Permanent ID

20DCS103

Assignment : 01

20TDCS122 Temporary ID

1) List out advantages of Algorithm and Flowchart.

* Advantages of Algorithm :

→ Algorithm is a step-wise representation of a solution to a given problem, which makes it easy to understand.

→ An algorithm is not dependent on any programming language, so it is easy to understand for anyone even without programming knowledge.

→ An algorithm uses a definite procedure.

→ Every step in an Algorithm has its own logical sequence so it is easy to debug.

* Advantages of Flowchart :

→ Flowchart is an excellent way of communicating the logic of a program.

→ It is very easy and efficient to analyze problem using flowchart.

→ During program development cycle, the flowchart plays the role of a blueprint, which makes program development process easier.

→ After successful development of a program, it needs continuous timely maintenance during the course of its operation. The flowchart makes program or system maintenance easier.

→ It is easy to convert the flowchart into any programming language code.

20DCS103

2) Write an algorithm to find average of Even number between 1-50.

* Algorithm :-

Step: 1 Start

Step: 2 i=1, sum=0, count=0, average.
(taking variables)

Step: 3 if i%.2 == 0 then go to step 4,
else go to step 5.

Step: 4 sum = sum + i,
count = count + 1

Step: 5 i = i + 1

Step: 6 if i <= 50 then go to step 3,
else go to step 7.

Step: 7 print the value of sum.

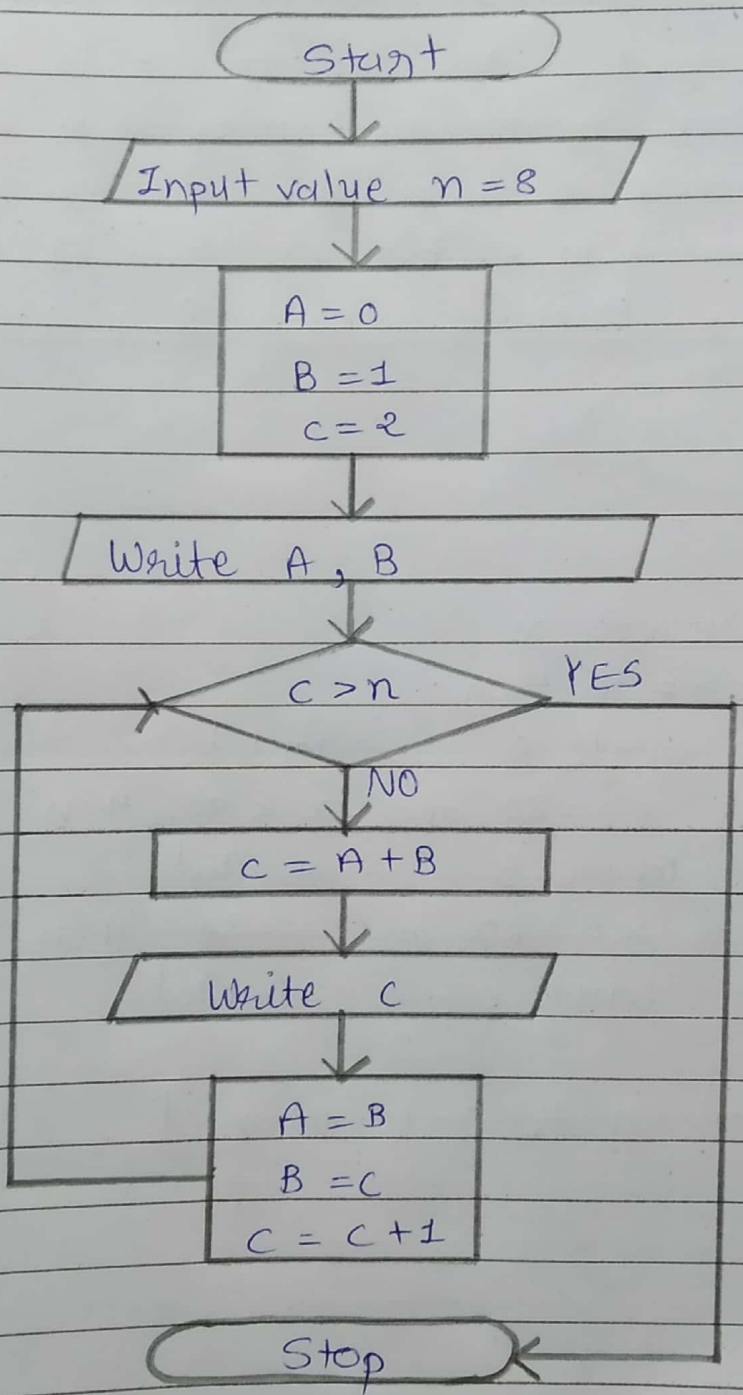
Step: 8 average = $\frac{sum}{count}$

Step: 9 print average value.

