Department of Computer Science(S.F)
I-B.Sc.Computer Science – Question Bank – K1 level
18UCS101 – PROGRAMMING IN C

UNIT – I

1)All keywords in C are in _____.
a) LowerCase letters
b) UpperCase letters
c) CamelCase letters
d) None of the mentioned
Answer: a

2) Variable name resolution (number of significant characters for the uniqueness of variable) depends on
_____.
a) Compiler and linker implementations
b) Assemblers and loaders implementations
c) C language
d) None of the mentioned
Answer: a

3) Which is valid C expression?
a) intmy_num = 100,000;
b) intmy_num = 100000;
c) int my num = 1000;
d) int $my_num = 10000;
Answer: b

4) The format identifier '%i' is also used for _____ data type.
a) char
b) int
c) float
d) double
Answer: b

5) Which data type is most suitable for storing a number 65000 in a 32-bit system?
a) signed short
b) unsigned short
c) long
d) int
Answer: b

6) Which of the following is a User-defined data type?
a) typedefint Boolean;
b) typedefenum {Mon, Tue, Wed, Thu, Fri} Workdays;

c) struct {char name[10], int age};
d) all of the mentioned
Answer: d

7) What is the size of an int data type?
a) 4 Bytes
b) 8 Bytes
c) Depends on the system/compiler
d) Cannot be determined
Answer: c

8) What is short int in C programming?
a) The basic data type of C
b) Qualifier
c) Short is the qualifier and int is the basic data type
d) All of the mentioned
Answer: c

UNIT-II

9) What is the precedence of arithmetic operators (from highest to lowest)?
a) %, *, /, +, –
b) %, +, /, *, –
c) +, -, %, *, /
d) %, +, -, *, /
Answer: a

10) Which of the following is not an arithmetic operation?
a) a *= 10;
b) a /= 10;
c) a != 10;
d) a %= 10;
Answer: c

11) Which of the following data type will throw an error on modulus operation(%)?
a) char
b) short
c) int
d) float
Answer: d

12) Which among the following are the fundamental arithmetic operators, i.e, performing the desired operation can be done using that operator only?
a) +, –
b) +, -, %
c) +, -, *, /

d) +, -, *, /, %
Answer: a

13) Are logical operator sequence points?
a) True
b) False
c) Depends on the compiler
d) Depends on the standard
Answer:a
5. Do logical operators in the C language are evaluated with the short circuit?
a) True
b) False
c) Depends on the compiler
d) Depends on the standard
Answer: a

14) What is the result of a logical or relational expression in C?
a) True or False
b) 0 or 1
c) 0 if an expression is false and any positive number if an expression is true
d) None of the mentioned
Answer: b

15) Which among the following is NOT a logical or relational operator?
a) !=
b) ==
c) ||
d) =
Answer: d

16) Relational operators cannot be used on _____
a) structure
b) long
c) strings
d) float
Answer: a

17) function tolower(c) defined in library <ctype.h> works for _____
a) Ascii character set
b) Unicode character set
c) Ascii and utf-8 but not EBCDIC character set
d) Any character set
Answer: d

18) Which type of conversion is NOT accepted?
a) From char to int

b) From float to char pointer
c) From negative int to char
d) From double to char
Answer:b

19) Which of the following type-casting have chances for wrap around?
a) From int to float
b) From int to char
c) From char to short
d) From char to int
Answer: b

20) Which of the following typecasting is accepted by C?
a) Widening conversions
b) Narrowing conversions
c) Widening & Narrowing conversions
d) None of the mentioned
Answer: c

21) For which of the following, "PI++;" code will fail?
a) #define PI 3.14
b) char *PI = "A";
c) float PI = 3.14;
d) none of the Mentioned
Answer: a

22) Operation "a = a * b + a" can also be written as _____
a) a *= b + 1;
b) (c = a * b)!=(a = c + a);
c) a = (b + 1)* a;
d) All of the mentioned
Answer: d

23) Which of the following operators has an associativity from Right to Left?
a) <=
b) <<
c) ==
d) +=
Answer: d

24) Which of the following are unary operators?
a) sizeof
b) –
c) ++
d) all of the mentioned

Answer:d

25) Associativity of an operator is _____
a) Right to Left
b) Left to Right
c) Random fashion
d) Both Right to Left and Left to Right
Answer:d

26) Which of the following method is accepted for assignment?
a) 5 = a = b = c = d;
b) a = b = c = d = 5;
c) a = b = 5 = c = d;
d) None of the mentioned
Answer:b

27) Which of the following is NOT possible with any 2 operators in C?
a) Different precedence, same associativity
b) Different precedence, different associativity
c) Same precedence, different associativity
d) All of the mentioned
Answer:c

28) Which of the following is possible with any 2 operators in C?
a) Same associativity, different precedence
b) Same associativity, same precedence
c) Different associativity, different precedence
d) All of the mentioned
Answer: d

29) Which of the following operators has the lowest precedence?
a) !=
b) &&
c) ?:
d) ,
Answer:d

30) The C code 'for(;;)' represents an infinite loop. It can be terminated by _____
a) break
b) exit(0)
c) abort()
d) terminate
Answer:a

31) Which of the following cannot be used as LHS of the expression in for (exp1;exp2; exp3)?

a) Variable
b) Function
c) typedef
d) macros
Answer: d

32) What is an example of iteration in C?
a) for
b) while
c) do-while
d) all of the mentioned
Answer:d

33 Which loop is most suitable to first perform the operation and then test the condition?
a) for loop
b) while loop
c) do-while loop
d) none of the mentioned
Answer:ca

34) Which keyword can be used for coming out of recursion?
a) break
b) return
c) exit
d) both break and return
Answer: b

35) The keyword 'break' cannot be simply used within _____
a) do-while
b) if-else
c) for
d) while
Answer: b

36) Which keyword is used to come out of a loop only for that iteration?
a) break
b) continue
c) return
d) none of the mentioned
Answer:b

37) goto can be used to jump from main() to within a function.
a) true
b) false
c) depends
d) varies

Answer: b

UNIT-III

38) Which of the following is a correct format for declaration of function?
a) return-type function-name(argument type);
b) return-type function-name(argument type){}
c) return-type (argument type)function-name;
d) all of the mentioned
Answer: a

39) Which of the following function declaration is illegal?
a) int 1bhk(int);
b) int 1bhk(int a);
c) int 2bhk(int*, int []);
d) all of the mentioned
Answer:d

40) The value obtained in the function is given back to main by using _____ keyword.
a) return
b) static
c) new
d) volatile
Answer:a

