DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE, PERAMBALUR
DEPARTMENT OF MCA
CA5103 – PROBLEM SOLVING AND PROGRAMMING TECHNIQUES
QUESTION BANK

PART – A QUESTIONS

UNIT - I

1. What is program?

   Set of explicit and unambiguous instruction expressed in a language. This set of instruction is called program.

2. Define algorithm?

   Algorithm is a set of explicit and unambiguous finite step which, when carried out for a given set of initial condition, produce the corresponding output and terminate in a finite time.

3. What is problem solving?

   Problem solving is a creative process which largely defines systematization and mechanization. In problem solving there a number of steps that can be taken to raise the level of one's performance.

4. What is problem definite phase?

   Useful progress in solving a problem initial we fully understand and trying to solve is called problem definition phase.

5. Dynamic programming?

   This method is used most often when we have to build up a solution to problem via a sequence of intermediate steps. The technique of Greedy search, Backtracking and branch and bound evaluation are variation in dynamic programming.

6. Define Top-Down design?

   A technique for algorithm design that time to accommodate the human limitation known as top down design or stepwise refinement.

7. How to construct a loop?

   To construct loop three things are done,

   (1) initial condition

   That need to apply before the loop begins to execute.

   (2) invariant relation

   That must apply after iteration of the loop.

   (3) terminate

   The condition under which the iterative process must terminate.

8. How to establish initial condition for loops?

   The smallest problem in this case corresponds to the sum of zero numbers. The sum of zero numbers is zero and the initial value of i and s

   i=0 s=0 =>solution foun=0

9. How to find the iterative construct?

   The process we have through to construct the loop is very similar to that of the mathematical induction.

10. What are the implementation algorithms?

    The implementation of an algorithm that has been properly designed is altop down design, there are

    - Use of procedure to emphasize modularity.
    - Choice of variables name.
    - Debugging programs.
    - Documentation of program.
    - Program testing

11. Define program verification?

Program verification refers to the application of mathematical proof techniques to establish that the result obtained by the exaction of a program with arbitrary inputs in accord with formally defined output specification.

12. State transition?

The progress towards completion is made by changing the values of a variable followed by a transfer of control to the next instruction on the current execution path.

13. What is input assertion and output assertion?

Input assertion:

The input assertion should specify any constraints that have been placed on the values of the input variables used by the program. It is expressed in logic notation.

Output assertion:

It must specify symbolically the result that the program is expected to produce for input data that satisfy the input assertion.

14. What is general form of implication?

$P \ni Q$

$\ni$ - logical connective .

P - is termed the assumption.

Q – conclusion.

15. Show the truth table for defining implication?

P	Q	$P \ni Q$
True	True	True
True	False	False
False	True	True
False	False	True

16. What is symbolic execution?

Symbolic execution enables us to transform the verification procedure into proving that the input assertion with symbolic values substituted for all input variables.

17. What is loop invariant?

A loop invariant should be a properly that capture the progressive computational role of the loop while at the same time of remaining true before and after each loop traversal.

18. What are the steps to verify a loop in symbolic execution?

Step1: Vc(A-B): $\ni I_B$

Step2: Vc(B-B):$I_B \wedge C_B \ni I_B$

Step3: Vc(B-C): $I_B \sim C_B \ni P_C$

19. What is partially correct?

The complete program segment is partially correct, with the implementation the result produce will be correct.

20. What are the two conditions for proof termination?

The expression € is greater than zero.

TC1 (B): $I_B \wedge C_B \ni € > 0$

The expression before execution is strictly greater than it value € after loop execution

TC2 (B-B): $I_B C_B \ni (€_0 > €) \wedge (€ \geq 0)$

21. What is the resource most relevant in relation to efficiency?

- ❖ Central processor time
- ❖ Internal memory

Because of high cost of computing resources.

22. What are the suggestions in designing efficient algorithm?

- ➢ Redundant computation.
- ➢ Referencing array element.
- ➢ Inefficiency due to late termination.
- ➢ Early detection storage.
- ➢ Trading storage for efficiency gains.

23. What are the qualities and capabilities of algorithm?

- They are simple but powerful and general solution.
- They can be understood by others.
- They can be easily modified if necessary.
- They are correct for clearly defined situations,
- They are able to understand on a number of level
- They are use of economical in computer line computer storage.
- They are not dependent on being and satisfy to its designer.

24. What is computational complexity?

The essence of the computation while at the same time it must be divorced from any programming language.

25. What is order notation?

It is referred to as the 0-notation an algorithm in which the dominant mechanism is executed en^2 times for C is a constant and n problem size is said to have an order n^2
Written in $0(n^2)$.

26. What are the behaviors of algorithm?

1. Worst case
2. Best case

27. What is worst case complexity?

The size n corresponds to the maximum complexity for encountered among all problem of size n. This determines worst case complexity.

28. What average case?

It is generally assumed that all possible points of termination are equally likely. The average search cost is sum of all possible search costs.

Average search cost = (m+1)/2

29. Algorithm description for exchanging two variables?

1. Save the original value of a in t.
2. Assign to the original value of b.
3. Assign to b the original value of that is stored in t.

30. Write the application of exchanging two variables?

* Sorting Algorithm

31. What is counting?

Counting mechanism is very frequently used in computer algorithm. Generally a count must be made of the number of item in set which posses some particular property.

32. What is exchanging?

The exchange mechanism as a programming tool is most usefully implemented as a proceeding that accepts two variables and returns their exchanged value.

33. Write the application for counting?

- All form of counting.

34. What is factorial computation?

E.g. n!=1*2*3*……(n-1)*n form >=1 by definition

0! =1

Formulating the computer arithmetic unit can only multiply two numbers at a time.

35. Write down the application for factorial computation?

- ✓ Probability
- ✓ Statiscal.
- ✓ Mathematical computation

36. What is Sine function computation?

It is define by the infinite series of expansion

$\text{Sin}(x) = X / 1! - X^3/3! + X^5 / 5! - X^7 / 7!$

37. Write the application for Sine function computation?

- ♣ Mathematical
- ♣ Statistical computation.

38. What is base Conversion?

Frequently in computing it is necessary to convert a decimal number to the binary, octal or hexadecimal number system. Such conversion is base conversion.

39. What are the applications for the base conversion?

Interruptions of stored computer data and instructions.

UNIT -2

1. How to find a complementary value?

The complementary value expression on

G2 :=(gl+(m1gl)12

That is an formula on the to find in the complementary value.

2. How are we terminating the iterative process?

We seem to have no way of knowing in advance how many interactions will be needed in the general case to calculate an acceptable value for a given square root. We are therefore going to need some other criterion for stopping the iterative process.

3. How should we choose our initial guess?

Our considerations so far tell us that in one sense it does not matter much what initial guess we make since the balancing of our algorithm will ensure we fairly rapidly coverage on an acceptable square root.

4. What is a feedback principle?

- That is, we keep making correction to our estimate in a way dependent on how much the previous solution deviated from the desired result.
- The more drastic the deviation the more drastic correction.

5. What dijkstra method used?

Dijkstra gives an interesting and efficient method for finding the smallest prime factor of large number.

6. What is mean by Eudorus?

Euclid is published one algorithm 2000 year ago. That is, which has come to be known as Euclid's algorithm, was probably invented by a predecessor of Euclid's called eudorus.