Step: 10 Stop

20DCS103

3) Design flowchart to create given series "0, 1, 1, 2, 3, 5, 8, 13". (Fibbonacci series).



20DCS103

4) Write an algorithm and draw a flowchart to read the marks of a student and classify them into different grades. If the marks secured are greater than or equal to 90, the student is awarded Grade A; if they are greater than or equal to 80 but less than 90, Grade B is awarded; if they are greater than or equal to 65 but less than 80, Grade C is awarded; otherwise Grade D is awarded.

* Algorithm :

Step : 1 Start

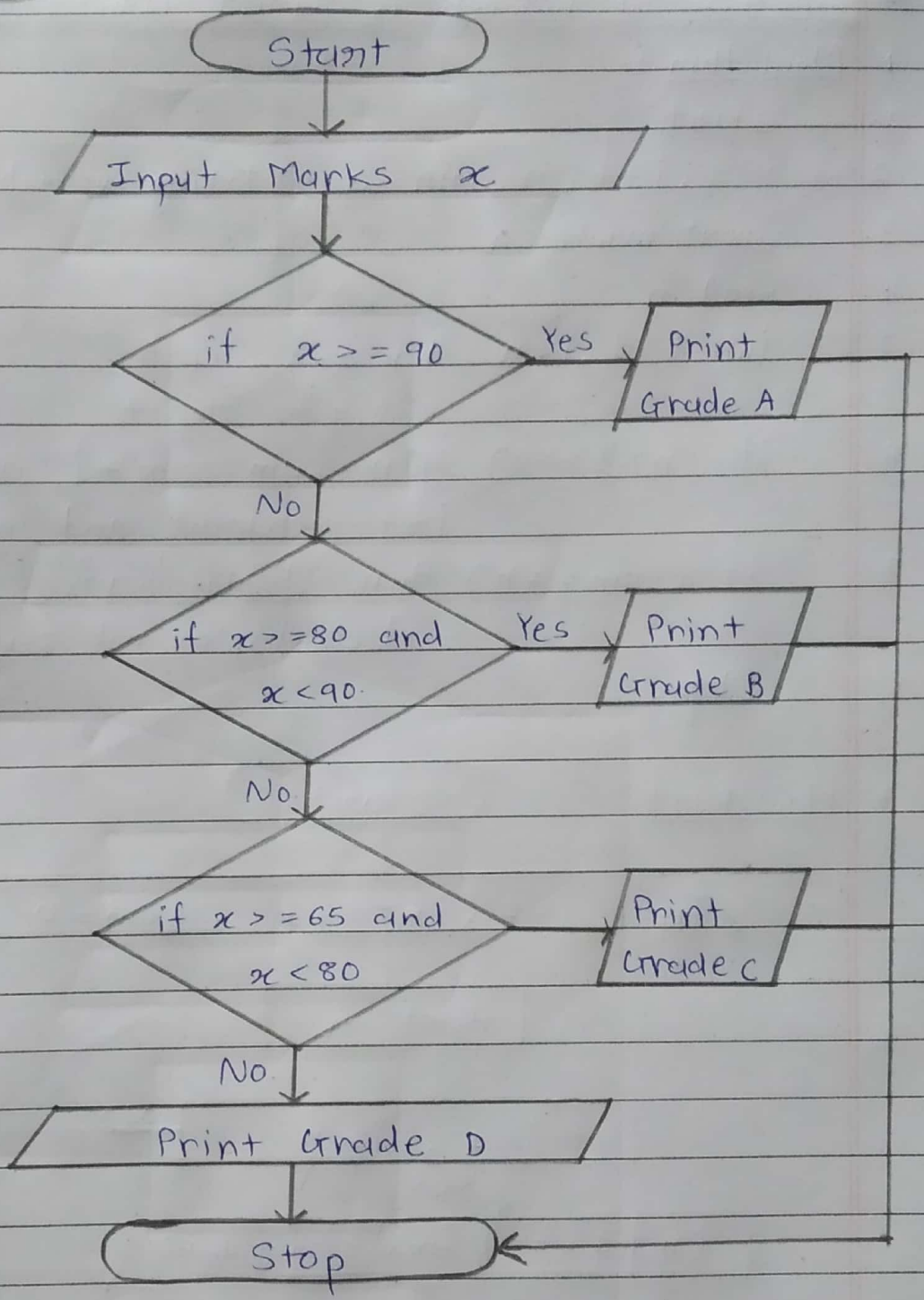