41)What is the return-type of the function sqrt()?
a) int
b) float
c) double
d) depends on the data type of the parameter
Answer:c

42)What is the default return type if it is not specified in function definition?
a) void
b) int
c) double
d) short int
Answer:b

43) Functions can return structure in C?
a) True
b) False
c) Depends on the compiler
d) Depends on the standard
Answer:a

44) Functions can return enumeration constants in C?
a) true
b) false
c) depends on the compiler
d) depends on the standard
Answer:a

45) What is the scope of an external variable?
a) Whole source file in which it is defined
b) From the point of declaration to the end of the file in which it is defined
c) Any source file in a program
d) From the point of declaration to the end of the file being compiled
Answer:d

46) What is the scope of a function?
a) Whole source file in which it is defined
b) From the point of declaration to the end of the file in which it is defined
c) Any source file in a program
d) From the point of declaration to the end of the file being compiled
Answer:d

47) Array sizes are optional during array declaration by using _____ keyword.
a) auto
b) static
c) extern
d) register
Answer: c

48) Functions have static qualifier for its declaration by default.
a) True
b) False
c) Depends on the compiler
d) Depends on the standard
Answer: b

49)Is initialisation mandatory for local static variables?
a) Yes
b) No
c) Depends on the compiler
d) Depends on the standard
Answer:b

50) Assignment statements assigning value to local static variables are executed only once.
a) True
b) False

c) Depends on the code
d) None of the mentioned
Answer:b

51)What is the format identifier for “static a = 20.5;”?
a) %s
b) %d
c) %f
d) Illegal declaration due to absence of data type
Answer: b

52) When compiler accepts the request to use the variable as a register?
a) It is stored in CPU
b) It is stored in cache memory
c) It is stored in main memory
d) It is stored in secondary memory
Answer:a

53)Which data type can be stored in register?
a) int
b) long
c) float
d) all of the mentioned
Answer:d

54) Which of the following operation is not possible in a register variable?
a) Reading the value into a register variable
b) Copy the value from a memory variable
c) Global declaration of register variable
d) All of the mentioned
Answer:d

55) Which among the following is the correct syntax to declare a static variable register?
a) static register a;
b) register static a;
c) Both static register a; and register static a;
d) We cannot use static and register together
Answer: d

56) Register variables reside in _____
a) stack
b) registers
c) heap
d) main memory
Answer:b

57) What is the scope of an automatic variable?
a) Within the block it appears
b) Within the blocks of the block it appears
c) Until the end of program
d) Within the block it appears & Within the blocks of the block it appears
Answer:d

58) Automatic variables are allocated space in the form of a _____
a) stack
b) queue
c) priority queue
d) random
Answer:a

59) Which of the following is a storage specifier?
a) enum
b) union
c) auto
d) volatile
Answer: c

60) If storage class is not specified for a local variable, then the default class will be auto.
a) True
b) False
c) Depends on the standard
d) None of the mentioned
Answer:a

61) Automatic variables are stored in _____
a) stack
b) data segment
c) register
d) heap
Answer:a

UNIT-IV

62)Property which allows to produce different executable for different platforms in C is called?
a) File inclusion
b) Selective inclusion
c) Conditional compilation
d) Recursive macros
Answer: c

63) What is #include <stdio.h>?
a) Preprocessor directive

b) Inclusion directive
c) File inclusion directive
d) None of the mentioned
Answer:a

64) C preprocessors can have compiler specific features.
a) True
b) False
c) Depends on the standard
d) Depends on the platform
Answer:a

65) C preprocessor is conceptually the first step during compilation.
a) True
b) False
c) Depends on the compiler
d) Depends on the standard
Answer:a

66)Preprocessor feature that supply line numbers and filenames to compiler is called?
a) Selective inclusion
b) macro substitution
c) Concatenation
d) Line control
Answer:d

67)#include<somefile.h> are _____ files and #include “somefile.h” _____ files.
a) Library, Library
b) Library, user-created header
c) User-created header, library
d) They can include all types of file
Answer:d

68)What is a preprocessor?
a) That processes its input data to produce output that is used as input to another program
b) That is nothing but a loader
c) That links various source files
d) All of the mentioned
Answer:a

69)Which of the following are C preprocessors?
a) #ifdef
b) #define
c) #endif
d) all of the mentioned
Answer:d

70) #include statement must be written _____
a) Before main()
b) Before any scanf/printf
c) After main()
d) It can be written anywhere
Answer:b

71) The C-preprocessors are specified with _____symbol.
a) #
b) $
c) ” ”
d) None of the mentioned
Answer: a

72) What is #include directive?
a) Tells the preprocessor to grab the text of a file and place it directly into the current file
b) Statements are not typically placed at the top of a program
c) All of the mentioned
d) None of the mentioned
Answer:a

73) The preprocessor provides the ability for _____
a) The inclusion of header files
b) The inclusion of macro expansions
c) Conditional compilation and line control
d) All of the mentioned
Answer:d

74) If #include is used with file name in angular brackets.
a) The file is searched for in the standard compiler include paths
b) The search path is expanded to include the current source directory
c) The search path will expand
d) None of the mentioned
Answer:a

75)What is the sequence for preprocessor to look for the file within <>?
a) The predefined location then the current directory
b) The current directory then the predefined location
c) The predefined location only
d) The current directory location
Answer:a

76) Which directory the compiler first looks for the file when using #include?
a) Current directory where program is saved
b) C:COMPILERINCLUDE

c) S:SOURCEHEADERS
d) Both C:COMPILERINCLUDE and S:SOURCEHEADERS simultaneously
Answer: b

77) What would happen if you create a file stdio.h and use #include “stdio.h”?
a) The predefined library file will be selected
b) The user-defined library file will be selected
c) Both the files will be included
d) The compiler won’t accept the program
Answer:b

78) If the file name is enclosed in double quotation marks, then _____
a) The preprocessor treats it as a user-defined file
b) The preprocessor treats it as a system-defined file
c) The preprocessor treats it as a user-defined file & system-defined file
d) None of the mentioned
Answer: a

79) If the file name is enclosed in angle brackets, then _____
a) The preprocessor treats it as a user-defined file
b) The preprocessor treats it as a system-defined file
c) The preprocessor treats it as a user-defined file & system-defined file
d) None of the mentioned
Answer:a

80) The “else if” in conditional inclusion is written by?
a) #else if
b) #elseif
c) #elsif
d) #elif
Answer:d