7. What is the basic strategy for computing the god of two numbers?
    - Divide the large of the two numbers by the smaller number.
    - If the smaller number exactly divided into the larger number exactly divides number is the god.

8. What purpose to use a mod function?

    The mod function allows us to compute the remainder resulting from an integer division. We can use

    R: n mod m

9. Which condition to use to terminate the loop?

    While non-zero remainder do
    "Continue search for gcd"
    That condition to terminate the loop.

10. What is advantage of sieve method?

    The advantage of sieve method has is that it does not involve any costly division to establish the primes.

11. What is a use of pseudo-random numbers?

    The uniform distribution of pseudo random numbers can be used as a basis for generating other distributions such as the normal and exponential distributions.

12. What is a power of an array?

    The power of the array is largely derived from the fact that it provides us with a very simple and efficient way of retiring to and performing computations on collection of data that share some common attribute.

13. What is on pincer process?

    That is a two partition will eventually meet and when they do we will have the desired partition for our complete data set.

14. How to finding the square root of a number?
    - Choose a number n less than the number m we want the square root
    - Square n and if it is greater than m decrease n by 1 and repeat step2 else go to step3.
    - When the square of our guess at the square root is less than m we can start increasing n by 0.1 until we again compute a guess greater than m. At this point we start decreasing our guess by 0.01 and so on until we have computed that square root we require to the desired accurate.

15. What is a basic swatter tour computing the gcd of two numbers?
    - Divide the larger of the two numbers by the smaller numbers
    - If the smaller number exactly divides into the larger number then the smaller number is the gcd.

16. What is a greatest common divisor of two integer's algorithm description?
    - Establish the two positive non-zero integer smaller and larger whose gad is being sought.

    Repeat

    - Get the remainder homes dividing the larger integer's integer by the smaller integers?
    - Let the smaller integer assumes the role of the larger integer.
    - Let the remainder assume the role of the divisor.
    - Return the gad of the original pair of integer.

17. computing the prime factors of an integer define algorithm description?
    - If next prime is divisor of n then save next prime as a factor and reduce n by next prime. Else get next prime.
    - Try next prime as a divisor of n end.

18. Define pseudo-random numbers algorithm development?
    - The sequence should appear as though each number had occurred by chance.
    - Each number should have a specified probability of falling within a given range.

19. What is a power generation condition?
    - Where we have an odd power it must have been generated from the power that is an one less.
    - Where we have on even power, it can be computed from a power that is half is size.

20. What is an Fibonacci number needed two parts?
    1. A part of that determines that doublings strategy by generating the binary representation of n9 and
    2. A second part that actually computes the nth Fibonacci according to the doubling swatergy.

21. What is array ordered reversal algorithm description?
    - Establish the array a[1,n] of n elements to be reversed.
    - To computer r the number of exchanges needed to reverse the array
    - While there are still pairs of array elements to be exchanged.
    - Return the reversed array.

22. Which the three situation possible to finding the maximum number in a set?
    - The second number can be less than our temporary candidate can be less than our temporary candidate for the maximum.
    - The second number can be equal to our temporary candidate for the maximum
    - The second dumber can be greater than our temporary candidate for the maximum.

23. What is an removal of duplicate two situations?
    - A pair of duplicate has been encountered.
    - The two elements are different.

24. What is an sub sequence termination position?
    - Append the current element on to the largest of the preceding subs sequences that allows the monotone increasing extensions to be maintained.
    - Append the ith element to the appropriate preceding monotone increasing subsequence and update the maximum subsequence to date as necessary.

25. What is a sorting algorithm usually fall of two classes?
    - The simple and less sophisticated algorithms are characterizers by the fact that they require of the order of n2 comparisons to sort n items.
    - The advanced sorting algorithms take of the order of n log2n comparisons to sort n items of data algorithms within this come close to this optimum possible performance for sorting random data.

26. What is a sorting process?
    - Find the smallest element in the sorted array.
    - Place the smallest element in position a [1]

27. What is a sorting algorithm description?
    - Establish the array a [1...n] of n elements.
    - While there are still elements in the unsorted part of the array do.
    - Find the maximum min and its location p in the unsorted pact of the array.
    - Exchange the minimums min in the unsorted part of the array with the first element a[i] in the sorted array.

28. What is a sorting by exchange algorithms description?
    - Establish the array a [1...n] of n elements.
    - While the array is still not stored do
    - Set the order indictor sorted true.
    - For all adjacent pairs of element in the unsorted part of the array.
    - Return the stored array.

29. What is a sorting by insertion algorithm description?
    - Establish the array a [1...n] of n elements.
    - Find the minimum a3 put it in place to act as sentinel.
    - While there are still elements to be inserted in the ordered part do
    - Select next element x to be inserted
    - While x to the less than preceding element do
    - (b.1) more preceding element up on position.
    - (b.2) extend search back one element function
    - (c) Insert x at current position.

30. What is a development of sorting by diminishing increment?
    The development of these algorithms is slightly more complex than other sorts we have seen. As an example it brings home the importance of good clean top-down design.

31. What is a basic step in the partitioning algorithm?
    - Extend the two partitions inwards until a wrongly partitioned pair is encountered.
    - While the two partitions have not crossed.
    - Exchange the wrongly portioned pair.
    - Extend the two partitions inwards again until another wrongly partitioned pair is encountered.

32. What is general strategy of binary search?
    Examine middle value of remaining data and one of these bases of this comparison eliminates half of the remaining data set.

33. What are the basic steps of a hash searching?
    - Derive a hash value module the table size from the search key.
    - It the key not at hash value index in array then
    - Perform a toward linear search from current position in the array modulo the array size.

34. How to terminate the tree on binary tree?
    - Successfully on finding the key.
    - Unsuccessfully at an empty location.
    - Unsuccessfully with a full table.

UNIT - 3

1. What is character set and use?

   The characters that can be used to from words number and expressions depend upon the computer on which the program is run.

2. Categories of characters?
   - Letters
   - Digits
   - Special characters
   - White space.

3. What is the use of white space?

   The compiler ignores white space unless they are a part of a string constant. White spaces may be used to separate words but are prohibited between the characters of keywords and identifiers.

4. What is meant by trigraph characters?

   Many non-English keyboards do not support all the characters c introduce the concept of tri-graph sequences to provide a way to enter certain character that are not available in some key board.

5. How many character use in trigraph sequence?

   Trigraph sequence consists of three characters two question mark and followed by another character. Ex??=

6. What is meant by constant?

   Constants in c refer to fixed values that do not change during the execution of a program.

7. Rules for identifiers?
   1. First character must be an alphabet.
   2. Must consist of only letters, digit or underscore.
   3. Only first 31 characters are significant.
   4. Cannot use a keyword.
   5. Must not contain whitespace.

8. Types of constants?
   1. Numeric constants.
   2. Character constants.

9. Categories of numeric constants?
   1. Integer constant
   2. Real constant.

10. Categories of character constants?
    1. Single character constant.
    2. String constants.

11. Define integer constant?
    - An integer constant refers to sequence of digits.
    - Define c tokens?
    - The smallest individual unit is called c token.

12. Define tokens?

    Individual words and punctuation marts are called token.