Step : 2 Input marks x

Step : 3 if x >= 90 then
print Grade A is awarded
else if x >= 80 and x < 90 then
print Grade B is awarded.
else if x >= 65 and x < 80 then
print Grade C is awarded.
else
print Grade D is awarded.

Step : 4 Stop.

20DCS103

* Flowchart:-



20DCS103

5.) Write an algorithm and flowchart to find whether a number is Perfect Number or not.

* Algorithm :-

Step : 1 Start

Step : 2 int i = 1, i++, n, sum = 0 (Taking variables)

Step : 3 Input numbe n

Step : 4 Read n

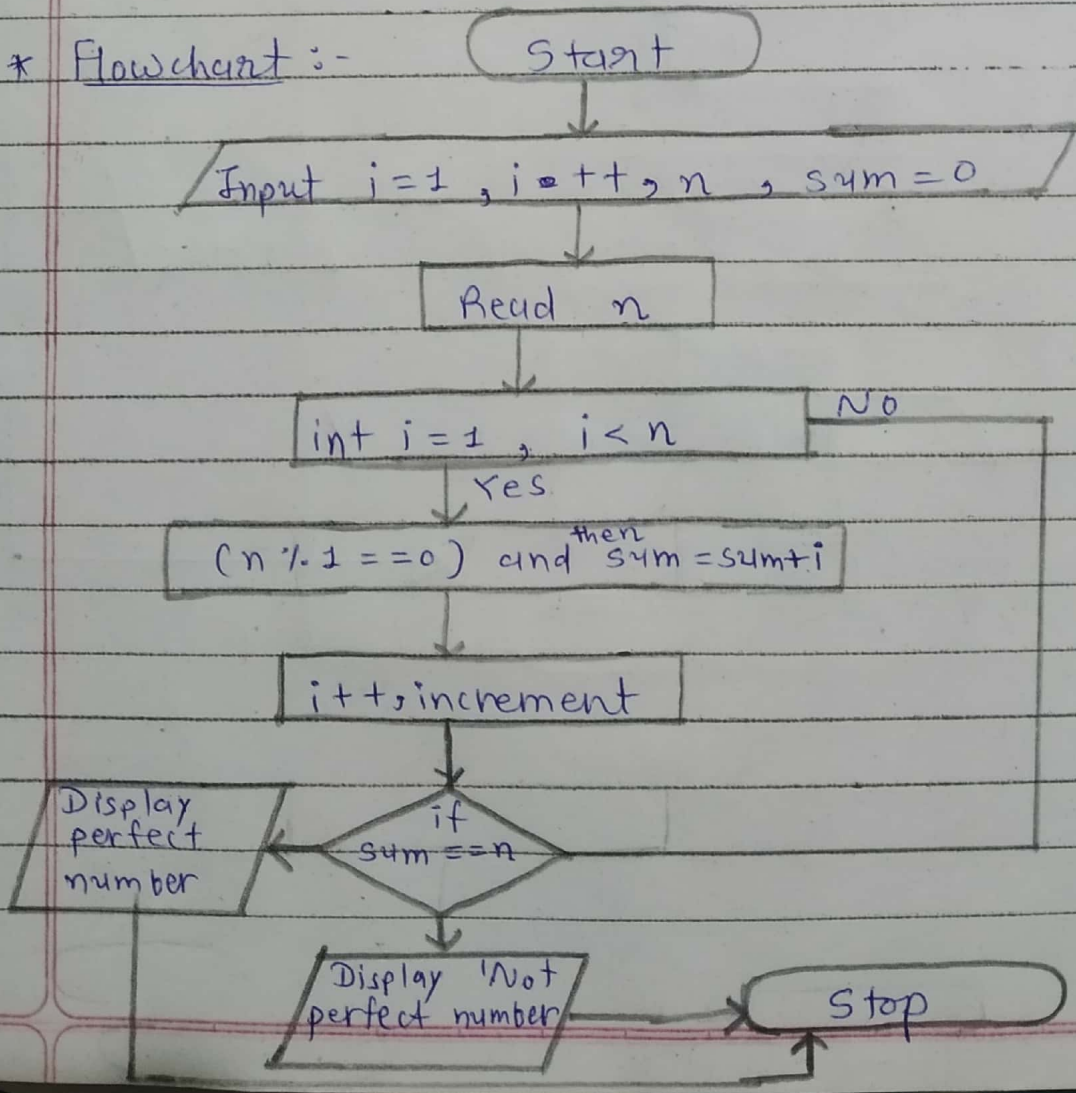
Step : 5 int i = 1, i < n then goto step-6
else goto step-7

