INSTITUTE OF COMPUTER APPLICATIONS & MANAGEMENT (BVICAM)
(Affiliated to Guru Gobind Singh Indraprastha University, Approved by AICTE, New Delhi)
A-4, Paschim Vihar, Rohtak Road, New Delhi-110063, Visit us at:http://www.bvicam.in/

Course Code: MCA-103 Course Name: Programming in C

Practice Questions (Theory)

Q 1. Elaborate where was C originally developed and by whom? What has been done to standardize the language?

Q 2. Explain the major components of a C program? Discuss the significance attached to the name main?

Q 3. Describe the term arguments'? Where do arguments appear within a C program? What other term is sometimes used for an argument?

Q 4. Summarize the meaning of each of the following program characteristics: integrity, clarity, simplicity, efficiency, modularity and generality. Why is each of these characteristics important?

Q 5. Elaborate the characters comprise the C character set?

Q 6. Summarize the rules for naming identifiers. Are uppercase letters equivalent to lowercase letters? Can digits be included in an identifier name? Can any special characters be included?

Q 7. Detail the mechanism to determine the number of bytes allocated to each data type for a particular C compiler?

Q 8. What is the relative precedence's of the relational, equality and logical operators with respect to one another and with respect to the arithmetic and unary operators? What are their associativities?

Q 9. What are the commonly used input/output functions in C? How are they accessed?

Q 10. If a control string within a scanf function contains multiple character groups, how are the character groups separated? Are whitespace characters required?

Q 11. Compare the syntax associated with statement labels with that of case labels (case prefixes).

Q 12. Suppose function F1 calls function F2 within a C program. Does the order of the function definitions make any difference? Explain?

Q 13. How is an automatic variable defined? How is it initialized? What happens if an automatic variable is not explicitly initialized within a function?

Q 14. In a multifile program, what is the default storage class for a function if a storage class is not explicitly included in the function definition?

Q 15. What are subscripts? How are they written? What restrictions apply to the values that can be assigned to subscripts?

Q 16. When passing an array to a function, how must the array argument be written? How is the corresponding formal argument written?

Q 17. How can a list of strings be stored within a two-dimensional array? How can the individual strings be processed? What library functions are available to simplify string processing?

Q 18. Under what conditions can the elements of a multidimensional array be initialized if the array is defined in terms of an array of pointers?

Q 19. Suppose a formal argument within the definition of the host function p is a pointer to the guest function q. How is the formal argument declared within p? In this declaration, to what does the data type refer? How is function q accessed within function p?

Q 20. The skeletal structure of a C program is shown below.

```
 void funct(int *p);
main ( ) { static int a[5] = {l0, 20, 30, 40, 50);
.....
 f unct (a) ;
 ..... }
void funct(int *p)
{int i, sum = 0;
for (i = 0; i < 5; ++i)
 sum += *(p + i);
printf("sum=%d", sum);
return; }
```

(a) What kind of argument is passed to funct?

(b) What kind of information is returned by f unct?

(c) What kind of formal argument is defined within funct?

(d) What is the purpose of the for loop that appears within f unct?

(e) What value is displayed by the printf statement within funct?

Q 21. Write a complete C program, using pointer notation, that will generate a table containing the following three columns:

t	ae^{bt} sin ct	ae^{bt} cos ct

Structure the program in the following manner: write two special functions, f 1 and f2, where f 1 evaluates the quantity aebt sin ct and f2 evaluates aebt cos ct. Have main enter the values of a, b and c, and then call a function, table-gen, which will generate the actual table. Pass f1 and f2 to table-gen as arguments. Test the program using the values a =2, b =-0.1, c =0.5 where the values oft are 1,2,3, . . . ,60.

Q 22. Consider a very simple customer billing system. In this system the customer records will be stored within an array of structures. Each record will be stored as an individual structure (i.e., as an array element) containing the customer's name, street address, city and state, account number, account status (current, overdue or delinquent), previous balance, current payment., new balance and payment date.

The overall strategy will be to enter each customer record into the computer, updating it as soon as it is entered, to reflect current payments. All of the updated records will then be displayed, along with the current status of each account. The account status will be based upon the size of the current payment relative to the customer's previous balance. The structure declarations are shown below.

```
struct date { int month; int day; int year; };
struct account { char name[80];
char street[80]; char city[ 801;
 int acct-no;
char acct-type;
float oldbalance;
```

float newbalance ;
float payment;
struct date lastpayment; } customer[1001;
Notice that customer is a 100-element array of structures. Thus, each array element (each structure) will represent one customer record. Each structure includes three members that are character-type arrays (name, street and city), and one member that is another structure (last payment). The status of each account will be determined in the following manner: 1. If the current payment is greater than zero but less than 10 percent of the previous outstanding balance, the account will be overdue. 2. If there is an outstanding balance and the current payment is zero, the account will be delinquent. 3. Otherwise, the account will be current.

