10th Class		
Computer Science	Model Paper 3	Paper: II
Time: 1.45 Hours	(Subjective Type)	Marks: 40

(Part-I)

2. Write short answers to any FOUR (4) questions: (8)

(i) What are the Reserved Words?

Ans Every programming language has a list of words that are predefined. Each word has its specific meaning already known to the compiler. These words are known as **reserved words** or keywords.

(ii) Describe the purpose of a compiler.

Ans A **compiler** is a software that is responsible for conversion of a computer program written in some high level programming language to machine language code.

(iii) Differentiate between constants and variables.

Ans **Constants** are the values that cannot be changed by a program *e.g.*, 5, 75.7, 1500, etc.

A **variable** is actually a name given to a memory location, as the data is physically stored inside the computer's memory. The value of a variable can be changed in a program. It means that, in a program, if a variable contains value 5, then later we can give it another value that replaces the value 5.

(iv) Define format specifier.

Ans Format specifiers are used to specify format of data type during input and output operations. Format specifier is always preceded by a percentage (%) sign.

(v) What is the difference between *scanf* and *getch*?

Ans ***scanf*** is a built-in function in C language that takes input from user into the variables.

getch() function is used to read a character from user. The character entered by user does not get displayed on screen. This function is generally used to hold the execution of program because the program does not continue further until the user types a key.

(vi) What are logical operators? Describe with an example.

Ans Logical operators perform operations on Boolean expressions and produce a Boolean expression as a result.

As result of a relational operation is a Boolean expression, so logical operators can be performed to evaluate more than one relational expressions.

3. Write short answers to any FOUR (4) questions: (8)

(i) Define condition.

Ans A **condition** could be any valid expression including arithmetic expressions, relational expressions, logical expressions, or a combination of these.

(ii) Differentiate between sequential and selection statements.

Ans **Sequential** control is the default control structure in C language. According to the sequential control, all the statements are executed in the given sequence. On the other hand, the statements which help us to decide the statements which should be executed next, on the basis of conditions, are called **selection statements.**

(iii) Identify error in the following code:

```
if (a < 7 < b)
        printf ("7");
```

Ans **Error:** '7' will be printed as string.

(iv) Write down output of the following code:

```
char c1 = 'Y', c2 = 'N' ;
int n1 = 5, n2 = 9;
n1 = n1 + 1;
c1 = c2;
if (n1 = = n2 && c1 = = c2)
        printf ("%d = %d and %c = %c", n1, n2, c1, c1);
else
        if (n1 < n2 && c1 = = c2)
          printf ("%d < %d and %c = %c", n1, n2, c1, c2);
        else
          printf ("Better Luck Next Time!)"
```

Ans **Output:** 6 < 9 and N = N.

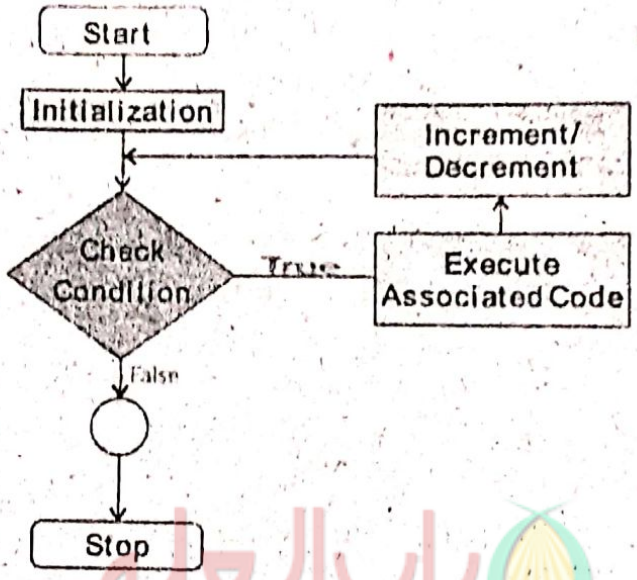
(v) What is array initialization.

Ans Assigning values to an array for the first time, is called **Array Initialization.** An array can be initialized at the time of its declaration, or later. Array initialization at the time of declaration can be done in the following manner:

data_type array_name [N] = {value1, value2, value3, ..., valueN};

(vi) Describe the structure of a *for* loop.

Ans In order to understand the *for* loop structure, let's look at the following flow chart:



4. Write short answers to any FOUR (4) questions: (8)

(i) Identify the error in the following code:

```
int a[ ] = {1, 2, 3, 4, 5};
for (int j = 0, j < 5 ; j++)
        printf ("%d ", a(j));
```

Ans **Error:** Square bracket is used instead of parenthesis in printf statement.

(ii) Write down output of the following code:

```
for (int i = 50; i <= 50; i++)
{
        for (j = i; j >= 48; j--)
            printf ("j = %d \ n" , j);
        printf ("i = %d\n", i);
}
```

Ans **Output:**

j = 50
j = 49
j = 48
i = 50

(iii) What are the Built-in functions.

Ans The functions which are available in C Standard Library are called **built-in functions.**

(iv) What do you know about the return keyword?

Ans The return keyword or return statement is used to return some value or simple pass the control to the calling function.

