

Model Viva Questions for “Programming in C lab”

Common to: CSE 2nd sem IT 2nd sem

Title of the Practical: Assignment to prepare general algorithms and flow chart.

Q1: What is a flowchart?

A1: A flowchart is a diagram that shows a continuous flow of materials or steps to (WIP), work in process. Basically it shows the process, steps or task for something to be completed from beginning to end

Q2: What is an Algorithm?

A2: An algorithm is an effective method for solving a problem expressed as a finite sequence of steps. Algorithms are used for calculation, data processing, and many other fields.

Q3: What is a Brute force algorithm?

A3: Brute Force Algorithms: The brute force approach starts at some random point and iterates through every possibility until it finds the solution.

Q4: What is Simple Recursive Algorithms

A4: Simple Recursive Algorithms: This type of algorithm goes for a direct solution immediately, then backtracks to find a simpler solution.

Q5: What is Backtracking Algorithms?

A5: Backtracking algorithms test for a solution, if one is found the algorithm has solved, if not it recurs once and tests again, continuing until a solution is found.

Q6: What is Divide and Conquer Algorithms:

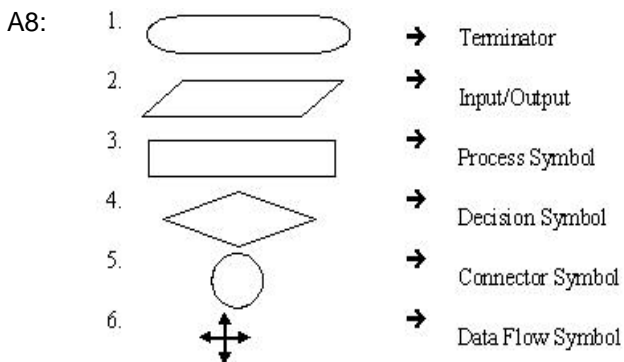
A6: A divide and conquer algorithm is similar to a branch and bound algorithm, except it uses the backtracking method of recurring in tandem with dividing a problem into sub problems.

Q7: What are advantages of flow charts?

A7: Advantages of flowchart:-

1. It provides an easy way of communication because any other person besides the programmer can understand the way they are represented. 2. It represents the data flow. 3. It provides a clear overview of the entire program and problem and solution. 4. It checks the accuracy in logic flow.

Q8: What are different symbols in flowcharts?



Q9: What is a connector symbol?

A9: Connector Symbol: This symbol is used to connect the various portion of a flow chart. This is normally used when the flow chart is split between two pages.

Q10: What is a data flow symbol?

A10: Data Flow Symbol : This symbol is used to display the flow of the program. It shows the path of logic flow in a program.

Title of the Practical: Assignment to write character, operators' symbols of C Language

Q1: What are the three primary constants and variable types in C?

A1: integer, float, character.

Q2: What is the maximum length of variable?

A2: 31 characters.

Q3: Can a C keyword be used as a variable name?

A3: No.

A4: What does an expression contain?

A4: An expression may contain any sequence of constants, variables, and operators.

Q5: How is an expression evaluated?

A5: An expression is evaluated based on the hierarchy and operator precedence.

Q6: What does the modulus operator perform?

A6: Modulus operator performs division and gives the remainder.

Q7: Name different logical operators.

A7: Logical OR, logical AND, Logical NOT.

Q8: What is the syntax of the conditional operator?

A8: expression1? Expression 2: expression 3.

Q9: What is an expression?

A9: An expression is any valid combination of tokens.

Q10: What is a type conversion?

A10: The process of converting one predefined type into another is called type conversion.

Title of the Practical: Assignment to identify valid and invalid variables, constants and expressions

Q1: What is a variable?

A1: Variable represents a named memory location whose values can be manipulated during program run.

Q2: How a variable is declared?

A2: type name:

Q3: What is a constant?

A3: A constant is a data item whose data value can never change during the program run.

Q4: List some valid logical expression.

A4: $x > y$, $(y = z) >= (x / z)$, $x || y \& \& z$.

Q5: What is a keyword?

A5: Keyword is a word carrying special meaning and purpose.

Q6: What are Literals?

A6: Literals are data items that never change their values during a program run.

Q7: What are two categories of C constants?

A7: (a) Primary constants (b) Secondary constant

Q8: What is the range of integer constant?

A8: The range of integer constant depends upon the compiler.

Q9: What is the range of integer constant for compilers like Turbo C?

