to the first zone of the next line.
Semicolon in the print statement does not add any additional space in the strings.
Example:
Consider the following program
Program:
```
10 PRINT “PAKISTAN” ; “Zindabad”
```
Output:
```
PAKISTANZINDABAD
```

Consider another program with a comma instead of a semicolon.

Program:

```
10 PRINT “PAKISTAN” , “ZINDABAD”
```

Output:

```
PAKISTAN          ZINDABAD
```

b) The ‘INPUT’ statement:

The INPUT statement stores user’s entered values in variables after the execution of the program. A user executes the program and enters values, the INPUT statement stores those values in specified variables.

Syntax:

Line# INPUT <TEXT MESSAGE> ; <VARIABLE>

Example:

```
10 CLS
20 INPUT “Enter a number”; A
30 PRINT A*A
```

c) Assignment Statement / LET statement:

Assignment statement stores value in variable during the program writing process. Let statement is used to store values in variables at coding time.

“It assigns given vale to specified variable before program execution”

Syntax:

Line# LET <VARIABLE>=<VARIABLE or EXPRESSION>

Example:

```
10 LET A=20
20 LET B=30
30 LET C=A+B
```

d) Read-data Statement:

READ-DATA statement is another method to store value(s) in variable(s). It stores the values in variables when program is written. This statement is spread over two lines. One line contains READ statement and a list of variables. The other line contains DATA statement and values of variables.

Syntax:

Line# READ (Variable1), (Variable2), ……

Line# DATA (value 1), (value 2), …..

‘Value 1’ will be stores in ‘variable 1’, ‘value 2’ in ‘variable 2’ and so on.

Example:

```
10 CLS
20 READ A, B, C
30 DATA 10, 20, 30
40 PRINT A, B, C
50 END
```

Output:

```
10          20          30
```

e) REMark Statement:

The remark statement documents the program. It explains different lines. The text written after remark is not compiled or executed.

Syntax: REM <ANY TEXT>

Example:

```
10 CLS                REM This statement clears the screen.
```

f) The Tab Function:

This function is used in PRINT statement. It inserts specified number of spaces in the strings or values of variables.

Example:

```
10 PRINT “PAKISTAN” TAB(5) “IS” TAB(5) “MY COUNTRY”
```

Output:

```
PAKISTAN     IS     MY COUNTRY
```

Q10. Explain the following commands:

Ans.

1. **AUTO COMMAND:** It generates an automatic line number each time <ENTER> key is pressed. The default increment value for each line to next line is 10.
   Example:

   ```
   AUTO
   10 CLS
   20
   30
   ```

2. **CLEAR COMMAND:** It clears value assigned to numeric and string variables. It sets all numeric variables to 0 and all string variables to NULL.
   Example:

3. **DELETE COMMAND:** It deletes specified program lines or line range.
   Examples:

Delete 50	It will delete line no. 50.
Delete 30-80	It will delete lines from 30 to 80.
Delete -50	It will delete all lines upto line number 50.
Delete 70-	It will delete all lines from line number 70 onwards.

4. **EDIT COMMAND:** It edits a specified line.
   Syntax: EDIT <LINE#>
   Example: EDIT 10

5. **FILES COMMAND:** It is used to display all files from specified location/drive.
   Syntax: FILES (“file-specs”)

5. **KILL COMMAND:** It deletes a specified file from the disk.
   Syntax: KILL <FILE-NAME>
   Example: KILL PROGRAM.BAS

6. **LIST COMMAND:** It is used to display the program or part of the program currently in memory.
   Examples:

LIST:	To list all line of the program currently in memory.
LIST 20-40:	To list lines from 20 to 40.
LIST 40-	To list all lines from 40 onwards.
LIST -60:	To list all lines from beginning to line number 60.

7. **LOAD COMMAND:** It loads a file from the disk into computer’s memory.
   Syntax: LOAD “DRIVE-NAME\FILE-NAME”
   Example: LOAD file1.bas

8. **NEW COMMAND:** It deletes the old program from memory and clears all variables.
   Syntax: NEW
   Example: NEW

9. **RENUM COMMAND:** It is used to renumber program lines.
   Syntax:
   RENUM <NEW-FIRST-LINE#>,<OLD-LINE-NUMBER>, <INCREMENT>
   Example:
   RENUM 10, 5, 10

10. RUN COMMAND: It executes the program or specified line of the program currently in memory.

Syntax: RUN <LINE#> or RUN
Example: RUN ----It will run the whole program.
RUN 20 ----It will run line number 20.