(v) Define the Functions Parameters.

Ans **Parameters** are variables of different data types, that are used to receive the values passed to the function as input.

(vi) Identify the errors in the following code segments.

```
int product (int n1, int n2)
        return n1 * n2;
```

Ans **Errors:** Function body curly braces are missing.

(Part-II)

NOTE: Attempt any TWO (2) questions.

Q.5. Define logical operators. Write its types in detail. (8)

Ans **Logical Operators:**

Logical operators perform operations on Boolean expressions and produce a Boolean expression as a result. As we know that result of a relational operation is a Boolean expression, so logical operators can be performed to evaluate more than one relational expressions. Following table shows the logical operators offered by C language:

Operator	Description
&&	Logical AND
||	Logical OR
!	Logical NOT

Table: Basic logical operators and their description.

AND operator (&&):

AND operator && takes two Boolean expressions as operands and produces the result true if both of its operands are true. It returns false if any of the operands is false. Following table shows the truth table for AND operator:

Expression	Result
False && False	False
False && True	False
True && False	False
True && True	True

Table: Truth table for AND operator.

OR operator (||):

OR operator accepts Boolean expression and returns true if at least one of the operands is true. Table shows the truth table for OR operator:

Expression	Result
False || False	False
False || True	True
True || False	True
True || True	True

Table: Truth table for OR operator.

NOT Operator (!):

NOT operator negates of reverses the value of Boolean expression. It makes it true, if it is false, and false if it is true. Following table presents the truth table for Not operator:

Expression	Result
!(True)	False
!(False)	True

Table: Truth table for NOT operator.

Example of Logical Operators:

Table illustrates the concept of logical operators with the help of examples.

Logical Expression	Explanation	Result
3 < 4 && 7 > 8	3 is less than 4 AND 7 is greater than 8?	False
3 = = 4 || 3 > 1	3 is equal to 4 OR 3 is greater than 1?	True

!(4 > 2 || 2 = = 2)	NOT (4 is greater than 2 OR 2 is equal to 2)?	False
6 < = 6 && ! (1 > 2)	6 is less than or equal to 6 AND NOT (1 is greater than 2)?	True
8 > 9 || !(1 < = 0)	8 is greater than 9 OR NOT (1 is less than or equal to 0)?	True

Table: Illustration of logical operators with examples.

Q.6. Write a program that counts multiples of a given number lying between two numbers. (8)

Ans

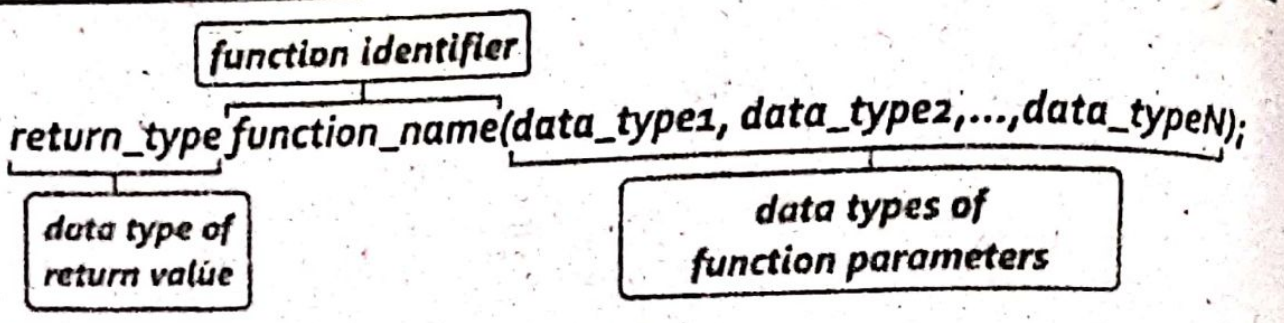
```
#include <stdio.h>
void main ()
{
        int n, lower, upper, count = 0;
        printf ("Enter the number: ");
        scanf ("%d", &n);
        printf ("Enter the lower and upper limit of multiples: \n");
        scanf ("%d%d", & lower, &upper);
        for (int i = lower; i < = upper ; i++)
          if (i % n = = 0)
                count++;
        printf ("Number of multiples of %d between %d and %d
        are %d", n, lower, upper, count);
}
```


Q.7. What is signature of a function? Describe it with the help of examples. (8)

Ans **Signature of a Function:**

A function is a block of statements that gets some inputs and provides some output. Inputs of a function are called parameters of the function, and output of the function is called its return value. A function can have multiple parameters, but it cannot return more than one values.

Function signature is used to define the inputs and output of a function. The general structure of a function signature is as follows:



Example Function Signatures:

Following table shows the descriptions of some functions and their signatures;

Function Description	Function Signature
A function that takes an integer as input and returns its square.	int square (int);
A function that takes length and width of a rectangle as input and returns the perimeter of the rectangle.	Float perimeter (float, float);
A function that takes three integers as input and returns the largest value among them.	int largest (int, int, int);
A function that takes radius of a circle as input and returns the area of circle.	float area (float);
A function that takes a character as input and returns 1, if the character is a vowel, otherwise returns 0.	int is Vowel (char);

Table: Some functions and their Signatures.