Step : 6 if (n % 1 == 0) then sum = sum + i and
i++ increment and goto step-5

Step : 7 if (sum == n) then display 'perfect number'
else display 'not perfect number'

Step : 8 Stop

* Flowchart :-



20DCS103

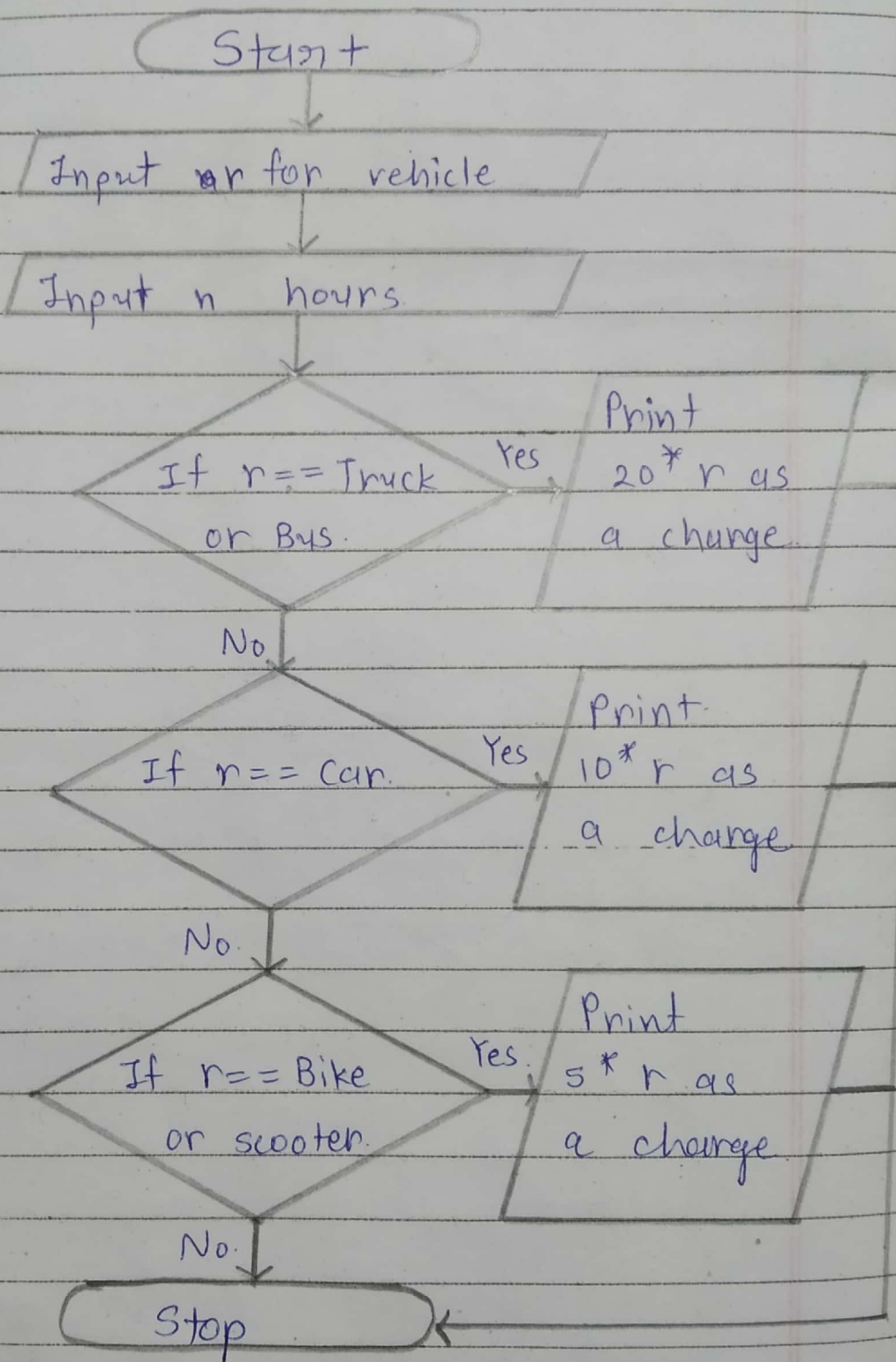
6) Write an algorithm and draw a flowchart to calculate the parking charges of a vehicle. Input type of a vehicle (bus, truck, car) and number of hours. Calculate the charges as given below : truck/bus - 20 rs per hour, car - 10 rs per hour, scooter/bike - 5 rs per hour.