Q 23. We wish to evaluate the formula y =x", where x and y are floating-point values, and n can be either integer or floating point. If n is an integer, then y can be evaluated by multiplying x by itself an appropriate number of times. For example, the quantity A? Could be expressed in terms of the product (x)(x)(x). On the other hand, if n is a floating-point value, we can write log y =n log x, or y =e(" log *I. In the latter case x must be a positive quantity, since we cannot take the log of zero or a negative quantity. Write a program to realize the above with appropriate union types.

Q 24. Suppose pointer variable points to a structure that contains an array as a member. How can an element of the embedded array be accessed? Illustrate with appropriate example.

Q 25. Define a structure consisting of two floating-point members, called real and imaginary. Include the tag complex within the definition. Declare the variables xl, x2 and x3 to be structures of type complex. Declare a pointer variable, px, which points to a structure of type complex. Write expressions for the structure members in terms of the pointer variable. Declare a one-dimensional, 100-element array called cx whose elements are structures of type complex. Write expressions for the members of the 18th array element (i.e., element number 17).

Q 26. If a program contains four calls to fork( ) one after the other how many total processes would get created?

Q 27. Differentiate between a zombie process and an orphan process?

Q 28. Enlist the purpose of the functions getpid( ), getppid( ), getpppid( ) with appropriate code fragments?

Q 29. How does waitpid( ) prevent creation of Zombie or Orphan processes?

Q 30. Write one or more preprocessor directives for each of the following situations.

(a) If the symbolic constant BOOLEAN has been defined, define the symbolic constants TRUE and FALSE so that their values are 1 and 0, respectively, and negate the definitions of the symbolic constants YES and NO.

(b) If flag has a value of 0, define the symbolic constant COLOR to have a value of 1. Otherwise, if the value of flag is less than 3, define COLOR to have a value of 2; and if the value off lag equals or exceeds 3, define COLOR to have a value of 3.

(c) If the symbolic constant SIZE has the same value as the symbolic constant WIDE, define the symbolic constant WIDTH to have a value of 132; otherwise, define WIDTH to have a value of 80.

(d) Use the “stringizing” operator to define a macro called error (text) that will display text as a string.

(e) Use the “token-pasting” operator to define a macro called error(i) that will print the value of the string variable errori (e.g., errorl).

Q 31. Define an enumeration type called flags, having the following members: first, second, third, fourth and fifth.

Q 32. Summarize the various preprocessor directives, other than #include and #define. Indicate the purpose of the more commonly used directives?

Q 33. Describe the three logical bitwise operators. What is the purpose of each?

Q 34. Describe the precedence and the associativity for each of the logical bitwise operators.

Q 35. Summarize the rules governing the use of the fopen function. Describe the information that is returned by this function

Q 36. Contrast the use of the fread and fwrite functions with the use of the fscanf and fprintf functions. How do the grammatical rules differ? For what kinds of applications is each group of functions well suited?

Q 37. What is the purpose of the fclose function? Must a call to this function appear within a program that utilizes a data file?

Q 38. What is a structure? How does a structure differ from an array?

Q 39. What is the precedence of the period (.)operator? What is its associativity?

Q 40. What is meant by the address of a memory cell? How are addresses usually numbered?

Q 41. What is the purpose of the indirection operator? To what type of operand must the indirection operator be applied?

Q 42. Describe two different ways to specify the address of an array element?

Q 43. What is meant by dynamic memory allocation? What library function is used to allocate memory dynamically? How is the size of the memory block specified? What kind of information is returned by the library function?

Q 44. In what way does an array differ from an ordinary variable?

Q 45. What advantage is there in defining an array size in terms of a symbolic constant rather than a fixed integer quantity?

Q 46. If an array is passed to a function and several of its elements are altered within the function, are these changes recognized in the calling portion of the program? Explain

Q 47. Can an array be passed from a function to the calling portion of the program via a return statement?

Q 48. How are multidimensional arrays defined? Compare with the manner in which one-dimensional arrays are defined?

Q 49. What is meant by the storage class of a variable? Name the four storage-class specifications included in C.

Q 50. What is the purpose of a header file? Is the use of a header file absolutely necessary?

Q 51. Summarize the rules governing the use of the return statement. Can multiple expressions be included in a return statement? Can multiple return statements be included in a function?

Q 52. Summarize the rules associated with the use of the four relational operators, the two equality operators, the two logical connectives and the unary negation operator. What types of operands are used with each type of operator?

Q 53. What is the purpose of the while statement? When is the logical expression evaluated? What is the minimum number of times that a while loop can be executed?

Q 54. Suppose a break statement is included within the innermost of several nested control statements. What happens when the break statement is executed?

Q 55. What is the purpose of the goto statement? How is the associated target statement identified?