81) For each #if, #ifdef, and #ifndef directive.
a) There are zero or more #elif directives
b) Zero or one #else directive
c) One matching #endif directive
d) All of the mentioned
Answer:d

82)The #else directive is used for _____
a) Conditionally include source text if the previous #if, #ifdef, #ifndef, or #elif test fails
b) Conditionally include source text if a macro name is not defined
c) Conditionally include source text if a macro name is defined
d) Ending conditional text
Answer: a

83) Which is an indirection operator among the following?
a) &
b) *
c) ->
d) .
Answer:b

84) Which of the following does not initialize ptr to null (assuming variable declaration of a as int a=0;)?
a) int *ptr = &a;
b) int *ptr = &a – &a;
c) int *ptr = a – a;
d) All of the mentioned
Answer:a

85) Which of the following can never be sent by call-by-value?
a) Variable
b) Array
c) Structures
d) Both Array and Structures
Answer: b

86)Which type of variables can have same name in different function?
a) global variables
b) static variables
c) Function arguments
d) Both static variables and Function arguments
Answer:d

87) Arguments that take input by user before running a program are called?
a) main function arguments
b) main arguments
c) Command-Line arguments
d) Parameterized arguments
Answer:c

88) What is the maximum number of arguments that can be passed in a single function?
a) 127
b) 253
c) 361
d) No limits in number of arguments
Answer:b

89) Which of the following arithmetic operation can be applied to pointers a and b?
(Assuming initialization as int *a = (int *)2; int *b = (int *)3;)
a) a + b

b) a – b
c) a * b
d) a / b
Answer:b

90) What is the size of *ptr in a 32-bit machine (Assuming initialization as int *ptr = 10;)?
a) 1
b) 2
c) 4
d) 8
Answer:c

91) Which of following logical operation can be applied to pointers?
(Assuming initialization int *a = 2; int *b = 3;)
a) a | b
b) a ^ b
c) a & b
d) None of the mentioned
Answer: d

92) What is the correct syntax to send a 3-dimensional array as a parameter? (Assuming declaration int a[5][4][3];)
a) func(a);
b) func(&a);
c) func(*a);
d) func(**a);
Answer:a

93) What are the applications of a multidimensional array?
a) Matrix-Multiplication
b) Minimum Spanning Tree
c) Finding connectivity between nodes
d) All of the mentioned
Answer:d

94) What are the applications of a multidimensional array?
a) Matrix-Multiplication
b) Minimum Spanning Tree
c) Finding connectivity between nodes
d) All of the mentioned
Answer:a

95) Which of the following is the correct syntax to declare a 3 dimensional array using pointers?
a) char *a[][];
b) char **a[];

c) char ***a;
d) all of the mentioned
Answer:a

UNIT-V

96) What does argc and argv indicate in command-line arguments?
(Assuming: int main(intargc, char *argv[]))
a) argument count, argument variable
b) argument count, argument vector
c) argument control, argument variable
d) argument control, argument vector
Answer:b

97) What type of array is generally generated in Command-line argument?
a) Single dimension array
b) 2-Dimensional Square Array
c) Jagged Array
d) 2-Dimensional Rectangular Array
Answer:c

98) A program that has no command line arguments will have argc _____
a) Zero
b) Negative
c) One
d) Two
Answer: c

99)What is the index of the last argument in command line arguments?
a) argc – 2
b) argc + 1
c) argc
d) argc – 1
Answer: d

100)Which of the following is not possible in C?
a) Array of function pointer
b) Returning a function pointer
c) Comparison of function pointer
d) None of the mentioned
Answer:d

101)Which of the following is a correct syntax to pass a Function Pointer as an argument?
a) void pass(int (*fptr)(int, float, char)){}
b) void pass(*fptr(int, float, char)){}

c) void pass(int (*fptr)){}
d) void pass(*fptr){}
Answer: a

102) How to call a function without using the function name to send parameters?
a) typedefs
b) Function pointer
c) Both typedefs and Function pointer
d) None of the mentioned
Answer: b

103) Which of the following operation is illegal in structures?
a) Typecasting of structure
b) Pointer to a variable of same structure
c) Dynamic allocation of memory for structure
d) All of the mentioned
Answer:a

104)Which of the following is not possible under any scenario?
a) s1 = &s2;
b) s1 = s2;
c) (*s1).number = 10;
d) None of the mentioned
Answer:d

105)Which of the following uses structure?
a) Array of structures
b) Linked Lists
c) Binary Tree
d) All of the mentioned
Answer:d

106)Which of the following reduces the size of a structure?
a) union
b) bit-fields
c) malloc
d) none of the mentioned
Answer:b

107)What is the function of the mode ‘ w+’?
a) create text file for writing, discard previous contents if any
b) create text file for update, discard previous contents if any
c) create text file for writing, do not discard previous contents if any
d) create text file for update, do not discard previous contents if any
Answer:b

108)If the mode includes b after the initial letter, what does it indicates?
a) text file
b) big text file
c) binary file
d) blueprint text
Answer:c

109)flush(NULL) flushes all _____
a) input streams
b) output streams
c) previous contents
d) appended text
Answer:b

110) _____removes the named file, so that a subsequent attempt to open it will fail.
a) remove(const *filename)
b) remove(filename)
c) remove()
d) fclose(filename)
Answer: a

111) What is the function of FILE *tmpfile(void)?
a) creates a temporary file of mode “wb+”
b) creates a temporary file of mode “wb”
c) creates a temporary file of mode ” w”
d) creates a temporary file of mode “w+”
Answer:a

112) What does tmpfile() returns when it could not create the file?
a) stream and NULL
b) only stream
c) only NULL
d) does not return anything
Answer:a

113) EOF is an integer type defined in stdio. hand has a value _____
a) 1
b) 0
c) NULL
d) – 1
Answer:d

114) fwrite() can be used only with files that are opened in binary mode.
a) true
b) false

Answer:a

115)How does C++ compiler differs between overloaded postfix and prefix operators?
a. C++ doesn't allow both operators to be overlaoded in a class
b. A postfix ++ has a dummy parameter
c. A prefix ++ has a dummy parameter
d. By making prefix ++ as a global function and postfix as a member function.
Answer:d

116)Which function will return the current file position for stream?
a) fgetpos()
b) fseek()
c) ftell()
d) fsetpos()
Answer: c

117)Which functions is declared in <errno. h>?
a) fseek()
b) ftell()
c) ferror()
d) fsetpos()
Answer:c

118)setvbuf() and setbuf() function controls buffering for the stream.
a) true
b) false
Answer:a