13. Type of tokens?

C has six types of tokens such as

1. Keywords
2. Constants
3. Identifiers
4. Strings
5. Special symbol
6. Operators

14. Define keywords.

Keywords have fixed meanings and these meaning cannot be changed.

15. Define identifiers.

Identifiers refer to the name of variables function and arrays. These are user-defined names and consist of a sequence of letters and digits with a letter as a first character.

16. Type of integer constants?

- There are three types of integers, namely.
- Decimal integer
- Octal integer
- Hexadecimal integer

17. Define decimal integer.

Decimal integer consists of a set of digits o through 9. Preceded by an optional –or-sign.

Example:

123-32

18. Define octal integer.

An octal integer constant consists of any combination of digits from the set o through 7 with leading 0.

Example:

O37

19. Define hexadecimal integer?

A sequence of digits preceded by oX of ox is considered as hexadecimal integer. They include alphabets A through F or a through f .A through F represent the number 10 through 15.

Example:

0x2 0x9f

20. Define real constants?

Whole numbers with decimal point is called as the real constants.

0.0083

21. Representation of real constants?

Two way representation of real constant

1. Decimal notation
2. Scientific notation (or) exponential notation.

22. What is meant by decimal notation?

Decimal notation having a whole numbers followed by a decimal point and the fractional point it is possible to omit digits before the decimal point on digits ofter the decimal point.

Example

215. .95

23. What is mantissa?

The mantissa is either a real number expressed in decimal notation or an integer.

24. What is an exponent?

The exponent is an integer number with an optional plus or minus sign.

25. What is a single character constant?

A single character constant contains a single character enclosed with a pair of single quotation marks.

Example:

‘5’,’x’

26. What is meant by String constant?

A string constant is a sequence of characters enclosed in double quitter. The characters may be letters, numbers, spefial character and bland space.

“Hello” “1987”

27. Backslash character constant?

- C supports some special backslash character constant that are used in output function the symbol ‘\n’ stands for newline character.
- Combination of backslash character is called is known as escape sequence.

28. Define variable? Give some example.

A variable is a data name that may be used to store a data value. A variable may take different values at different times during the execution.

Example:

Average, height

29. List out the Rules for variables.

- They must begin with a letter some system permit underscores as the first character.
- ANSI c recognizes a length of 31 character.
- However, length should not be normally more than eight characters. Since only the first eight characters are treated as significant by many compilers.

30. What is data type?

The variety of data types available allows the programmer to select the type appropriate to the needs of the application as well as the machine.

31. What are the kinds of data types?

ANSIC supports three classes of data types.

1. Primary / fundamental data type.
2. Derived data type.
3. User defined data type.

32. Define void?

- The void type usually used to specify the type of functions and it has no values.
- The type of function is said to be void when it does not return and value to the calling function.

33. How to declare the variable?

- After designing suitable variable names we must declare them to the compiler. Declaration does two things:
- It tells the compiler what the variable name is.
- It specifies what type of data the variable will hold.

34. What are the types of variable declaration?

There are two types of variable declaration such as,

1. Primary type declaration.
2. User defined type declaration.

35. Define primary type declaration? Give example with syntax?

A variable can be used to store a value of any data type.

Syntax:

Data-type v1, v2...vn;

V1, v2...vn are the names of variables. Variables are separated by commas. A declaration statement must end with a semicolon.

36. User-defined type declaration?

C supports a feature known as “type definition” that allows users to define an identifier that would represent and existing data type. The user-defined data type identifier used to declare variable.

Syntax:

Typedef type identifier;

37. What is a advantage of typedef?

The main advantage of typedef is that we can create meaningful data type names for increasing the readability of the program.

38. What is an enumeration constant?

The identifier is a use defined enumerated data type which can be used to declare variables that can have one of the values enclosed within the braces known as enumeration.

39. What is storage class?

Storage class that provides information about their location and visibility. The storage class decides the portion of the program within which the variables are recognized.

40. Types of storage classes?

There are four types of storage classes such as,

1. Auto
2. Register
3. Static
4. Extern

41. What is garbage?

Automatic variables contain undefined valued known as garbage unless they are initialized explicitly.

42. How to assigning the values to variables?

Values can be assigned to variables using the assignment operator as follows.

Variable-name=constants;

43. Define initialization.

The process of giving initial values to variables is called initialization. C permits the initialization of more than one variable in one statement using multiple assignment operators.

44. Define symbolic constants.

- We use certain unique constants in a program these constants may appear repeatedly in a number of places in the program
- One example of such a constant is 3.142, representing the value of the mathematical constant “p:”

45. What are the problems in defining symbolic constants?

- We face two problems in the subsequent use of such programs.
- Problem in modification of the program.
- Problem in understanding the program.

46. What is mean by constant identifiers?

Symbolic names are sometimes called constant identifiers. Since the symbolic names are constants, they do not appear in declarations.

47. How to declare the variable as constants?

We can declare the variable with the qualifier const at the time of initialization.

Example:

Const int class-size=40

48. How to declare a variable as volatile?

ANSI defines another qualifier volatile that could be used to tell explicitly the compiler that a variable's value may be changed at anytime by some external sources.

Example:

Volatile int date;

49. What is data overflow?

When the value of the variable is too big is called data overflow. The largest value that a variable can hold also depends on the machine.

50. What is data underflow?

When the value of the variable is too small that is called data underflow.

51. Define operators? Give examples?

An operator is a symbol that tells the computer to perform certain mathematical of logical manipulations.

Example:

=, +,-,*, &, and<.

52. What are the types of operators?

C operator can be classified into a number of categories. They include,

1. Arithmetic operators
2. Relational operators
3. Logical operators
4. Assignment operators
5. Increment and decrement operators
6. Conditional operators
7. Bitwise operators
8. Special operators

53. What is an expression?

An expressions is a sequence of operands an operators that reduces to a single value.

Example:

10+15 is a expression whose value is 25.

54. Define arithmetic operators with example.

C provides all the basic arithmetic operators. The operators are +,-,*,/ and % all work the same way as they do in other language.

55. Define integer arithmetic.

When both the operands in a single arithmetic expression such as a+b are integers, the expression is called and integer expression and the operation is called integer arithmetic. Integer arithmetic always yields an integer value.

56. Define real arithmetic.

An arithmetic operation involving only real operands is called real arithmetic. A real operand may assume valued either in decimal or exponential notation.

57. Define mixed-mode arithmetic operator?

When one of the operands is real and the other is integer. The expression is called a mixed-mode arithmetic expression.

58. Define relational operation?

- We often compare two quantities and depending on their relation take certain decisions.
- Example, we may compare the age of two persons. These comparison can be done with the help of relational operations.

59. What is relational expression?

An expression such as a<b or 1<20 containing a relational operator is termed as a relational expression. The value of a relational expression is either one or zero. One if specified relation is true and zero if the relation is false.

60. List out the kinds of relational operators.

<, >, <=, >=, ==, !=

61. Define logical operators. Given example.

The logical operators && and || are used when we want to test more than one condition and make decision.

Example:

A>b&&x==10

62. What are the types of logical operators?

C has the following three logical operators such as.

&&	AND
||	OR
!	NOT

63. Define logical expression.

An expression of this kind, which combines two or more relational expressions, is termed as logical expression or a compound relational expression.