* Algorithm :-

Step: 1 Start.
Step: 2 Take variables r, n, x.
Step: 3 Input r 1 for Truck or Bus.
2 for Car
3 for Scooter or Bike.
Step: 4 Read r
Step: 5 Input number of hours : n
Step: 6 Read n
Step: 7 if (r==1) then x = 20*n
Display x as a charge.
else if (n==2) then x = 10*n.
Display x as a charge.
else if (n==3) then x = 5*n
Display x as a charge.
Step: 8 Stop.

20DCS103

* Flowchart :-



20DCS103

7) Define : Debugging, Interpreter, Compiler, Keyword, Token.

→ Debugging : It is a methodical process of finding and reducing the number of bugs or defects in a computer program, thus making it behave as originally expected.

→ Interpreter : An interpreter directly executes instructions written in a programming or scripting language without previously converting them to an object code or machine code.
Ex : BASIC, Perl, Python, etc.

→ Compiler : A compiler is a special program that processes statements written in a particular programming language and turn them into machine language or "code" that computer's processor uses.
Ex :- C Compiler.

→ Keyword : Keywords are reserved words with predefined meaning which already known to the compiler.
Ex : double, break, etc.

→ Token : Basically, every C program is a collection of instructions and every instruction is a collection of some individual units, which are called Token. Moreover, they are said to the basic building blocks of a C program.
Ex :- Keywords, Constants, etc..

20DCS103

8) What is difference between Header file and library file ? What is the purpose of header file ? Is the use of header file absolutely required ?

Header files	Library files
→ Header files contain function declaration	→ Library files contain function definations.
→ They have the extension .h	→ They have the extension .lib
→ Header files are human readable as they are in the form of source code	→ Library files are not human readable as they are in the form of machine code.
→ They are available inside "include sub directory" which itself is in Turbo compiler.	→ They are available inside "lib sub directory" which itself is in a Turbo Compiler.
→ Header files in our program are included by using a command #include which is internally handle by pre-processor.	→ Library files in our program are included in last stage by special software called as a linker.

→ The purpose of header files :-

→ System header files declare the interfaces to parts of the operating system. It needs to include them in the program to supply the definitions and declarations one needs to invoke system calls and libraries. Also, that is the reason why the use of header files are absolutely required.

9) Explain datatype in C with appropriate example.

→ Datatypes :-

Datatype is a set of value with predefined characteristics which are used to declare variable, constants, arrays, pointers and functions.

→ There are four primary datatypes in c language:

1) Integer :-

Integers are whole numbers which means numbers without decimal point. The keyword "int" is used to define or represent integer datatype in c language. This datatype is used ~~to~~ with different type modifiers like short, long, signed and unsigned.

Ex:- 23, 45, etc...

2) Floating point and double types :-

The numbers with decimal point are called floating point (real) number. The floating point datatype has two variants float and double.

Both float and double are similar but differ in number of decimal places.