A9: Range: -32768 to 32767.

Q10: What is the maximum length of character constants?

A10: The maximum range of character constant can be 1 character.

Title of the Practical: Program based on Input/Output statements.

Q1: What is the output of printf("%d") ?

A1: When we write printf("%d",x); this means compiler will print the value of x. But as here, there is nothing after %d so compiler will show in output window garbage value.

Q2: What is the difference between printf() and sprintf() ?

A2 : sprintf() writes data to the character array whereas printf(...) writes data to the standard output device.

Q3: What three primary constants and variable type in c?

A3: integer float and character.

Q4: What is a variable?

A4: A variable is a named storage location. It has two associated values : rvalue that gives its contents and lvalue that gives its address.

Q5: What is the maximum length of a variable name?

A5: A variable name can be maximum 31 characters.

Q6: How input and output in c can be achieved??

A6: Using scanf() and printf().

Q7: What is the function of prefix version of increment and decrement?

A7: The prefix first changes (increments or decrements) its operand's value and then uses it i.e. , it follows change-then-use rule.

Q8: How operators with equal precedence are evaluated ?

A8: Using associativity.

Q9: What type of expression is arithmetic expression?

A9: Arithmetic expression can either be integer expression or real expressions or mixed mode expressions.

Q10: What we do in implicit conversion ?

A10: In implicit conversion, all operands are converted upto the type of the largest operand , which is called type promotion.

Title of the Practical: Program based on Arithmetic expression

Q1: What type of expression is arithmetic expression?

A1: Arithmetic expression can either be integer expression or real expressions or mixed mode expressions.

Q2: Name the four basic data types in "C" language?

A2: The four basic data types in "c" language are as follows

- (i) Char
- (ii) Int
- (iii) Float
- (iv) Double

Q3: Describe at least five different format specifiers?

A3: %d: -An integer whole number

%f: -a floating point number

%c: -a single character

%p: -a machine memory address

Q4: Define and explain scanf () function?

A4: The scanf () function can be used to get input into a program and it requires two arguments. First a format specifier defines the type of data to be entered, then the name of the variable in which the input will be stored. This scanf () function is responsible for giving input into the program.

Q5: What are the maximum and minimum possible ranges of values for long and short type?

A5: If the int variable is created by default as a 'long' type it typically will have a possible range of values from a maximum of +2147483647 and a minimum of -2147483648. For 'short' type these are the maximum and minimum values +32767 and minimum -32768.

(While answering this question you can specify the approximate value raised to power).

Q6: What exactly is a 'variable scope', 'local variables' and 'global variables'?

A6: The extent to which a variable is accessible in a program is called the 'variable scope'.

Variables declared internally inside a function are known as 'local' variables.

Variables declared externally outside a function are known as 'global' variables.

Q7: What are signed values?

A7: When an int variable is declared it can by default contain either positive or negative integer values. These are known as 'signed' values. The range of positive values is determined by your system.

Q8: Explain about register variables relative to compiler?

A8: A register variable declaration informs the compiler that the specified variable will be heavily used by the program.

The intention is for the compiler to place register variables in the machine register of the computer to speed access times.

Local internal variables can only be declared as register variables.

Q9: State some significant uses of arrays in C programming?

A9: One of the most significant uses of arrays in c programming concerns with their ability to store strings of text. Each element in an array of the char data type can store a single character. Adding the special null character escape sequence in the arrays final element promotes the array to string. It can be referenced using the array name and can be displayed using the %s format specifier.

Q10: What is enum used for and state its format?

A10: The enum keyword provides a handy way to create a sequence of integer constants in a concise manner. Optionally the declaration can include a name for the sequence after the enum keyword. The constant names follow with a comma and placed within braces. The name assigned can be used to call the function again at later stage. The constants can be assigned any individual value but the following constant will always increment it by one.

Title of the Practical : Program based on Library functions.

Q1: Explain about the constants which help in debugging?

A1: A #if directive test can be offered with #else and #else if directives. This allows conditional branching of the program to run sections of the code according to the result. Constants defined with a #define directive can be undefined with the #undef directive. The #ifdef directive has a companion directive #ifndef. These commands can be useful when debugging problem code to hide and unhide sections of the program.

Q2: Define and explain about! Operator?