119) The functions vprintf(), vfprintf(), and vsprintf() are not equivalent to the corresponding printf() functions except the variable argument list.
a) true
b) false
Answer:b

120)The_____function reads atmost one less than the number of characters specified by size from the given stream and it is stored in the string str.
a) fget()
b) fgets()
c) fput()
d) fputs()
Answer: b

121) Choose the correct difference between getc() and fgetc().
a) If it is not a macro, it may evaluate stream more than once
b) if it is amacro, it may not evaluate stream more than once

c) if it is a macro, it may evaluate stream more than once
d) no difference between fgetc() and getc()
Answer:c

122)What type of return-type used in String operations?
a) void only
b) void and (char *) only
c) void and int only
d) void, int and (char *) only
Answer:d

123) String operation such as strcat(s, t), strcmp(s, t), strcpy(s, t) and strlen(s) heavily rely upon.
a) Presence of NULL character
b) Presence of new-line character
c) Presence of any escape sequence
d) None of the mentioned
Answer:a

124)Which pre-defined function returns a pointer to the last occurence of a character in a string?
a) strchr(s, c);
b) strrchr(s, c);
c) strlchr(s, c);
d) strfchr(s, c);
Answer: b

125)Which of the following function compares 2 strings with case-insensitively?
a) strcmp(s, t)
b) strcmpcase(s, t)
c) strcasecmp(s, t)
d) strchr(s, t)
Answer:c

126)Which of the following causes an error?
a) Trying to read a file that doesn't exist
b) Inability to write data in a file
c) Failure to allocate memory with the help of malloc
d) All of the mentioned
Answer:d

127)stderr is similar to?
a) stdin
b) stdout
c) Both stdout and stdin
d) None of the mentioned
Answer:b

128) Which of the following function can be used to terminate the main() function from another function safely?
a) return(expr);
b) exit(expr);
c) abort();
d) both exit(expr); and abort();
Answer: b

129)Which of the following causes an error?
a) Trying to read a file that doesn't exist
b) Inability to write data in a file
c) Failure to allocate memory with the help of malloc
d) All of the mentioned
Answer: d

130)Which of the following is NOT a delimiter for an input in scanf?
a) Enter
b) Space
c) Tab
d) None of the mentioned
Answer: d

****

Department of Computer Science(S.F)
I-B.Sc.Computer Science – Question Bank – K2 level
18UCS101 – PROGRAMMING IN C

UNIT-I

1. What does static variable mean?

Ans: Static variables are the variables which retain their values between the function calls. They are initialized only once their scope is within the function in which they are defined.

2. What is a pointer?

Ans: Pointers are variables which stores the address of another variable. That variable may be a scalar (including another pointer), or an aggregate (array or structure). The pointed-to object may be part of a larger object, such as a field of a structure or an element in an array.

3. What are the uses of a pointer?

Ans: Pointer is used in the following cases

i) It is used to access array elements

ii) It is used for dynamic memory allocation.

iii) It is used in Call by reference

iv) It is used in data structures like trees, graph, linked list etc.

4. What is a structure?

Ans: Structure constitutes a super data type which represents several different data types in a single unit. A structure can be initialized if it is static or global.

5. What is a union?

Ans: Union is a collection of heterogeneous data type but it uses efficient memory utilization technique by allocating enough memory to hold the largest member. Here a single area of memory contains values of different types at different time. A union can never be initialized.

6. What are the differences between structures and union?

Ans: A structure variable contains each of the named members, and its size is large enough to hold all the members. Structure elements are of same size.

A union contains one of the named members at a given time and is large enough to hold the largest member. Union element can be of different sizes.

7. What are the differences between structures and arrays?

Ans: Structure is a collection of heterogeneous data type but array is a collection of homogeneous data types.

Array

1-It is a collection of data items of same data type.

2-It has declaration only

3-.There is no keyword.

4- array name represent the address of the starting element.

Structure

1-It is a collection of data items of different data type.

2- It has declaration and definition

3- keyword struct is used

4-Structure name is known as tag it is the short hand notation of the declaration.

8. In header files whether functions are declared or defined?

Ans: Functions are declared within header file. That is function prototypes exist in a header file,not function bodies. They are defined in library (lib).

9. What are the differences between malloc () and calloc ()?

Ans: Malloc Calloc 1-Malloc takes one argument Malloc(a);where a number of bytes 2-memory allocated contains garbage values

1-Calloc takes two arguments Calloc(b,c) where b no of object and c size of object

2-It initializes the contains of block of memory to zerosMalloc takes one argument, memory allocated contains garbage values.

It allocates contiguous memory locations. Calloc takes two arguments, memory allocated contains all zeros, and the memory allocated is not contiguous.

10. What are macros? What are its advantages and disadvantages?

Ans: Macros are abbreviations for lengthy and frequently used statements. When a macro is called the entire code is substituted by a single line though the macro definition is of several lines.

The advantage of macro is that it reduces the time taken for control transfer as in case of function. The disadvantage of it is here the entire code is substituted so the program becomes lengthy if a macro is called several times.

UNIT-II

11. Difference between pass by reference and pass by value?

Ans: Pass by reference passes a pointer to the value. This allows the callee to modify the variable directly.Pass by value gives a copy of the value to the callee. This allows the callee to modify the value without modifying the variable. (In other words, the callee simply cannot modify the variable, since it lacks a reference to it.)

12. What is static identifier?

Ans: A file-scope variable that is declared static is visible only to functions within that file. A function-scope or block-scope variable that is declared as static is visible only within that scope. Furthermore, static variables only have a single instance. In the case of function- or block-scope variables, this means that the variable is not ―automatic‖ and thus retains its value across function invocations.

13. Where is the auto variables stored?

Ans: Auto variables can be stored anywhere, so long as recursion works. Practically, they‘re stored onthe stack. It is not necessary that always a stack exist. You could theoretically allocate functioninvocation records from the heap.

14. Where does global, static, and local, register variables, free memory and C Program instructions get stored?

Ans: Global: Wherever the linker puts them. Typically the ―BSS segment‖ on many platforms. Static: Again, wherever the linker puts them. Often, they‘re intermixed with the globals. The only difference between globals and statics is whether the linker will resolve the symbols across compilation units.Local: Typically on the stack, unless the variable gets register allocated and never spills.Register: Nowadays, these are equivalent to ―Local‖ variables. They live on the stackunless they get register-allocated.