11. SAVE COMMAND: It saves the program on a specified drive.

Syntax: SAVE "<DIRVE : FILE-NAME>"
Example: SAVE "A:Test"

12. SYSTEM COMMAND: It is used to exit from BASIC.

Syntax: SYSTEM
Example: SYSTEM

Q-10.05 Write a short note on BASIC and describe some of its key features.

Ans. Look Q1

Q-10.06 What types of entries are made in BASIC? What is difference between commands and statements?

Ans.

Q-10.07 Describe briefly the following commands:-
Auto, delete, list, renum and system.

Ans. Look Q10

Q-10.08 How many types of variable are used in BASIC? Give examples of invalid variables name with explanations.

Ans. Look Q6

Q-10.09 Describe various types of constants used in BASIC with examples.

Ans. Look Q5

Q-10.10 Pick out numeric and string from the following data items:-
"LAHORE";27:"60W"; 1.008, "16" "20 KM" "23RD MARCH" "IX;A-1.

Ans.

String Items	Numeric Items
"Lahore"	**27**
"60W"	**IX**
"16"	**A-1**
"20 KM"	
"23rd March"	

Q-10.11 Explain various types of operators used in BASIC with examples?

Ans. Look Q7.

Q-10.12 How does computer know whether an instruction you typed in a program?

Ans. In program mode, each instruction is written after a line number. A computer knows that an instruction is in program mode by the line number.

Q-10.13 What commands clear the computer memory?

Ans. NEW command clears computer's memory.

Q-10.14 What will be the output when you print "37 C =" 37*9/5 + 32?

Ans. 98.6

Q-10.15 Write a note on assignments statements with examples?

Ans. Look Q9(part c)

Q-10.16 What is the difference between STOP and END statements? Explain it with example?

Ans. The STOP statement stops the loop while the END statement ends the program.

Q-10.17 Write a program that prints 3 numbers, their sum and average using LET statements?

Ans. **Program:**

```
10      CLS
20      LET A=10
30      LET B=14
40      LET C=6
50      LET SUM=A+B+C
60      LET AVG=SUM/3
70      PRINT “THREE NUMBERS ARE”; A, B, C
80      PRINT “SUM IS:”;SUM
90      PRINT “AVERAGE IS:”;AVG
100     END
```

Q-10.18 Write a program that prints 3 numbers, their sum and average using INPUT statement?

Ans. **Program:**

```
10      CLS
20      INPUT “ENTER FIRST NUMBER”; A
30      INPUT “ENTER SECOND NUMBER”; B
40      INPUT “ENTER THRID NUMBER”; C
50      LET SUM=A+B+C
60      LET AVG=SUM/3
70      PRINT “THREE NUMBERS ARE”; A, B, C
80      PRINT “SUM IS:”;SUM
90      PRINT “AVERAGE IS:”;AVG
100     END
```

Q-10.19 Write a program for the square and cube of any number using INPUT statements?

Ans. **Program:**

```
10      CLS
20      INPUT “ENTER FIRST NUMBER”; A
30      LET SQ=A^2
40      LET CUBE=A^3
50      PRINT “THE SQUARE IS”; SQ
60      PRINT “THE CUBE IS”; CUBE
70      END
```

Q-10.20 Tell whether the following are valid BASIC program. If not give a possible correction and its output if any.

Ans. On the book

Q-10.21 What will be the output of the following program?

Ans. On the book

Q-10.22 Write a BASIC program to calculate the sum, product and average of 4 numbers using INPUT statement and READ-DATA statement.