A2: The logical operator! NOT is a unary operator that is used before a single operand. It returns the inverse value of the given operand so if the variable "c" had a value of true then! C would return value of false. The not operator is very much useful in C programs because it can change the value of variables with successful iterations. This ensures that on each pass the value is changed.

Q3: What is operator precedence?

A3: Operator precedence defines the order in which C evaluates expressions.

e.g. in the expression $a=6+b*3$, the order of precedence determines whether the addition or the multiplication is completed first. Operators on the same row have equal precedence.

Q4: Explain about the functions strcat() and strcmp()?

A4: This function concatenates the source string at the end of the target string. Strcmp() function compares two strings to find out whether they are the same or different. The two strings are compared character by character until there is a mismatch or end of one of the strings is reached, whichever occurs first. If in case two strings are identical, a value of zero is returned. If there is no matches between two strings then a difference of the two non matching values are returned according to ASCII values.

Q5: Why do we use structures?

A5: We use structures for different purposes which are

1) Construct individual arrays

E.g. 1) one array for storing names, another array for storing prices, and another for storing number of pages.

2) Use a structure variable.

A structure contains a number of data types grouped together. These data types may or may not be of the same type.

Q6: State the difference between sprintf () and sscanf() Functions?

A6: The sprintf() function works similar to the printf() function except for one small difference, instead of sending the output to the screen as printf() does, this function writes the output to an array of characters. Once str has been built, its contents can be displayed on the screen.

Q8: How do you write data to low level disk I/O?

A8: In low level disk I/O, data cannot be written as individual characters, or as strings or as formatted data. There is only one way data can be written to read in low level disk I/O functions as buffer full of bytes. Writing a buffer full of data resembles the fwrite() function.

Q9: Define these functions fopen(), fread(), fwrite() and fseek()?

A9: Fopen(): - On opening a file, a pointer is set up which points to the first record in the file.

Fread(): - The pointer moves to the next record while reading a file. It always reads the file where the pointer is pointed.

Fwrite(): -The pointer moves to the beginning of the next record on closing the file the pointer is deactivated. This pointer writes the record where the pointer is currently places.

Fseek(): - Function lets us move the pointer from one record to another.

Q10: State the difference between Unions and structures?

A10: Both structures and unions are used to group a number of different variables together. But while a structure enables us treat a number of different variables stored at different places in memory, a union enables us to treat the same space in memory as a number of different variables.

Title of the Practical: A Program based on go to statement

Q1: What is a goto statement ?

A1: The goto statement is a jump statement which jumps from one point to another point within a function.

Q2: What is the difference between goto and longjmp() and setjmp()?

A2: A goto statement implements a local jump of program execution, and the longjmp() and setjmp() functions implement a nonlocal, or far, jump of program execution.

Q3: What is the problem with goto statement?

A3: The big problem with goto statement is that when we do use them we can never sure how we got to a certain point in our code.

Q4: Why the use of goto keyword must be avoided?

A4: Because it usually violets the flow of execution.

Q5: Define structural language and procedural language?

A5: Structural language is one which can have a specific structure. C is a structured language before structural language procedural language was used.

Q6: What is Preprocessor?

A6: preprocessor is a program which is invoked before the compilation process. it access input data and process output data which has been used as input data to the other program like compiler.

Q7: What is modular programming?

A7: A large program can be divided into small Modules or into small programs or functions To solve a particular problem all the Modules are combined together. This is called Modular Programming.

Q8: What is the code for clrscr() function?

A8: clrscr() is a function which creates a screenful of black dots which seems us to be clearing the screen or making a new screen.

Q9: What is the purpose of main() function?

A9: main() is the user defined function . the program execution is always started from main. it is not possible to write the program without main().

Q10: What is the function of break statement when used within switch?

A10: The break statement when used in aswitch takes the control outside the switch .

Title of the Practical: Two Programs based on 'if' and 'Nested if'.

Q1: What is the purpose of if statement?

A1: The keyword if tells the compiler that what follows is a decision control instruction.

Q2: How does if statement work?

A2: If the condition within the parenthesis is true then the statement is executed otherwise not.

Q3: How do we express a condition itself in C?

A3: We express a condition using C's relational operators.

Q4: What is equality operator?

A4: == is used to compare the equality of two statements.

Q5: What is the syntax of if statement?

A5: if(expression)
Statement;

Q6: What are the different ways of decision making in a program?

A6: There are the ways:- if- else , conditional operators, switch statement.

Q7: What is the default scope of if statement?

A7: Next statement.