15. Difference between arrays and linked list?

Ans: An array is a repeated pattern of variables in contiguous storage. A linked list is a set of structures scattered through memory, held together by pointers in each element that point to the next element. With an array, we can (on most architectures) move from one element to the next by adding a fixed constant to the integer value of the pointer. With a linked list, there is a ―next‖ pointer in each structure which says what element comes next.

16. What are enumerations?

Ans: They are a list of named integer-valued constants. Example:enum color { black , orange=4, yellow, green, blue, violet };This declaration defines the symbols black, orange,yellow,etc. to have the values 1, 4, 5, … etc. The difference between an enumeration and a macrois that the enum actually declares a type, and therefore can be type checked.

17. Describe about storage allocation and scope of global, extern, static, local and register variables?

Ans:Globals have application-scope. They‘re available in any compilation unit that includes an appropriate declaration (usually brought from a header file). They‘re stored wherever the linker

puts them, usually a place called the BSS segment.

Extern:It is essentially global.

Static: Stored the same place as globals, typically, but only available to the compilation unit that contains them. If they are block-scope global, only available within that block and its subblocks.

Local: Stored on the stack, typically. Only available in that block and its subblocks.

(Although pointers to locals can be passed to functions invoked from within a scope where that local is valid.)

Register: It is local. The only difference is thatthe C compiler will not let you take the address of something you‘ve declared as register.

18. What are register variables? What are the advantages of using register variables?

Ans: If a variable is declared with a register storage class,it is known as register variable.The register variable is stored in the cpu register instead of main memory.Frequently used variables are declared as register variable as it‘s access time is faster.

19. What is the use of typedef?

Ans: The typedef help in easier modification when the programs are ported to another machine.

A descriptive new name given to the existing data type may be easier to understand the code.

20. Can we specify variable field width in a scanf() format string? If possible how?

Ans: All field widths are variable with scanf(). You can specify a maximum field width for a

Givenfield by placing an integer value between the _%‘ and the field type specifier. (e.g. %64s). Such aspecifier will still accept a narrower field width.The one exception is %#c (where # is an integer). This reads EXACTLY # characters, and it istheonly way to specify a fixed field width with scanf().

21. Out of fgets() and gets() which function is safe to use and why?

Ans: fgets() is safer than gets(), because we can specify a maximum input length. Neither one is completely safe, because the compiler can‘t prove that programmer won‘t overflow the buffer he pass to fgets ().

22. Difference between strdup and strcpy?

Ans: Both copy a string. strcpy wants a buffer to copy into. strdup allocates a buffer using malloc().Unlike strcpy(), strdup() is not specified by ANSI .

23. What is recursion?

Ans: A recursion function is one which calls itself either directly or indirectly it must halt at a definite point to avoid infinite recursion.

24. Differentiate between for loop and a while loop? What are it uses?

Ans: For executing a set of statements fixed number of times we use for loop while when the number ofiterations to be performed is not known in advance we use while loop.

25. What is storage class? What are the different storage classes in C?

Ans: Storage class is an attribute that changes the behavior of a variable. It controls the lifetime, scope and linkage. The storage classes in c are auto, register, and extern, static, typedef.

26. What the advantages of using Unions?

Ans: When the C compiler is allocating memory for unions it will always reserve enough room for thelargest member.

27. What is the difference between Strings and Arrays?

Ans: String is a sequence of characters ending with NULL .it can be treated as a one dimensional Arrayof characters terminated by a NULL character.

28. What is a far pointer? Where we use it?

Ans: In large data model (compact, large, huge) the address B0008000 is acceptable because in Thesemodel all pointers to data are 32bits long. If we use small data model(tiny, small, medium) theabove address won‘t work since in these model each pointer is 16bits long. If we are working ina small data model and want to access the address B0008000 then we use far pointer. Far pointeris always treated as a 32bit pointer and contains a segment address and offset address both of16bits each. Thus the address is represented using segment : offset format B000h:8000h. For anygiven memory address there are many possible far address segment : offset pair. The segmentregister contains the address where the segment begins and offset register contains the offset ofdata/code from where segment begins.

29. What is a huge pointer?

Ans: Huge pointer is 32bit long containing segment address and offset address. Huge pointers arenormalized pointers so for any given memory address there is only one possible huge address segment: offset pair. Huge pointer arithmetic is doe with calls to special subroutines so its arithmetic slower than any other pointers.

30. What is a normalized pointer, how do we normalize a pointer?

Ans: It is a 32bit pointer, which has as much of its value in the segment register as possible. Sincea segment can start every 16bytes so the offset will have a value from 0 to F. for normalizationconvert the address into 20bit address then use the 16bit for segment address and 4bit for theoffset address. Given a pointer 500D: 9407,we convert it to a 20bitabsolute address 549D7,Which then normalized to 549D:0007.

31. What is near pointer?

Ans: A near pointer is 16 bits long. It uses the current content of the CS (code segment) register (ifthe pointer is pointing to code) or current contents of DS (data segment) register (if the pointer ispointing to data) for the segment part, the offset part is stored in a 16 bit near pointer. Using nearpointer limits the data/code to 64kb segment.

32. In C, why is the void pointer useful? When would you use it?

Ans: The void pointer is useful because it is a generic pointer that any pointer can be cast into andback again without loss of information.

33. What is a NULL Pointer? Whether it is same as an uninitialized pointer?

Ans: Null pointer is a pointer which points to nothing but uninitialized pointer may point to anywhere.

34. Are pointers integer?

Ans: No, pointers are not integers. A pointer is an address. It is a positive number.

35. What does the error 'Null Pointer Assignment' means and what causes this error?

Ans: As null pointer points to nothing so accessing a uninitialized pointer or invalid location may cause an error.

36. What is generic pointer in C?

Ans: In C void* acts as a generic pointer. When other pointer types are assigned to generic pointer,conversions are applied automatically (implicit conversion).

37. Are the expressions arr and &arr same for an array of integers?

Ans: Yes for array of integers they are same.

38. IMP>How pointer variables are initialized?

Ans: Pointer variables are initialized by one of the following ways.

I. Static memory allocation

II. Dynamic memory allocation

39. What is static memory allocation?

Ans: Compiler allocates memory space for a declared variable. By using the address of operator, Thereserved address is obtained and this address is assigned to a pointer variable. This way of assigning pointer value to a pointer variable at compilation time is known as static memory allocation.

40. What is dynamic memory allocation?

Ans: A dynamic memory allocation uses functions such as malloc() or calloc() to get memory dynamically. If these functions are used to get memory dynamically and the values returned by these function are assigned to pointer variables, such a way of allocating memory at run time is known as dynamic memory allocation.