Ans. **Program Using INPUT Statement:**

```
10      CLS
20      INPUT “ENTER FIRST NUMBER”; A
30      INPUT “ENTER SECOND NUMBER”; B
40      INPUT “ENTER THRID NUMBER”; C
50      INPUT “ENTER FOURTH NUMBER”; D
60      LET SUM=A+B+C+D
70      LET PROD=A*B*C*D
80      LET AVERAGE= SUM/4
70      PRINT “THE SUM IS:”; SUM
80      PRINT “THE PRODUCT IS:”; PROD
90      PRINT “AVERAGE IS:”;AVG
100     END
```

Program Using READ-DATA Statement:

```
10      CLS
20      READ A, B, C, D
30      DATA 10, 23, 12, 45
40      LET SUM=A+B+C+D
50      LET PROD=A*B*C*D
60      LET AVERAGE= SUM/4
70      PRINT “THE SUM IS:”; SUM
80      PRINT “THE PRODUCT IS:”; PROD
90      PRINT “AVERAGE IS:”;AVG
100     END
```

Chapter No. 11: Control Statements

Q1. Explain the use of GOTO statement?

Ans. The 'GOTO' Statement:(Unconditional Transfer of Control)

All computer programs are lists of statements in a linear order. So, the statements that appear first in the program are executed first. But, in BASIC, user can change the order of execution. The GOTO statement is used to change the order of execution in which the program is run.

GOTO statement is a transfer of control statement. It transfers the control to a specified line number.

Syntax: LINE# GOTO<line#>

Example: Program:

```
10      CLS
20      PRINT "PAKISTAN"
30      GOTO 20
40      PRINT "ZINDABAD"
50      END
```

Output:

```
PAKISTAN
PAKISTAN
PAKISTAN
           .
```

Q2. Explain the conditional transfer of control. Also explain the statements causing the conditional transfer of control.

Ans. Conditional Transfer of Control:

To transfer program control based on the result of an expression or condition is called 'conditional transfer of control'.

In BASIC following statements are used to transfer control conditionally:-

IF-THEN Statement:

IF-THEN statement is used to make decisions based on some condition(s). It consists of a condition and a block of statements. If the condition is true the block of statements is executed otherwise the control is transferred to the next line after the IF-THEN statement.

Syntax: IF <condition> THEN <Statement>

Example: IF a=10 THEN PRINT " The value is 10"

IF-THEN-ELSE Statement:

In IF-THEN-ELSE statement, the control is transferred to some part of the program if the condition is 'true' and if the condition is false the control is transferred to some other specified part of the program.

Syntax: IF <condition> THEN <Statement-1> ELSE <Statement-2>

Example: IF A=10 THEN PRINT "TRUE" ELSE PRINT "FALSE"

ON–GOTO Statement:

The ON-GOTO statement jumps the control based on a condition to a specified line number. It is used for multiple branching. It needs a variable. When the variable's value is 1 the control is transferred to first specified location, if value is 2 the control is transferred to second specified location and so on.

Syntax: Line# ON <variable> GOTO l-line#, 2-line# …..

Example: 30 ON A GOTO 20, 30, 40

Programming Example:

```
10 CLS
20 INPUT "ENTER A NUMBER FROM 1 TO 4"; A
30 ON A GOTO 40,50,60,70
40 PRINT "ONE"
50 PRINT "TWO"
60 PRINT "THREE"
70 PRINT "FOUR"
80 END
```

Q3. Explain the use of ON-ERROR-GOTO statement.

Ans. <u>ON-ERROR-GOTO Statement:</u>

ON-ERROR-GOTO statement displays error message. Whenever an error occurs, it transfers the control to a specified line number. The ON-ERROR-OFF no longer suppresses error message i.e. the computer handles errors in normal way.

Example:

```
10      REM ** ERROR DETECTING PROGRAM**
20      ON ERROR GOTO 200
30      LET X=0
40      PRINT 27/X
50      PRINT: PRINT"IMPOSSIBLE"
60      END
200     PRINT "ERROR CODE"; ERR
210     PRINT "OCCURS AT LINE"; ERR
220     GOTO 50
```

Q4. Explain the use of FOR-NEXT loop.

Ans. <u>FOR-NEXT Loop:</u>

In some cases, a part of a program is repeated several times. It is necessary to use a loop to execute the part of program repeatedly. FOR-NEXT loop is a **counter loop**. It executes the statements with in FOR and NEXT statements specified number of times. The 0number of iterations in FOR-NEXT loop are known in advance.

<u>Syntax:</u>

```
FOR <VARIABLE>=<INITIAL VALUE>   to  <FINAL VALUE> STEP <VALUE>
   ……………………............Body of the loop (statements)…………………………….
   ……………………............Body of the loop (statements)…………………………….
NEXT <VARIABLE>
```

<u>Example:</u>

```
FOR I=1 to 10 STEP 2
  PRINT "PAKISTAN"
  PRINT "ZINDABAD"
NEXT I
```

Q5. What is meant by WHILE-WEND loop?