64. Define assignment operators.

Assignment operators are used to assign the result of an expression to a variable. We have using the assignments operator'='.

65. What is increment operator and decrement operator?

The increment operator ++ adds 1 to the operands while the decrement operator - - subtracts 1.both are unary operators and takes the following form.

++m orm++

--m or m- -

We use the increment and decrement statement in for and while loops extensively.

66. Define conditional operators and its syntax with example.

A ternary operator "?" is available in c to construct conditional expressions of the form.

Syntax:

Exp1? exp2:exp3

Only one expression is evaluated at a time. We can't evaluate two expressions at a time

Example;

```
A=10;
B=15;
X= (a>b)? a: b;
```

67. Define bitwise operator.

C has a distinction of supporting special operators known as bitwise operators for manipulation of data at bit level. These operators are used for testing the bits, or shifting them right to left bitwise operators may not be a applied to float or double.

68. What are the operators used in bitwise operators?

&	Bitwise	AND
|	Bitwise	OR
^	Bitwise	exclusive OR
<<	Shift left	
>>	shift right	

69. Define special operators.

C supports some special operators such as comma operator, size of operators, pointer operators and member selection operator.

70. What is the use of comma operator?

The comma operator can be used to link the related expressions together. It's evaluated from left to right.

71. What is a size of operator?

The size is a compile time operator and when used with an operand, it return the number of bytes the operand occupies. The operand may be a variable, constant or a data type qualifier.

72. Define arithmetic expression with one example?

An arithmetic expression is a combination of variables, constants and operators arranged as the syntax of the language.

Example:

Algebra exp	c exp
Axb-c	a*b-c

73. Given the syntax for evaluation of expressions.

Expressions are evaluated using an assigned statement of the form.

Variable=expression.

74. What are priority levels of arithmetic operator in c?

There are two distinct priority levels of arithmetic operator in c such as

- High priority * / %
- Low priority + -

75. Define implicitly type conversion.

C allows mixing of constants and variables of different types in an expression. C automatically converts and intermediate values to the proper type. So that the expression can be evaluated with losing any significance. This automatic conversion is known as implicit type conversion.

76. Define duplicity conversion.

- There are instances when we want to force a type conversion in a way that is different from the automatic conversion.
- The process of such a local conversion is known as explicit conversion or casting a value.

Syntax:

(type-name) expression

77. Define operator's precedence.

- Each operators in c has a precedence associated with it. This precedence is used to determine how an expression involving more than one operator is evaluated.
- There are distinct levels of precedence and an operator may belong to one of these levels.

78. Define associatively.

The operators at the higher level of precedence are evaluated first. The operators of the same precedence are evaluated either from left to right or from right to left depending on the level. This is known as associatively.

79. Define mathematical functions.

Mathematical function such as cos, sqrt, log etc are frequently used in analysis of real life problems. Most of the c compiler supports these basic math functions. There are systems that have a more comprehensive math library and one should consult the reference manual to find out which functions are available.

80. Define while statement with syntax? Given example

The simplest of all the looping structures in c is the while statement.

Syntax:

```
While (text condition)
{
        Body of the loop
}
```

The while is an entry controlled loop statement. The test condition is evaluated, and if the condition is true, then the body of the loop is executed. This process is repeated until the test condition finally becomes false and the control is transferred out of the loop.

81. How to reaching a character?

Reaching a single character can be done by using the function get char.

Syntax:

Variable-name=get char ( );

Get char is used to the right hand side of an assignment statement.

82. How to write a single character?

Putchar functions is an analogous function for writing character one at a time to the terminal.

Syntax:

Putchar (variable-name);

83. Define formatted input.

Formatted input refers to an input data that has been arranged in a particular format.\

Example:

15.75 123 john

Syntax:

Scanf (“control string”, arg1,arg2....argn);

84. Define control string?

The control string specifies the field format in which the data is to be entered it is also known as format string.

85. Define inputting integer.

%wd

The percent sign(%) indicates that a conversion specification follows’ is an integer that specifies the fields width of the number to be read %wd known as data type characters.

86. What is meant by inputting real numbers?

The field width of real number is not to be specified and specification % f for both the notation namely decimal point notation and exponential notation.

Example:

Scanf (“%f%f%c”, &x, &y, &z);

87. What is meant by inputting character string using example?

A scanf function can input strings containing more than one character

%ws or%wc

%c is used to read a single character.

Example:

Scanf ("%c%c", &a, &b);

88. What is meant by mixed data reading input?

The input line containing mixed mode data it is possible in scanf statement in such cases, are should be exercised to ensure that the input data items match the control specifications in order any type.

Example:

Scanf ("%d%c%f%s", &count, &code, &ratio, name);

89. How detect error in input?

After the compaction of reaching list, it return values of number of items that are successfully read this value can be used to test whether any errors occurred in reading the input.

Example:

Scanf ("%d%f%s", &a, &b, name);

90. What is a rule for scanf statement?

- Each variables to be read must have a filed specification
- For each field's specification, there must be a variables address of proper type.
- Any non-whitespace character used in the format string must have a matching charactering the user input.
- Never end the format string with whitespace it is a fatal error.
- The scanf reads until
- N white space character is found in a numeric specification or
- The maximum number of character have been read or
- An error is detected is reached
- The end of the file is reached

91. What is meant by formatted output?

Printf function for printing captions and numerical results. The Printf statement provided to commas alignments and spacing of Printf-outs on the terminals.

Syntax:

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

92. What are the types of control string?

Control string consists of three type of item

- Characters that will be printed on the screen as they appear.
- Format specifications that define the output format for display of each item.
- Escape sequence character such as \n \t and \b.

93. What is an output specifications integer number?

- The format specification an integer number %wd.
- Where w specifies the minimum field width for the output.

94. How the output number displayed?

The output of a real number may be displayed in decimal notation using format specification.

%w.pf

W is minimum number of positions to be used for the display of the value and the integer indicates the number of digits to be displayed after the decimal point.

95. How to print a single character to terminal?

- A single character can be displayed in a desired position using the format.

  %wc

- The character will be displayed right justifies in the field of w column. The defaults value for w is.

96. How to print string?

The format specification of outputting string is similar to that of real numbers it,

%w.ps

97. What is meant by mixed mode data output?

It is a mix data types in one Printf statement.

Example:

Printf ("%d%f%s%c", a, b, c, d);

98. List out the decision making statement.

- If statement
- Switch statement
- Conditional operator statement
- Go to statement

99. Define if statement.

The if statement is a powerful decisions making statement and is used to control the flow of execution of statement.

100. List out the types of if statement.

- Simple if statement
- If-else statement
- Nested if else statement
- Else if ladder.

101. What is meant by single statements?

The statement-block may be a single statement or a group of statement is called a single statement.

102. What is meant by if-else statement?

The if-else statement in an extension of the simple if statement.

Syntax:

```
If (test expression)
{
True block statement(s)
}
Else
{
False-block statement(s)
}
Statement-x
```

103. What is meant by Nested if?

If statement has another if statement is called nesting if statement

104. What is the syntax of nesting if?

```
If (test condition-1)
If (test condition-2);
{
Statement-1;
}
Else
{
Statement-2;
}
}
Else
{
Statement-3;
}
Statement-x;
```

105. What is meant by else if ladder?

Multipath decisions are involved in else-if ladder.