41. What is the purpose of realloc?

Ans: It increases or decreases the size of dynamically allocated array. The function realloc (ptr,n) uses two arguments. The first argument ptr is a pointer to a block of memory for which the size is to be altered. The second argument specifies the new size. The size may be increased or decreased. If sufficient space is not available to the old region the function may create a new region.

42. What is pointer to a pointer?

Ans: If a pointer variable points another pointer value. Such a situation is known as a pointer to a pointer.

Example:

int *p1,**p2,v=10;

P1=&v; p2=&p1;

Here p2 is a pointer to a pointer.

43. What is an array of pointers?

Ans: if the elements of an array are addresses, such an array is called an array of pointers.

44. Difference between linker and linkage?

Ans: Linker converts an object code into an executable code by linking together the necessary built infunctions. The form and place of declaration where the variable is declared in a program determine the linkage of variable.

45. Is it possible to have negative index in an array?

Ans: Yes it is possible to index with negative value provided there are data stored in this location. Even if it is illegal to refer to the elements that are out of array bounds, the compiler will not produce error because C has no check on the bounds of an array.

46. Why is it necessary to give the size of an array in an array declaration?

Ans: When an array is declared, the compiler allocates a base address and reserves enough space Inmemory for all the elements of the array. The size is required to allocate the required space andhence size must be mentioned.

47. What modular programming?

Ans: If a program is large, it is subdivided into a number of smaller programs that are called modules or subprograms. If a complex problem is solved using more modules, this approach is known as modular programming.

UNIT-III

48. What is a function?

Ans: A large program is subdivided into a number of smaller programs or subprograms. Each Subprogramspecifies one or more actions to be performed for the larger program. Such sub programs arecalled functions.

49. What is an argument?

Ans: An argument is an entity used to pass data from the calling to a called function.

50. What are built in functions?

Ans: The functions that are predefined and supplied along with the compiler are known as Builtinfunctions. They are also known as library functions.

51. Difference between formal argument and actual argument?

Ans: Formal arguments are the arguments available in the function definition. They are preceded Bytheir own data type. Actual arguments are available in the function call. These arguments are Givenas constants or variables or expressions to pass the values to the function.

52. Is it possible to have more than one main() function in a C program ?

Ans: The function main() can appear only once. The program execution starts from main.

53. What is the difference between an enumeration and a set of pre-processor # defines?

Ans: There is hardly any difference between the two, except that #defines has a global effect (throughout the file) whereas an enumeration can have an effect local to the block if desired. Some advantages of enumeration are that the numeric values are automatically assigned whereas in #define we have to explicitly define them. A disadvantage is that we have no control over the size of enumeration variables.

54. How are Structure passing and returning implemented by the complier?

Ans: When structures are passed as argument to functions, the entire structure is typically

pushed onthe stack. To avoid this overhead many programmer often prefer to pass pointers to structureinstead of actual structures. Structures are often returned from functions in a location pointed toby an extra, compiler-supported _hidden‘ argument to the function.

55. IMP>what is the similarity between a Structure, Union and enumeration?

Ans: All of them let the programmer to define new data type.

56. Can a Structure contain a Pointer to itself?

Ans: Yes such structures are called self-referential structures.

57. How can we read/write Structures from/to data files?

Ans: To write out a structure we can use fwrite() as Fwrite(&e, sizeof(e),1,fp);Where e is a

Structurevariable. A corresponding fread() invocation can read the structure back from file. Callingfwrite() it writes out sizeof(e) bytes from the address &e. Data files written as memory imageswith fwrite(),however ,will not be portable, particularly if they contain floating point fields orPointers. This is because memory layout of structures is machine and compiler

dependent. Therefore, structures written as memory images cannot necessarily be read back by

programs running on other machine, and this is the important concern if the data files you‘re

writing will ever be interchanged between machines.

58. Write a program which employs Recursion?

Ans: int fact(int n) { return n > 1 ? n * fact(n – 1) : 1; }

59.Write a program which uses Command Line Arguments?

Ans:

#include

void main(int argc,char *argv[])

{

int i;

clrscr();

for(i=0;i

printf(—\n%d‖,argv[i]);

}

UNIT-IV

60. Difference between array and pointer?

Ans:

Array

1- Array allocates space automatically

2- It cannot be resized

3- It cannot be reassigned

4- sizeof (arrayname) gives the number of bytes occupied by the array.

Pointer

1-Explicitly assigned to point to an allocated space.

2-It can be sized using realloc()

3-pointer can be reassigned.

4-sizeof (p) returns the number of bytes used to store the pointer variable p.

61. What do the ‘c’ and ‘v’ in argc and argv stand for?

Ans: The c in argc(argument count) stands for the number of command line argument the program isinvoked with and v in argv(argument vector) is a pointer to an array of character string thatcontain the arguments.

62. IMP>what are C tokens?

Ans: There are six classes of tokens: identifier, keywords, constants, string literals, operators and other separators.

63. What are C identifiers?

Ans: These are names given to various programming element such as variables, function, arrays.It is a combination of letter, digit and underscore.It should begin with letter. Backspace is not allowed.

64. Difference between syntax vs logical error?

Ans:

Syntax Error

1-These involves validation of syntax of language.

2-compiler prints diagnostic message.

Logical Error

1-logical error are caused by an incorrect algorithm or by a statement mistyped in such a way that it doesn‘t violet syntax of language.

2-difficult to find.

65. What is preincrement and post increment?

Ans: ++n (pre increment) increments n before its value is used in an assignment operation or any expression containing it. n++ (post increment) does increment after the value of n is used.

66. Write a program to interchange 2 variables without using the third one.

Ans:

a ^= b; ie a=a^b

b ^= a; ie b=b^a;

a ^= b ie a=a^b;

here the numbers are converted into binary and then xor operation is performed.

67. What is the maximum combined length of command line arguments including the space between adjacent arguments?

Ans: It depends on the operating system.

68. What are bit fields? What is the use of bit fields in a Structure declaration?

Ans: A bit field is a set of adjacent bits within a single implementation based storage unit that we will call a word.The syntax of field definition and access is based on structure.

Struct {

unsigned int k :1;

unsigned int l :1;

unsigned int m :1;

}flags;

the number following the colon represents the field width in bits.Flag is a variable that contains three bit fields.

69. What is a preprocessor, what are the advantages of preprocessor?

Ans: A preprocessor processes the source code program before it passes through the compiler.