Ans. While-wend Loop:

While-wend loop is a controlled loop. It executes a given set of statements until a given condition becomes FALSE.

Syntax:

```
        WHILE <condition>
                Statement1
                Statement2
                …………..
                …………..
        WEND
```

Example:

```
        WHILE n<10
          PRINT "PAKISTAN"
          INPUT n
        WEND
```

Q6. What is meant by the term 'nested loops'?

Ans. <u>Nested Loops</u>:

FOR-NEXT or WHILE-WEND loop may be placed inside another loop. So the loop inside another loop is called 'Nested Loop'.

<u>Rules for Nesting a Loop</u>:

1. Each loop should have different variable name.
2. Inner loop should be ended before outer loop.
3. The outer loop should be ended after inner loop.

Q7. What is the difference between conditional and unconditional transfer of control?

Ans. Unconditional Transfer of Control:

In BASIC user can transfer control from one line to another specified line anywhere in the program. This is called transfer of control. If user transfer control without specifying any condition, it is called unconditional transfer of control.
The GOTO statement is an unconditional transfer of control statement. It transfers the control to a specified line number.

<u>Conditional Transfer of Control:</u>
To transfer program control based on the result of an expression or condition is called 'conditional transfer of control'.
In BASIC, IF-THEN, IF-THEN-ELSE and ON-GOTO statements cause conditional transfer of control.

Q8. What is the difference between counter and controlled loops?

Ans. Counter Loops: A counter loop is a loop in which the number of iterations is known in advance. Programmer needs to mention the number of times the loop will execute.
Example: In BASIC FOR-NEXT is a counter loop.

Controlled Loops: A controlled loop is a loop which executes a set of statements until a specific / mentioned condition becomes false.
Example: In BASIC, WHILE-WEND is an example of controlled loop.

Chapter No. 12 Arrays

12.06 What is an array? Illustrate the difference with examples between one dimensional and two dimensional arrays.

Ans. **Array:**

An array is a collection of like variables arranged in a sequence identified by their subscripts. It can store multiple values of the same data type.

SUBSCRIPTS:

The subscripts must be any positive integral value e.g. A(2),A(3) B$(6)etc. A subscript cannot be a decimal number or a fraction.

HOW TO DECLARE AN ARRAY:

In GW-BASIC, 'DIM' statement is used to declare an array. User gives the name of the array and its size within parenthesis.

SYNTAX OF DIM STATEMENT:

Line # ARRAY-NAME (size or no .of locations)

EXAMPLE OF DIM STATEMENT:

10 DIM A (5) ----This statement will declare an array with name A and size 5.

ONE-DIMENSIONAL AND TWO-DIMENSIONAL ARRAYS:

S#	ONE-DIMENSIONAL ARRAYS	TWO-DIMENSIONAL ARRAYS
1.	One-dimensional arrays are like list where there is only one column.	Two-dimensional arrays are like tables where there are multiple rows and columns.
2.	This type of arrays requires only one loop to fill values or to read or process values.	This type of arrays requires two loops, one for rows and the other for columns.
3.	Only one subscript is required to represent an element in one-a dimensional array.	More than one subscript is required to represent an element in two- dimensional array.

12.07 What is the difference between simple and subscripted variables? Explain with examples.

Ans. A SIMPLE VARIABLE

A simple variable has no other information to be processed or addressed except its name. A simple variable can store only one value at a time. eg A,B$ etc.

A SUBSCRIPTED VARIABLE

A subscripted variable needs numeric subscript to be mentioned after its name with in parenthesis .eg.A (3),B(S),G(3,4)etc.

12.08 What is the use of DIM statement? When it is needed?

ANS THE USES OF DIM STATEMENT

DIM is a short name for dimension. The DIM statement is used to declare an array of numeric or string data type in the computer memory. User gives the name of the array and its size with in parenthesis.