Syntax:

```
If (condition 1)
Statement-1;
Else if (condition-2)
Statement-2;
Else if (condition 3)
Statement-3;
Else if (condition n)
Statement-n;
Else
Default-statement;
Statement-x;
```

106. Define switch statement.

Switch is a multiday decision statement. The switch statement tests the value of a given variable against a list of case value and when a match is found. A block of statements associated with that case is executed.

107. What is the syntax of switch statement?

```
Switch (expression)
{
Case value-1;
Block-1;
Break;
Case value-2;
Block-2;
Break;
.....
Default:
Default-block
Break;
}
Statement-x;
```

108. What is meant by conditional operator?

Conditional operator useful for making two way decisions. This operator is a combination of ?: and takes three operands. This operators is popularly known as the conditional operator.

Syntax:

Conditional expression? Expressional: expressional2

109. Which statement is branching statement is an unconditional statement?

Go to statement is an unconditional branching statement.

110. What is meant by label?

A label is any valid variable name and must be followed by a colon. The label is placed immediately before the statement where the control is to be transferred.

111. What a syntax of go to statement?

```
Go to label:          label:
.....                 Statement;
.....                 .....
Label                 .....
Statements            go to label;
```

112. Explain the do statement with syntax?

On some occasions, it might be necessary to execute the body of the loop before the test is performed. Such situations can be handled with the help of the do statement. This takes the form

```
Do
{
Body of the loop
}
While (test condition);
```

113. What is the use of for loop? Give the syntax of for loop

The for loop is another entry-controlled loop that provides a more concise loop control structure. The general form of the for loop is:

```
for (initialization; test condition; increment)
{
Body of the loop
}
```

114. Define nesting of for loop.

Nesting of loops, that is, one for statement within another for statement.

115. What is the difference between pref

Prefix operator	Postfix operator
A prefix operator first adds 1 to operand And then the result is assigned To the variable on left.	A postfix operator first assigns the value To the variable an left And then increments

UNIT - 4

1. Define array.

An array is fixed-size sequenced collection of elements of the same data type it is simply a grouping of like-type data the array can be used to represent a list of numbers or a list of names.

Example:

List of employees in an organization.
Test scores of a class of students.

2. Define data structure:

An array provides a convenient structure for representing data it is classified as one of the data structures.

3. What are the types of array?

(i) One dimensional array
(ii) Two dimensional array
(iii)Three dimensional array

4. Define one dimensional array?

A list of items can be given one variable using only one subscript and such a variable called a single subscripted variable or one dimensional array.

Example:

```
int  number [5];
```

5. Declaration of one-dimensional array.

The array must be declared before they are used that the compiler can allocate space for them in memory.

The general form of array declaration:

Type variable-name [size];

Example:

Float height[50];

6. How to initialize of one dimensional array?

An array is declared its elements must be initialized otherwise they will contain "garbage" the array can be initialized at either of the following stage:

- At compile time
- At run time.

7. Define compile time initialization.

Initialize the elements of arrays in the same way as the ordinary variables are declared.

The general form of initialization of array is

Type array-name[size]={list of values};

Example:

```
int number[3]={0,0,0};
```

8. Define run time initialization.

An array can be explicitly initialized at run time. Its applied for initializing large arrays.

Example:

```
int x[3];
```

9. What are the operations are performed on array?

The two most frequent operations performed on arrays:

- Searching
- Sorting

10. Define sorting and what are the types of sorting.

Sorting is the process of arranging elements in the list according to their values in ascending order.

A sorted list is called an ordered list the three types of sorting.

- Bubble sort
- Selection sort
- Insertion sort

11. Define searching and what are the types of searching.

Searching is the process of finding the location of the specified element in a list. The specified element is often called the search key.

The two types of searching;

- Sequential search
- Binary search

12. Define two dimensional arrays.

The array variables that can store a list of values there could be a table of values are stored.

A two dimensional array requires in two subscripts of row number and column number.

The general form:

Type array name [row-size] [column-size];

Example:

int V[4][3];

13. Initializing two dimensional arrays.

The one dimensional array two dimensional array may be initialized into a list of initial values enclosed in braces.

Examples:

int table [2][3]={0,0,0,1,1,1}

14. Define multi dimensional array.

The arrays of three or more dimensions are determined by the compiler.

The general form of multi dimensional array:

Type array-name [s1][s2][s3]…..[sn];

Example:

int x[3][3][12];

15. Define dynamic arrays.

The dynamic memory of allocation and the arrays create at run time are called dynamic arrays.

16. Define static arrays.

The arrays that receive static memory allocation are called static arrays.

17. Define static memory allocation.

An array created at compile time by specifying size in the source code has a fixed size and cannot be modified are runtime the process of allocating memory at compile time is known as static memory allocation.

18. What are the applications of array?

* Using printers for accessing arrays.
* Passing arrays as functions parameters.
* Arrays as members of structures
* Using structure type data as array elements.
* Arrays as dynamic data structures and
* Manipulating character arrays and strings.

19. Define strings.

A string is a sequence of character that is treated as a single data item.

Example:

Printf (“well done!”);

Output:

Well done!

20. Define character string.

A string is an array of characters. It is declared as follows:

```
Char name [size];
```

Example

```
Char name [20];
```

21. What are the operations performed on character string?

1. Input/output
2. Manipulations
3. Copying one string into another
4. Combining string together
5. Comparisons
6. Extracting a portion of a string.

22. Define initializing string variables.

Character arrays may be initialized when they are declared.

The general form of string variable:

```
Char string name [size];
```

Example:

```
char city [10];
```

22. Why we are using the scanf functions?

The input function scanf can be used with %s format specification to read in a string of characters.

Example:

```
Char address[10];
Scanf ("%s",address);
```

23. Why we are using the getchar() and gets() functions?

To read a single character from the terminal using the function getchar().

The getchar() function call takes the form:

```
Char ch;
Ch= getchar();
```

If the gertchar function has no parameters.

24. Why we are using the printf function?

The printf functions with %s format to print string to the screen. If the format %s can be used to display the null characters

Examples:

```
Printf ("%s", ramesh);
```

26. Why we are using the putchar() and puts() functions?

The put char to output the values of character variables. It takes the following form:

```
Char ch ='r';
Putchar (ch);
```

The function putchar requires one parameter.

27. Define concatenation.

The process of combining two strings together is called concatenation.

28. Define string handling function.

The large number of string handling functions that can be used to carry out many of the string manipulations.

The commonly used string handling function:

Function	Actions
Strcat()	concatenates two strings
Strcmp()	compare two strings
Strcpy()	copies one string over another
Strlen()	finds the length of a string

29. Define strcat() frnction.

The strcat function joins two strings together. It takes the following from:

Strcat(string1,string2);

Example:

strcat(part1,"ramesh");

30. Define strcmp() functions.

The strcmp() functions compare two strings identified by the arguments and has a value0 if they are equal. If they are not, it has the numeric difference between the first non matching characters.

The general form:

Strcmp(string1,string2);

Example:

Strcmp(name1, name2);

Strcmp("ram","ramesh");

31. Define strcpy() function.

The strcpy() works almost like a string assignment operator.