1- a preprocessor involves the readability of program

2- It facilitates easier modification

3- It helps in writing portable programs

4- It enables easier debugging

5- It enables testing a part of program

6- It helps in developing generalized program

70. What are the facilities provided by preprocessor?

Ans:

1-file inclusion

2-substitution facility

3-conditional compilation

71. What are the two forms of #include directive?

Ans:

1.#include<filename>

2.#include

the first form is used to search the directory that contains the source file.If the search fails in the home directory it searches the implementation defined locations.In the second form ,the

preprocessor searches the file only in the implementation defined locations.

72. How would you use the functions randomize() and random()?

Ans:

Randomize() initiates random number generation with a random value.

Random() generates random number between 0 and n-1;

73. What do the functions atoi(), itoa() and gcvt() do?

Ans:

atoi() is a macro that converts integer to character.

itoa() It converts an integer to string

gcvt() It converts a floating point number to string

UNIT-V

74. How would you use the functions fseek(), freed(), fwrite() and ftell()?

Ans:

fseek(f,1,i) Move the pointer for file f a distance 1 byte from location i.

fread(s,i1,i2,f) Enter i2 dataitems,each of size i1 bytes,from file f to string s.

fwrite(s,i1,i2,f) send i2 data items,each of size i1 bytes from string s to file f.

ftell(f) Return the current pointer position within file f.

The data type returned for functions fread,fseek and fwrite is int and ftell is long int.

75. What is the difference between the functions memmove() and memcpy()?

Ans: The arguments of memmove() can overlap in memory. The arguments of memcpy() cannot.

76. What is a file?

Ans: A file is a region of storage in hard disks or in auxiliary storage devices.It contains bytes of information .It is not a data type.

77. IMP>what are the types of file?

Ans: Files are of two types

1-high level files (stream oriented files) :These files are accessed using library functions

2-low level files(system oriented files) :These files are accessed using system calls

78. IMP>what is a stream?

Ans: A stream is a source of data or destination of data that may be associated with a disk or other

I/O device. The source stream provides data to a program and it is known as input stream. The destination stream eceives the output from the program and is known as output stream.

79. What is meant by file opening?

Ans: The action of connecting a program to a file is called opening of a file. This requires Creatingan I/O stream before reading or writing the data.

80. What is FILE?

Ans: FILE is a predefined data type. It is defined in stdio.h file.

81. What is a file pointer?

Ans: The pointer to a FILE data type is called as a stream pointer or a file pointer. A file pointer points to the block of information of the stream that had just been opened.

82. How is fopen()used ?

Ans: The function fopen() returns a file pointer. Hence a file pointer is declared and it is Assignedas

FILE *fp;

fp= fopen(filename,mode);

filename is a string representing the name of the file and the mode represents:

—r for read operation

—w for write operation

—a for append operation

—r+,w+,a+ for update operation

83.How is a file closed ?

Ans: A file is closed using fclose() function

Eg. fclose(fp);

Where fp is a file pointer.

84. What is a random access file?

Ans:A file can be accessed at random using fseek() function

fseek(fp,position,origin);

fp file pointerposition number of bytes offset from origin

origin 0,1 or 2 denote the beginning ,current position or end of file respectively.

85. What is the purpose of ftell ?

Ans: The function ftell() is used to get the current file represented by the file pointer.

ftell(fp);

returns a long integer value representing the current file position of the file pointed by the

file pointer fp.If an error occurs ,-1 is returned.

86. What is the purpose of rewind() ?

Ans: The function rewind is used to bring the file pointer to the beginning of the file.

Rewind(fp);

Where fp is a file pointer.Also we can get the same effect by

feek(fp,0,0);

87. Difference between a array name and a pointer variable?

Ans: A pointer variable is a variable where as an array name is a fixed address and is not a

variable. Apointer variable must be initialized but an array name cannot be initialized. An array name beinga constant value , ++ and — operators cannot be applied to it.

88. Represent a two-dimensional array using pointer?

Ans:

Address of a[I][j] Value of a[I][j]

&a[I][j]

or

a[I] + j

or

*(a+I) + j

*&a[I][j] or a[I][j]

or

*(a[I] + j)

or

*(* (a+I) +j)

89. Difference between an array of pointers and a pointer to an array?

Ans:

Array of pointers

1- Declaration is: data_type *array_name[size];

2-Size represents the row size.

3- The space for columns may be dynamically

Pointers to an array

1-Declaration is data_type (*array_name)[size];

2-Size represents the column size.

90. Can we use any name in place of argv and argc as command line arguments ?

Ans: yes we can use any user defined name in place of argc and argv;

91. What are the pointer declarations used in C?

Ans:

1- Array of pointers, e.g , int *a[10]; Array of pointers to integer

2-Pointers to an array,e.g , int (*a)[10]; Pointer to an array of into

3-Function returning a pointer,e.g, float *f( ) ; Function returning a pointer to float

4-Pointer to a pointer ,e.g, int **x; Pointer to apointer to int

5-pointer to a data type ,e.g, char *p; pointer to char

92. Differentiate between a constant pointer and pointer to a constant?

Ans:

const char *p; //pointer to a const character.

char const *p; //pointer to a const character.

char * const p; //const pointer to a char variable.

const char * const p; // const pointer to a const character.

93. Is the allocated space within a function automatically deallocated when the function returns?

Ans: No pointer is different from what it points to .Local variables including local pointers variables in a function are deallocated automatically when function returns.,But in case of a local pointer variable ,deallocation means that the pointer is deallocated and not the block of memory allocated to it. Memory dynamically allocated always persists until the allocation is freedor the program terminates.

94. Discuss on pointer arithmetic?

Ans:

1- Assignment of pointers to the same type of pointers.

2- Adding or subtracting a pointer and an integer.

3-subtracting or comparing two pointer.

4-incrementing or decrementing the pointers pointing to the elements of an array. When a pointer to an integer is incremented by one , the address is incremented by two. It is done automatically

by the compiler.

5-Assigning the value 0 to the pointer variable and comparing 0 with the pointer. The pointer having address 0 points to nowhere at all.

95. What is the invalid pointer arithmetic?

Ans:

i) adding ,multiplying and dividing two pointers.

ii) Shifting or masking pointer.

iii) Addition of float or double to pointer.

iv) Assignment of a pointer of one type to a pointer of another type ?

96. What are the advantages of using array of pointers to string instead of an array of strings?

Ans:

i) Efficient use of memory.

ii) Easier to exchange the strings by moving their pointers while sorting.

97. Are the expressions *ptr ++ and ++ *ptr same?

Ans: No,*ptr ++ increments pointer and not the value pointed by it. Whereas ++ *ptr increments the value being pointed to by ptr.