SYNTAX OF DIM STATEMENT;

Line # ARRAY_NAME (size or no. of locations)

EXAMPLE OF DIM STATEMENT;

11 DIM A(5) -----This statement will declare an array with name A and size 5.

12.09 What is the difference of DIM statement in one dimensional and two dimensional arrays?

Ans. DIM STATEMENT IN ONE DIMENSIONAL AND TWO DIMENSIONAL ARRAYS:

DIM Statement in On Dimensional Arrays:

How to declare a one Dimensional Arrays:
In GW-BASIC , 'DIM' statement is used to declare a one dimensional array. User gives the name if the array and its size within parenthesis.

Syntax of SIM Statement for One Dimensional Array:
Line# ARRAY-NAME (Size or no. of locations)

Example of DIM statement:
10 DIM A (5) – This statement will declare a one dimensional array with name A and size 5.

DIM Statement in Two Dimensional Array:

How to Declare a Two Dimensional Array: In GW-BASIC 'DIM' statement is used to declare also a two dimensional array. User gives the name of the array and its number of rows and columns within the parenthesis separated by comma (,).

Syntax of DIM Statement for Two Dimensional Arrays:
Line# ARRAY-NAME(NO. OF ROWS, NO. OF COLUMNS)

Example of DIM Statement for Two Dimensional Array:
10 DIM A (5, 4) ---This statement will declare a two dimensional array with Name A with 5 rows and 4 columns.

12.10 Write a program that fills an array with 16 random integers each between 25 and 60 then prints the array.

Ans. PROGRAM TO FILL 16 RANDOM INTEGERS IN AN ARRAY AND THEN PRINTING THE ARRAYS:

```
10      CLS
20      DIM NUM(16)
30      FOR I= 1 TO 16
40      READ NUM(I)
50      NEXT I
60      DATA 29, 26, 67, 44, 31, 34, 59, 41, 49, 33, 38, 55, 44, 29, 51, 53
70      FOR I=1 TO 16
80      READ NUM(I)
90      PRINT NUM(I)
100     NEXT I
110     END
```

12.11 Write a program that fills an array P with 20 random integers and then prints the product of the elements of the array.

Ans. PROGRAM TO FILL SIXTEEN RANDOM INTEGERS IN AN ARRAY AND THEN PRINTING THE ARRAY:

```
CLS
20      DIM P(20)
30      LET PRODUCT =1
40      FOR I=1 TO 20
50      READ NUM(I)
60      NEXT I
70      DATA 29,26,67,44,31,79,23,78,90,12,56,76,23,55, 12,45,56,12,34
80      FOR I=1 TO 20
90      LET PRODUCT=PRODUCT * I
100     NEXT I
110     PRINT "PRODUCT IS:"; PROCUCT
120     END
```

Chapter No. 13 Sub-Program and File Handling

Q1. What is a structured program?

Ans. <u>Structured Program</u>

A structured program is written in modules or blocks. It has only one entry and one exit point. A structured program is easy to follow and understand.

Q2. What is a subprogram?

Ans. <u>Subprogram:</u>

A subprogram is a group of statements that perform a single operation on execution. Subprogram is written once and can be called (used) at several places in the program.

Advantages of a subprogram:

1. A subprogram is easy to design, document and debug.
2. A subprogram is written once and can be used at different places.

Q3. What are types of subprograms? Explain.

Ans. <u>Types of Subprograms:</u>

There are two types of subprograms:

1. Function Subprogram
2. Subroutine Subprogram

1. Function Subprogram:

A function (or function subprogram) is a routine that takes a value (string or value) as an input based on the input value.

2. Subroutine Subprogram:

A subroutine subprogram in BASIC is a series of statements that is written once in a program and can be called many time in the program.

Q4. What are different types of functions? Explain.

Ans. <u>Types of Functions:</u>

There are two types of functions:

1. System Defined Functions
2. User Defined Functions

1. System Defined Functions / Built-in Functions / Library Functions:

BASIC includes a wide range of library / built-in functions / library functions. They are already defined in BASIC. They are not defined again but used.

Chapter No. 14 Graphics in BASIC

14.05) What is meant by pixel and resolution?

Ans. **Pixel:** All images on a monitor screen are composed of very fine and small dots or picture elements. These fine dots are called ‘pixels’ or picture elements. The size, color and number of pixels varies from monitor to monitor.

Resolution: The number of pixels per inch on a monitor screen if the resolution of the monitor. Quality graphics can be produced if the resolution is high.

14.06) Describe the different modes to display graphics of the screen.