It takes the general form:

Strcpy(string1,string2);

Example:

Strcpy(name,"ramesh");

32. Define strlen() function.

The function counts and returns the number of characters in a string.

It takes the general form:

N=strlen(string);

Example:

X=strlen("amma");

33. What are the other string functions?

The heater file <string.h> contains many more string manipulation functions.

1. Strncpy()
2. Strncmp()
3. Strncat()
4. Strstr

34. Define strncpy().

The function strncpy that copies one string to another left-most n characters.

Example: strncpy(s1,s2,5);

35. Strncmp() functions:

A variation of the function strcmp is the function strncmp.

Strncmp(s1,s2,n);

The left-most n characters of s1to s2

➔ 0 if they equal

➔ Negative number if s1 sub-string is less than s2;and

➔ Positive number, otherwise.

36. Define strncat().

The another concatenation function that takes the three parameters.

Strncat(s1,s2,n);

37. Define strstr.

It is a two-parameter function that can be used to locate a sub-string in a string.

This takes the form:

strstr(s1,s2);

Example:

strstr("malaiyalam","laiya");

The function strstr search the string s1to s2.

38. Define user defined function.

A function defined by users are called user defined function for example we have used only one user defined function called main().

39. What are all elements of user defined function?

There are three elements of user defined function.

- Function definition
- Function call
- Function declaration

40. What is mean by function definition?

It is an independent program module that is specially written to implement the requirement of function.

41. What is meant by function call?

In order to use this function we need to invoke it at a required place in the program. This is known as the function call.

42. Define calling function and function declaration.

- The program that calls the function is referred to as the *calling function.*

- The calling function should declare any function that is to be used data in the program. This is known as function *declaration.*

43. Definition of functions.

It is also known as function implementation shall include the following

- Function name
- Function type
- List of parameter
- Local variable
- Function statement
- A return statement

All the six elements all grouped into two parts namely

- Function header
- Function body.

44. What is a general format for function definition?

```
Function_type function_name (prrameter list)
{
  Local variable declaration;
  Executable statement1;
  Executable statement2;
  …..
  …..
  Return statement;
}
```

45. What is ment by header function?

The header function consists of three parts:

1. The function type
2. The function name and
3. The format parameter list.

It must not consists of semi colon is used a header function.

46. What is function body?

It contains the declaration statements necessary for performing the required task. The body enclosed in braces contains three parts:

1. Local declaration that specify the variables needed by the function
2. Function statement that perform the task of the function.
3. Return statement that returns the value evaluated by the function.

47. What are return values and their types?

A function may or may not send back any value to the calling function. It is does it is done through the return statement.

For example:

```
If (error)
return;
```

48. What is call function?

A function can be called By simply using the function name followed by a list of actual parameters.

```
Main ()
{
        Int y;
        Y=mul(10.5); /* function call */
         Printf ("%d",y);
}
```

49. What is function declaration?

A function declaration consists of four parts

- Function type
- Function name
- Parameter list
- Terminating semicolon

The general format:

Function-type function-name (parameter list);

50. How will declare data type?

A data type declaration may be placed in to two places:

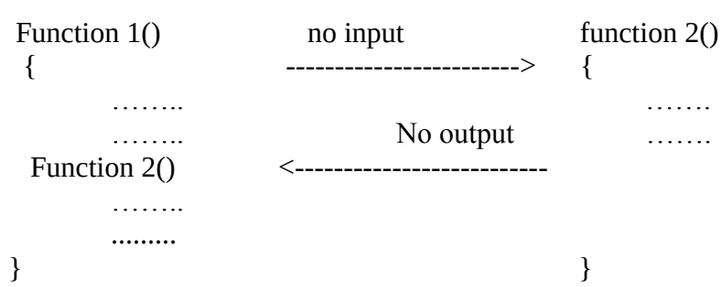
1. Above all the function
2. inside a function definition

51. What are the categories of functions?

- No argument and no return values
- Argument and no return values
- Argument and one return values
- No argument but return values
- Return multiple values
-

52. Define no arguments and no return value.

A function has no argument, it does not receive any data from the calling function.



53. Define arguments but no return values.

The nature of data communication between the calling function and called function with argument but no return value

```
Function 1()            value of argument        function 2(a)
{                       ----------------------->      {
        ……..                                              …….
         ……..              No return value                 …….
Function 2(a)           <-------------------------
        ……..
        .....
}                                                         }
```

54. What are all the rules of pointers?

- The type of the actual and format argument must be same.
- The actual argument must be the address of variable that are local to the calling function.
- The formal argument in the function header must be prefixed by the indirection operation.
- In the prototype the argument must be prefixed by the symbol.

55. Define recursion.

It is a special case of this process, where a function calls itself. A very simple example of recursion is presenter below.

```
Main ()
{
        Printf ("this is an example of recursion");
        Main ();
}
```

56. What is all the relation of array to function?

- The function must be called by passing only the name as the array.
- In the function definition the format parameter must be an array type, the size of the array does not need to be specified.
- The function return type must show that the argument is an array.

57. What are all the variables of storage classes?

The variable storage classes are most relevant:

- Automatic variables
- External variables
- Static variables
- Register variables.

58. Define structure.

It is a convenient tool and handling a group of logically related data items with different data type.

Example:

Set of attribute such as student-name, roll-no, and marks.

59. What is a general format as structure?

```
Struct tag-name
{
        Data-Members
        Data -Members
        ……
        ……
};
```

60. What is declaring structure variable?

A structure variable declaration is similar t declaration of variable of any other data type. It includes the following elements

- The keyword struct.
- The structure tagname
- List of variable names separated by commas.
- A terminating semicolon.

61. What is accessing structure members?

The dot (.) Operator is used to user to individual member of a structure

Syntax:

Struct varname.member name

Example: emp1.name ramesh.

62. Define union.

It is a concept borrowed from structures and there fore follow the same syntax as structure.

```
Union item
{
        int m;
        Float x;
        Char c;
} code;
```

UNIT-5

1. Define pointer?

A pointer is a derived data types, and it is built from one of the fundamental data types. Pointer contain memory address as their values, these memory address are the locations in the computer memory where program instructions and data are stored.

Pointers can be used to access and manipulate data stored in memory.

2. Write the benefits of pointer.

- Pointers are more efficient in handling arrays and data tables. Pointers can be used to return multiple values from a function via function arguments.
- Pointers permit references to functions and there by facilitating passing of functions as arguments to other functions.
- The use of pointers arrays to character strings results in saving of data storage space in memory.
- Pointers provide an efficient tool for manipulating dynamic data structures such as structures, linked lists, queue and trees.

3. What is computer memory?

- The computer memory is a sequential collection of storage cells. Each cell, commonly known as a byte, has a number called address associated with it.
- The addresses are numbered consecutively starting from zero. the last address depends on the memory size. A computer system having 64k memory will have its last address as 65,535

4. What is a pointer variable?

Memory address are simply numbers, they can be assigned to some variables, that can be stored in memory, such variables that hold memory address are called pointer variables.

5. Define-character strings.

- Strings are treated like character arrays and therefore, they are declared and initialized as follows.

  Char str [5]=”good”;

- In c language supports an alternative method to create strings using pointer variable of type char.