98. What would be the equivalent pointer expression foe referring the same element as a[p][q][r][s] ?

Ans : *(* (* (* (a+p) + q) + r) + s)

99. Are the variables argc and argv are always local to main?

Ans: Yes they are local to main.

100. Can main () be called recursively?

Ans: Yes any function including main () can be called recursively.

101. IMP>Can we initialize unions?

Ans: ANSI Standard C allows an initializer for the first member of a union. There is no standard Wayof initializing any other member (nor, under a pre-ANSI compiler, is there generally any way ofinitializing a union at all).

102. What's the difference between these two declarations?

Ans: struct x1 { … };

typedef struct { … } x2;

The first form declares a structure tag; the second declares a typedef. The main difference is that the second declaration is of a slightly more abstract type.its users don't necessarily know that it is a structure, and the keyword struct is not used when declaring instances of it.

103. Why doesn't this code: a[i] = i++; work?

Ans: The sub expression i++ causes a side effect. it modifies i's value. which leads to undefined behavior since i is also referenced elsewhere in the same expression.

104. Why can't we compare structures?

Ans:

There is no single, good way for a compiler to implement structure comparison which is consistent with C's low-level flavor. A simple byte-by-byte comparison could founder on random bits present in unused ―holes‖ in the structure (such padding is used to keep the alignment of later fields correct). A field-by-field comparison might require unacceptable amounts of repetitive code for large structures.

105. How are structure passing and returning implemented?

Ans: When structures are passed as arguments to functions, the entire structure is typically pushed on

the stack, using as many words as are required. Some compilers merely pass a pointer to the structure, though they may have to make a local copy to preserve pass-by-value semantics. Structures are often returned from functions in a location pointed to by an extra,compilersupplied

Hidden argument to the function. Some older compilers used a special,static location for structure returns, although this made structure-valued functions non-reentrant, which ANSI C disallows.

****

Department of Computer Science(S.F)

I-B.Sc.Computer Science – Question Bank – K3 level

18UCS101 – PROGRAMMING IN C

Unit- I

1.Examine the concept of Algorithms.

2.Categorize the structure of Flowchart .

3.List out the symbols used in flowchart.

4.Elucidate the history of computers.

5.List out the generations of computers.

6.List out the classification of computers.

7.List out the anatomy of basic computers.

8. Discuss about Constants.

9.Discuss about C character set.

10.Analyse about c-tokens.

11.List out the rules for defining variables.

12.Elucidate Data types.

13.Discuss about type conversions in c.

14.Distinguish between Input devices and outputdevices.

15.Conclude about precedence of arithmetic operators.

16.Elucidate in brief about the operators precedence and its associativity.

17.List out the mathematical functions in c.

18.Describe about the input/output statements with examples.

Unit-II

1.List out the control statements with examples.

2.Mention the differences between while loop and do..while loop

3.Examine the features of for statements.

4.Mention some of the user defined data types.

5.List out types of arrays.

6.Analyse arrays within structures .

7.Conclude about structures within structures.

8.Discover structures and functions with examples.

9.Simplify initialization and declaration variables.

10.Examine the concept about reading and writing strings from and to terminals.

11.List out string handling functions.

Unit-III

1.Examine the elements of user-defined functions.

2.Conclude the features of user-defined functions.

3.Analyse the concept of function calls and declarations.

4.Analyse the concept of nesting of functions.

5.Examine the concept of recursion.

6.Conclude passing arrays to functions.

7.Examine the features of passing arrays to strings.

8.Inspect the scope ,visibility and lifetime of variables in functions.

9.Write a program to swap two numbers using functions.

Unit-IV

1.Examine the features of pointers.

2.Examine the features of declaring and initializing pointer variables.

3.Conclude chain of pointers.

4.Conclude about pointer expressions.

5.Distinguish between pointer and arrays.

6.Analyse about array of pointers.

7.Compare and contrast between pointer to arrays and pointer to dfunctions.

8.Inspect about pointers and structures.

9. Elucidate about troubles with pointers

10.Write a program to add two numbers using pointer.

11.Write a program to transpose of a matrix using pointer.

Unit-V

1.List out the types of File operations.

2.List out the commonly used header files used in c.

3.Analyse Opening and closing a file with a neat sketch.

4.List out the types of File pointers .

5.Discuss about random access files.

6.Conclude about command line arguments.

7.Describe about compiler control directives.

8.Explain about macro substitution.

9.Write a program for binary file.

10.Write a program for ASCII file .

11.Write a program for Sequential file.

****

Department of Computer Science(S.F)

I-B.Sc.Computer Science – Question Bank – K4 & K5 level

18UCS101 – PROGRAMMING IN C

Unit- I

1.Examine the concept of Algorithms and Flowchart.

2.List out the symbols used in flowchart with an example.

3.Explain the history of computers.

4.Explain the generations of computers with neat sketch.

5.List out the classification of computers with neat sketch.

6.Explain the anatomy of basic computers and its components.

7.Discuss about C character set and tokens.

8.Explain about Data types.

9.Discuss about type conversions in c.

10.Explain about precedence of arithmetic operators.

11.Elucidate in brief about the operators precedence and its associativity.

12.Discuss about the mathematical functions in c.

18.Describe about the input/output statements with examples.

Unit-II

1.List out the control statements with examples.

2.Mention the differences between the following with suitable examples

(i) while loop and do..while loop

3.Explain the features of switch statements.

4.Explain about user defined data types.

5.Explain in brief arrays within structures .

7.Elaborate about structures within structures.

8.Discuss about structures and functions with examples.

9.List out string handling functions.

10.Write a program to reverse a string using string functions.

Unit-III

1.Examine the elements of user-defined functions.

2.Explain the concept of recursion.

3.Explain the scope ,visibility and lifetime of variables in functions.

4.Write a program to find the factorial of a number using recursion.

Unit-IV

1.Examine the features of pointers.

2.Distinguish between pointer and arrays.

3.Explain about array of pointers.

7.Compare and contrast between pointer to arrays and pointer to functions.

8.Explain about pointers and structures.

9. List out the troubles faced with pointers

10.Write a program to perform matrix multiplication using pointer.

11.Write a program to transpose of a matrix using pointer.

Unit-V

1.List out the types of File operations.

2.List out the commonly used header files used in c.

3.Explain about File pointers .

4.Discuss about random access files.

5.Explain about command line arguments.

6.Describe about compiler control directives.

7.Write a program for binary file.

8.Write a program for ASCII file .

9.Write a program for Sequential file.

****