Ans. In GW-BASIC, ‘screen’ function is used to set the graphics mode. It can set the screen to any of the following three modes:

a) **Text Mode:** In Text mode, only characters can be displayed on the screen. In this mode the screen is divided in 40 columns and 25 rows (40 x 25) or 80 columns and 25 rows (80 x 25).

b) **Medium Resolution Graphics Mode:** In medium resolution graphics mode, the screen is divided into a matrix consisting of 320 columns and 200 rows with 4 colors. Columns are numbered from 0 to 199.

c) High Resolution Graphics Mode: In high resolution graphics mode, the screen is divided onto 640 columns and 200 rows. Columns are numbered from 0 to 639 and rows are numbered from 0 to 199. The first location can be referred with (0,0) and the last location by (639,199) which is the lower right corner of screen.

Extra Question 1) Describe the use of ‘screen’ function with example.

Ans. **Screen Function:** Normally the monitor screen is in the text mode. In GW-BASIC, we cannot draw good graphics in text mode. For drawing good graphics (circles, lines, points etc.), the screen must be set to ‘medium’ or ‘high’ resolution mode. This is done with the help of ‘screen’ function.

Syntax of Screen Function: Line# SCREEN mode

‘Mode’ can be set in any of these three ways:

- Text Mode - SCREEN 0
- Medium Resolution Graphics Mode - SCREEN 1
- High Resolution Graphics Mode - SCREEN 2

Example of Screen Function: 10 SCREEN2

In above example, the screen will be set to ‘medium resolution graphics’ mode.

Extra Question

2) Describe the use of ‘color’ statement with example.

Ans. **Color Statement:** The ‘color’ statement is used to set the color in GW-BASIC. Its function also depends of the screen mode. It does not work in ‘high’ resolution graphics mode’.

Syntax of Color Statement in Text Mode:

Line# COLOR [foreground], [background]

The foreground color number must be an integer from 0 to 31. Color from 16 to 31 are blinking versions of the colors from 0 to 15. The background colors are from 0 to 15.

Example of Color Statement in Text Mode:

```
10 CLS
20 SCREEN 0
30 COLOR 4,2
40 PRINT “PAKISTAN”
```

Syntax of Color Statement in Medium Resolution Graphics Mode:

Line# COLOR [background], [palette]

Background Color and Palettes:

In this mode, there are 16 background colors from 0 to 15. There are two palettes (numbered from 0 to 1) available each consisting of 4 colors.

COLOR		**CODE**
Palette 0	Background	0
	Green	1
	Red	2
	Brown	3
Palette 1	Background	0
	Cyan	1
	Magenta	2
	White	3

Example of Color Statement in Medium Resolution Graphics Mode:

```
10 SCREEN 1
20 COLOR 7, 0
30 PSET (50, 50), 2
```

14.07 What do you mean by the background color and palettes?

Ans. …SEE ABOVE HEADING, ‘COLOR AND PALETTES’ FOR ANSWER…

14.08 What is the purpose and syntax of the PSET statement? Give examples.

Ans. Purpose of PSET statement:

This statement draws a point (dot) on screen specified coordinates (x, y).

Syntax of PSET statement:

Line# PSET (x, y), color

Example of PSET statement:

```
10 CLS
20 SCREEN 1, 0
30 COLOR 1, 0
40 PSET (40, 10), 2
```

This program draws a point at (40, 10) in brown color.

14.09 What is the purpose and syntax of the LINE statement? Give examples.

Ans. The LINE statement is used to draw a line between two points. The two point are specified in (x, y) coordinates.

Syntax of LINE statement:

Line# LINE (x, y) – (x, y), color

Example of LINE statement:

```
10 CLS
20 SCREEN 1, 0
30 COLOR 2, 0
40 LINE (80, 50) – (200,110), 2
50 END
```

14.12 What is the purpose and syntax of DRAW statement? Which commands in DRAW statement are used for diagonal movement?

Ans. Purpose of DRAW statement:

The 'DRAW' statement is used to draw lines and other shapes in GW-BASIC.

Syntax of DRAW statement:

Line# DRAW string

A string consists of single character command followed by a prefix that controls the size, direction etc. of the line. It is enclosed in the quotation marks.

Example:

30 DRAW "E30"

Drawing Commands:

The drawing commands are used with 'n' as the prefix to describe the number of points moved by the command.