  Char*str=”good”

6. Define pointer to functions?

- A function is a variable, has a type and an address location in the memory. It is possible to declare a pointer to a function.
- A pointer to a function is declared as follows

  Type (*fptr) ( );
- This tells the compiler that fptr is a pointer to a function.

7. Define pointer and structures?

The name of an array stands for the address of its zeroth element. The same thing is true of the names of arrays of structure variables.

Example:

```
Struct inventory
{
Char name [30];
int number;
Float price;
} product [2],*ptr;
```

8. What is a precedence operation?

While using structure pointers, it should take care of the precedence of operators. The operators '->' and '.' And [ ] enjoy the highest priority among the operators. They bind very tightly with their operands.

Example:

```
Struct
{
Int count;
Float *p;
} ptr;
```

9. What are the two major problems of i/o?

- It becomes cumbersome and time consuming to handle large volumes of data through terminals.
- The entire data is lost when either the program is terminated or the computer is turned off.

10. Define file.

A file is a place on the disk where a group of related data is stored.

The basic file operations are such as follows

1. Naming a file
2. Opening a file
3. Reading data from a file
4. Writing data from a file
5. Closing a file

11. What the two ways are of perform file operations?

1. The first one is known as the low level i/o and uses UNIX system calls.
2. The second method is referred to as the high-level i/o operation and uses functions in c's standard i/o library.

12. Write any six high level i/o functions.

Function name	operations
fopen ( )	create a new file for use
fclose ( )	close a file which has been opened for use
getc ( )	reads a character from a file
putc ( )	write a character to a file
fprintf ( )	write a set of data values to a file
fscanf ( )	reads a set of data values from a file
getw ( )	reads an integer from a file

13. What is defining and opening a file?

We want to store data in a secondary memory we must specify certain things about the file, to the operating system.

They include

1. File name
2. Data structure
3. Purpose

14. Define file name.

File name is a string characters that make up a valid filename for the operating system. It may contain two parts, a primary name and an optional period with the extension. File is defined data types.

Example:

Input data

Store

Prog.c

Student.c

Text.out

15. Define open a file.

We open a file we must specify what we want to do with the file.

General format,

File *fp;

Fp=Fopen (“file name”, “mode”);

Mode can be one of the following.

R- open the file for reading only.

w-open the file writing only.

a-open the file appending data to it.

16.Define closing a file.

A file must close as soon as all operations on it have been complete. It takes the following form:

fclose (file-pointers);

Example:

```
.....
FILE *f1,*f2;
P1=fopen (“input”, “w”);
P2=fopen (“output”, “r”);
.....
fclose (p1);
fclose (p2);
.....
```

17.Define getc and putc functions.

The simplest file i/o functions are Getc and putc. These are analogous to get char and put char functions and handle one character at a time.

Example:

putc(c, fp1);

Getc is used to read a character from a file that has been opened in read mode

Example:

C=getc (fp2);

18.Define getw and putw functions:

The Getw and putw are integer-oriented functions. They are similar to the get c and put c functions and are used to read and write integer values.

General form:

Put w (integer, fp);

Get w (FP);

19.Define the fprintf and fscanf function.

Most compilers support two other functions, namely Fprintf and Fscanf, that can handle a group of mixed of mixed data simultaneously.

The functions Fprintf and scanf perform i/o operations that are identical to the familiar prints and scanf functions, except of course that they work on files.

General form of Fprintf and Fscanf is

fprintf (fp,"control string:, list

fscanf (fp, "control string", list)

20. What is an error situation in I/O operations?

- Trying to read beyond the end-of-file mark.
- Drive overflow.
- Try to use a file that has not been opened.
- Opening a file with an invalid file name.

21. Define feof functions?

The feof function can be used to test for an end of file condition. It takes a file pointer as its only argument and returns a nonzero integer value if all of the data from the specified file has been read, and returns zero otherwise.

Example:

```
If (feof (fp)
Printf ("end of data.\n);
```

22. Define ferror function.

The ferror function reports the status of the file indicated. It also takes a file pointer as its arguments and returns a nonzero integer if an error has been detected up to that point, during processing.

Example:

```
If (ferror (fp)!=0)
Printf ("an error has occurred.\n");
```

23. Define ftell.

ftell takes a file pointer and return a number of type long, that corresponds to the current position.

This function is useful in saving the current position of a file, which can be used later in the program.

General form is,

```
N=ftell (fp);
```

24. What is rewind function?

Rewind takes a file pointer and resets the position to the start of the file.

Example:

```
Rewind (fp);
N=ftell (fp);
```

25. What is fseek function?

The fseek function is used to move the file position to a desired location with in the file.

General form is,

```
fseek (file-ptr, offset, position)
```

26. Define offset.

Offset is a number (or) variable of type long, and position is an integer number. The offset specifies the number of positions to be moved from the location specified by position.

27. Write the level of position.

There are three level of position is,

Value	meaning
0	beginning of file
1	current position
2	end of file

28. Define dynamic memory allocation?

The process of allocating memory at run time is known as dynamic memory allocation.

29. What is a memory allocation function?

Malloc-allocates request size of bytes and return a pointer to the first byte of the allocated space.

Calloc-allcoate space for any array of elements, initialize them to zero and then returns a pointer to the memory.

Free-free previously allocated space.

Realloc-modifies the size of previously allocated space.

30. Define the permanent storage & stack.

- The program instructions and global, static variables are stored in a region known as permanent storage area and the local variables are stored in another area called stack.
- The memory space that is located between these two regions is available for dynamic allocating during execution of a program. This free memory region is called heap.

31. What is allocating a block of memory?

A block of memory may be allocated using the function Malloc.

- The Malloc function reserves a block of memory specified size and returns a pointer of type void.

  General form is,

  ptr= (cast-type*) Malloc (byte-size);
- The Malloc returns a pointer to an area of memory with size byte size.

Example:

x=(int *)Malloc(100 * size of(int));

32. Allocating multiple blocks of memory: calloc

Calloc is another memory allocation function that is used for requesting memory space at run time for storing derived data types such as arrays and structures.

General form of Calloc is

Ptr=(cast-type *) Calloc (n, elem-size

Example:

```
.................
.....................
Struct student
{
Char name [25];
Float age;
Long int id-num;
};
Type def Struct student record;
Record *st-ptr;
Int class-size=30;
St-ptr=(record *)Calloc(class-size, size of(record));
```

33. Define linked list.

A linked list is a collection of structures ordered not by their physical placement in memory but by logical links that are stored as part of the data in structure itself.

Example:

```
Struct node
{
Int item:
Struct node *next;
};
```

34. What is self-referential structure?

Structures, which contain a member field that points to the same structure type are called self-referential structure.

35. Write the general form of the node.

```
Struct tag-name
{
Type member1;
Type member 2;
.....
.....
Struct tag-name *next;
};
```

36. Define tag-name.

The structure may contain more than one-item with different data types. However, one of the items must be a pointer of the type tag-name.

37. What are the advantages of linked list?

A linked list is dynamic data structure

The linked list over arrays is that linked lists can grow of shrink in size during the execution of a program.

Linked list does not waste memory space. It uses the memory that is just needed for the list at any point of time.

38. Write types of linked lists.

There are different types of linked lists, such as follows

Circularly linked lists

Two-way (or) doubly linked lists

Circular doubly linked lists.