Ex:- 2.5, 1.02, etc...

3). Character :-

A single character can be defined as character (char) type data. Character datatypes is a set of characters enclosed in single quotations.

Ex:- 'A', 'b', etc...

4). Void :-

The void type has no value. It is usually used to specify the type of the function. The type of the function is said to be void if it does not return any value.

Ex:- main(void), etc...

10) Define the rules for variable and constant declaration (#define and const).

** Rules for variable :-

1. They must begin with a letter.
2. Variable names may consist of letters, digits and the underscore character.
3. Uppercase and lowercase are significant.
4. ANSI standard recognizes a length of 31 characters. However, the length should not be

more than 8 characters.
5. It should not be a keyword.
6. White space is not allowed.

** Rules for constant declaration (#define):-

*. To create constant using #define preprocessor directive it must be defined at the beginning of the program because all the preprocessor directives must be written before the global declaration

Ex:- syntax : #define CONSTANTNAME value
#define PI 3.14.

*. Constant declaration using 'const' keyword:-

→ To create a constant, we prefix the variable declaration with 'const' keyword.

→ Ex:- syntax const datatype constantName ;
const int x = 10 ;
where 'x' is an integer constant with fixed value 10.

11) Explain preprocessor directive named Macro Expansion without argument and with argument? What do you mean by Macro Templates (symbolic name) and Macro Expansion (Symbolic constant)? Explain two advantages of using symbolic constant. How do variable and symbolic name differ?

20DCS103

→ Macro expansion is a tricky operation, fraught with nasty corner cases and situations that render what one thought was a nifty way to optimize the preprocessor's expansion algorith wrong in quite subtle ways.

→ Functions like macros can take arguments, just like true functions. To define a macro that uses arguments, one needs to insert parameters between the pair of parentheses in the macro definition that make the macro function-like.

→ Macro expansion (symbolic constants) are the name that substitute for a sequence of character that cannot be changed. The character may represent a numeric constant, a character constant or a string. When the program is compiled, each occurrence of a symbolic constants is replaced by its corresponding character sequence. They are usually defined at the beginning of the program. They symbolic constants may then appear later in the program In place of the numeric constants, character constants, etc., that the symbolic constants represent. Here are some advantages :

→ They can be used to assign names to values.
→ Replacement of value has to be done at one place and whenever the name appears in the text it gets the value by execution of the preprocesor.

20DCS103

12) What is role of comments in programming language.

→ Comments play pivotal role in the programming language with different aspects. Here are some crucial roles of comments in a programming language.

1. Code description :

→ Code description is used by the programmer to make others understand his/her intent. It contains the summary of the code.

2. Planning and reviewing :

→ In comments, we can write the pseudocode which we planned before writing the source code. Pseudocode is a mixture of natural language and high-level programming language. This helps in reviewing the source code more easily because pseudocode is more understandable than the program.

3. Resource inclusion :

→ Logos, diagrams and flowcharts consisting of ASCII art constructions can be inserted into source code formatted as a comment. Further, copyright notices can be embedded within source code as comments.

4. Algorithm description :

→ Comments are used for explanation of the methodology. Such explanations may include

diagrams and formal mathematical proofs. This may constitute the explanation of the code, rather than a clarification of its intent. For example, a programmer may add a comment to explain why an insertion sort was chosen instead of quicksort, as the former is, in theory, slower than the latter.

13). Draw and explain the basic structure of C program.

→ A C program may contain one or more sections shown below.

Documentation section
Link section
Definition section
Global declaration section
main () Function section
{
Declaration part
Executable part

}
Subprogram section

Function 1
Function 2
Function 3

Function n

20DCS103

1. Documentation section :-

→ The documentation section consists of a set of comment lines which includes the name of the program, the author and the other details for the future use.

2. Link section :-

→ The link section gives instructions to the compiler to link functions from the system library.

3. Definition section :-

→ The definition section defines all symbolic constants. These symbolic constants are generally written in uppercase so that they are distinguished from lowercase variable names.

4. Global declaration section :-

→ In the global declaration section, global variables are declared which can be used in all functions.

5. main() function section :-

→ In all c program, there must be one main() function section. The main() function allowed some formats like main(), main(void), int main(), void main(), etc...

The main() function contains two parts :

A) Declaration part

B) Executable part

→ All statements in these parts end with a semicolon.

20DCS103

A) Declaration part :-
The declaration part declares all the variables.

B) Executable part :-
There must be at least one statement in the executable part.

6. The subprogram section :-

→ The subprogram section consists of all the user defined functions that are called in the main function.

14) Write a c program to find largest among four numbers using ternary operator.

* c program :-

```
#include <stdio.h>

void main()
{
    int a,b,c,d,max;
    printf("Enter four numbers...\n");
    scanf("%d %d %d %d",&a,&b,&c,&d);

    max=(a>b && a>c && a>d) ? a : ((b>c && b>d)
        ? b : (c>d ? c : d));

    printf("The largest number among %d, %d, %d
        and %d is %d.",a,b,c,d,max);
}
```

20DCS103

15. Which of the following data declaration statements are illegal and why ?

→ 1.) int Alf, Bert=4, Cleo=4.3, Doris='D';
→ This is a legal data declaration statement as it is written according to the rules.

→ 2.) char Eric=257, Eric_Again, 3rd_Eric;
→ This is an illegal statement because variable 3rd-Eric starts with a number. It must be letter instead of a number.

3.) short Default, Default_Value, default, default2;
→ This is an illegal statement because it consists 'default' which is keyword and it must not be a keyword in data declaration statements.

4.) long int, Fred=123456789;
→ This is an illegal statement as it includes (,) in between which is not permitted in case of data declaration statement.

5.) Float x=123.456, Y=100 e-6;
→ This is an illegal statement because "Float" is not datatype as it starts with capital 'F'. Also, the white space between 100 and e-6 is not permittable in data declaration statements.

20DCS103

6) Unsigned negative = -1 ;

→ This is a legal statement.

7) const int three=4, Max, Enic=0 ;

→ This is a legal statement.

8) unsigned float George = 1.234 ;

→ This is an illegal statement because 'unsigned' and 'float' are declaration specifier which can not be used at the same time.

16) Given the following data declaration statements:
int Ali, John=2, Steve=3;
Assuming Integers are stored in 16 bits and long integers in 32 bits, what is the value in each variable after each of the following statements :

→ Here, int Ali, John=2, Steve=3 ;

1). Ali = John * 2 + Steve / Steve ;

=> Ali = 2 * 2 + 3 / 3

= 4 + 1

Ali = 5

2). Ali = John << 2 + Steve ;

→ Here 2 + Steve = 2 + 3 = 5.

and John << 2 + Steve ;

=> John << 2 + 3 ;

=> John << 5 ;

20DCS103

Here, John << 5 means
John x $(2)^5$
$= 2 \times (2)^5$ | ∵ John = 2 ;
$= 64.$

Therefore, Ali = 64 , John = 2 , Steve = 3.

3). Ali = 12 / John * Steve ;
Here, $Ali = \frac{12}{John} \times Steve$

$= \frac{12}{2} \times 3$

$= 18$

Therefore, Ali = 18 , John = 2 , Steve = 3

4). Ali = ((-1^ John) & 7) | Steve ;
→ Here ^ is Bitwise XOR, & is Bitwise AND, | is Bitwise OR. } operators are used in this statement.
By utilizing all the operators in this statement we can get the different values of Ali, John and Steve. which are written below.
Ali = 7 , John = 2 , Steve = 3.

5). Ali = John++ + ++Steve;
→ Here, ++ is an increment operator.
Firstly John++ => John + 1
=> 2 + 1
=> 3.
Secondly ++Steve => steve + 1
=> 3 + 1 => 4

Now, for the entire statement.

Ali = John ++ + ++ Steve

= 20 + 4

= 6

Here, John ++ is post-increment operator.

Therefore, Ali = 6, John = 2, Steve = 3

17) Write C program to convert numbers of days into years, weeks and days. Expected output:

Output:-
Total days: 925
Years: 2
Weeks: 27
Days: 6

Program:

```
1  #include <stdio.h>
2
3  void main()
4  {
5    int n;
6    printf("Enter the number of days to convent... \n");
7    scanf("%d",&n);
8
9    printf("Years: %d \n", n/365);
10   printf("Weeks: %d \n", (n%365)/7);
11   printf("Days: %d \n", (n%365)%7);
12 }
```