39. Creating a linked list?

The linked list is an abstract data type and performs the following basic operations

1. What Creating a list
2. Traversing the list
3. Counting the items in the list
4. Printing the list
5. Looking up an item for editing (or) printing
6. Inserting an item
7. Deleting an item
8. Concatenating two lists

40. What are the situations available in inserting an item?

Inserting items in three situations

1. Insertion at the front of the list
2. Inserting in the middle of the list
3. Inserting at the end of the list

41. What are the situations available in deleting an item?

Deleting items in three situations

1. Deleting the first item
2. Deleting the last item
3. Deleting between two nodes in the middle of the list

42. Define the application of linked lists.

Lists, queues and stacks are all inherently one dimensional
A tree represents a two dimensional linked list.
The pre-processor

43. Define pre-processor.

- The prepress or is an unique feature of c language. It provides several tools that are unavailable in other high-level language.
- It is used to tools to makes his program easy to read, easy to modify, portable, and more efficient.

44. Write any four pre-processor directives functions.

Directive	function
#define	defines a macro substitution
#ondef	undefined a macro
#include	specifies the file to be included.
#ifdef	test for a macro definition

45. Write the categories of pre-processor directives.

These directives can be divided into three categories

1. Macro substitution directives
2. File inclusion directives
3. Compiler control directives

46. Define macro substitution.

Macro substitution is a process where an identifier in a program is replaced by a predefined string composed of one or more taken.

47. Define macro definition.

The pre-processor accomplishes this task under the direction of #define statement this statement, usually known as a macro definion
The general form,

#define identifier string

48. What is the form of macro substitution?

There are different forms of macro substitution. The most common form are

1. Simple macro substitution
2. Argument macro substitution
3. Nested macro substitution

49. Define simple macro substitution.

Simple string replacement is commonly used to define constants
Example:

```
#define count 100
#define false 0
#define subject 6
#define capital “DELHI”
```

50. Write the macro with arguments.

The pre-processor permits us to define more complex and more useful form of replacement. It takes the form

#define identifier (f1, f2... fn)

Example:

#define cube(x)

51. Define macro call.

Subsequent occurrence of a macro with argument is known as macro call.

52. Define nesting of macros.

We can also use one macro in the definition of another macro. That is macro definition may be nested.

Consider the following macro definition.

```
#define     M          5
#define     N          M+1
#define     CUBE(x)    x*x*x
#define     SIXTH(x)   (CUBE(x)*CUBE(x))
```

53. What is file inclusion?

An external file containing functions or macro definitions can be including as a part of program. So that we need not rewrite those functions or macro definition.

#include “file name”

54. Write the general form of text condition _#if directives?

```
#if constant expression
{
Statement-1;
Statement-2;
.....................
}
#end if
```

55. Define #elif directive?

The 3 elif enables us to establish an “if-else..if” sequence for testing multiple conditions.

The general form of use of # elif is

```
#if expression 1
Statement sequence 1
# elif expression 2
Statement sequence 2
.....................................
#elif expression n
Statement sequence n
# end if
```

56. Define #pragma directive.

The # pragma is an implementation oriented directive that allows us to specify various instructions to be given to the compiler.

#pragma name

Example:

#pragma loop-opt(on)

57. Define # error directive-define?

#error directive is used to produce diagnostic message during debugging.

The general form is,

```
#error error message
Example:
#if! Defined (file-g)
# error no graphics facility
#end if
```

58. Stringizing operator – define.

ANSI c provides an operator # called Stringizing operator. It is use to definition of macro function.

This operator allows a formal argument with a macro definition to be converted to a string.

Example:

```
        #define sum (xy) prints (#xy"=%/n", xy)
Main ( )
{
.............................
Sum (a+b);
.....................
}
```

59. Define-token pasting operator.

The token pasting operator ## defined by ANSI standard enables us to combine two token within a macro definition to form a single token.

Example:

```
#define combine (s1, s2) s1 ##s2
Main ( )
{
.............
.................
Printf ("%f", combine (total, sales));
.................
}
```

PART – B QUESTIONS

UNIT – I

1. List out all the problem solving aspects and top-down design.
2. Write a brief notes on implementation of algorithms and program verification
3. Discuss about efficiency of algorithm and analysis of algorithm.
4. How to exchange the values? Write a program with neat explanations.
5. Write a program for the following:
    i. Factorial of N numbers.
    ii. Sine Series
    iii. Cos Series
    iv. Base Conversion
6. (a). Write a simple program to find out how many of the numbers from 1 to 10 is greater than.
   (b). Write a program in C that prompts the user with the following lines:
    a) Add two integers
    c) Compare two integers for the larger
    t) Test an integer for odd or even
    q) Quit
   (c). Write a function **celsius()** to convert degrees Fahrenheit to degrees Celsius.
   Note: The conversion formula is °C = 5/9 * (°F - 32) (5×3=15 Marks)

7. (a) Write a program to print this triangle:

```
   *
  ***
 *****
*******
*********
***********
```

   (b) Write a program to compute the average of the ten numbers 1, 4, 9, … , 81, 100 (i.e., the average of the squares of the numbers from 1 to 10).

UNIT – II

1. Explain briefly about factoring methods.
2. Write a brief note on array techniques.
3. What is meant by merging and write a algorithm for mergecopy and merge.
4. Describe all the sorting methods with algorithms.
5. Explain briefly about searching techniques.
6. a) Write a C program for hash searching using linear collision.
   b) Illustrate the Program for the following data:
      10, 12, 20, 23, 27, 30, 31, 39, 42, 44, 45, 49, 53, 57, 60.

UNIT – III

1. Write a brief note on constants and variables.
2. Explain briefly on c data types.
3. Explain the concepts of operators and expressions.
4. Discuss about all the managing i/o operations.
5. Explain decision making branching and looping statement.
6. State the differences between switch and if block.

UNIT – IV

1. Explain Arrays and its types.
2. Write a difference between structure and unions.
3. List out all the string handling functions with appropriate example.
4. Describe user defined functions. Write categories of functions with suitable example program.
5. Explain briefly character arrays & strings.
6. a) Write a program for Towers of Honoi problem using recursion.
   b) Write a program to count the number of vowels in a given string.

7. What is the difference between structure and union? Describe with example how a structure can be passed to function.
8. a)What are command line arguments? Explain.
   b) Write a program that reads a line of text from a data file character by character and displays the text on the screen.

UNIT – V

1. Explain the following concepts with suitable example program:
   i. Pointers and Array
   ii. Pointers and Structures
   iii. Chain of Pointers
   iv. Pointers and Character Strings
   v. Pointers within Structures
2. Write a brief note on file management in c with example.
3. Explain briefly what are the dynamic memory allocation functions and Linked Lists?
4. Discuss about the preprocessor with neat syntax and example.
5. Using Pointers,
   a. Write a line-reversing function.
   b. Write the character-counting function.
   c. Write the string-concatenation program
   d. Write the string-replacing function.

6. (a) Write an interactive, file oriented C program that will maintain a list of names, addresses and telephone numbers in alphabetical order. Include a menu that will allow the user to select any of the following features.
   (i) Add a record.
   (ii) Modify an existing record.
   (iii) Retrieve and display an entire record for a given name.
   (iv) Exit.

7. Give Comparison between call by value and call by reference with an example Program.