Q8: What are && and || ?

A8: These two are binary operators.

Q9: What is important in if else block?

A9: An if block need not be always associated with an else block . However an else block must be associated with an if block.

Q10: Name a unary operator.

A10: ! (not equal to) is a unary operator.

Title of the Practical: Program based on 'switch case' statement.

Q1: What is a switch?

A1: The control statement that allows us to make a decision from the number of choices is called a switch .

Q2: What is case keyword?

A2: The case keyword is followed by an integer or a character constant.

Q3: What happens when we run a program containing a switch?

A3: First , the integer expression following the keyword switch is evaluated.

Q4: Is there any problem with the following : `char*a=NULL; char& p = *a;`?

A4: The result is undefined. You should never do this. A reference must always refer to some object.

Q5: Is switch a replacement of if? Yes and No.

A5: Yes, because it offers a better way of writing program as compared to if, and no, because in certain cases we are left with no choices.

Q6: What is the difference between `const char *myPointer` and `char *const myPointer`?

A6: `const char *myPointer` is a non constant pointer to constant data; while `char *const myPointer` is a constant pointer to non constant data.

Q7: From which is the pointer to object of a base class type compatible ?

A7: Pointers to object of a base class type is compatible with pointer to object of a derived class . Therefore , we can use single pointer variable to point to objects of base class as well as derived class.

Q8: What is a **this** pointer?

A8: A special pointer known as **this** pointer stores the address of the object that is currently invoking a member function.

Q9:Where is switch statement is useful?

A9: Switch statement is useful while writing menu driven programs..

Q10: Why is Arrow operator ("> ") used?

A10: The arrow operator is used to access the public members of the class with a pointer to an object.

Title of the Practical: . At least one program based on each

(a) For statement, (b) While statement , (c) Do-while statement, (d) Break statement

Q1. What is the difference between declaring a variable and defining a variable?

A1: Declaration of a variable in C hints the compiler about the type and size of the variable in compile time. Similarly, declaration of a function hints about type and size of function parameters. No space is reserved in memory for any variable in case of declaration.

Q2. What is a static variable?

A2: A static variable is a special variable that is stored in the data segment unlike the default automatic variable that is stored in stack. A static variable can be initialized by using keyword static before variable name.

Example: static int a = 5;

Q3. What is a register variable?

A3: Register variables are stored in the CPU registers. Its default value is a garbage value. Scope of a register variable is local to the block in which it is defined. Lifetime is till control remains within the block in which the register variable is defined.

Q4. Where is an auto variable stored?

A4: Main memory and CPU registers are the two memory locations where auto variables are stored. Auto variables are defined under automatic storage class. They are stored in main memory. Memory is allocated to an automatic variable when the block which contains it is called and it is de-allocated at the completion of its block execution.

Auto variables:Storage : main memory. **Default value :** garbage value. **Scope :** local to the block in which the variable is defined. **Lifetime :** till the control remains within the block in which the variable is defined.

Q5. What is scope & storage allocation of extern and global variables?

A5: **Extern variables:** belong to the External storage class and are stored in the main memory. extern is used when we have to refer a function or variable that is implemented in other file in the same project. The scope of the extern variables is Global. **Global variables:** are variables which are declared above the main() function. These variables are accessible throughout the program. They can be accessed by all the functions in the program. Their default value is zero.

Q6. What is scope & storage allocation of static and local variables?

A6. **Static variables:** Memory is allocated at the beginning of the program execution and it is reallocated only after the program terminates. The scope of the static variables is local to the block in which the variables are defined.

Local variables: are variables which are declared within any function or a block. They can be accessed only by function or block in which they are declared. Their default value is a garbage value.

Q7. What are storage memory, default value, scope and life of Automatic and Register storage class?

A7. **1. Automatic storage class:Storage :** main memory. **Default value :** garbage value. **Scope :** local to the block in which the variable is defined. **Lifetime :** till control remains within the block.

2. Register storage class:Storage : CPU registers. **Default value :** garbage value. **Scope :** local to the block in which the variable is defined. **Lifetime :** till control remains within the block.

Q8. What are storage memory, default value, scope and life of Static and External storage class?

A8. **Static storage class:Storage :** main memory. **Default value :** zero **Scope :** local to the block in which the variable is defined. **Lifetime :** till the value of the variable persists between different function calls.

2. External storage class:Storage : main memory **Default value :** zero **Scope :** global **Lifetime :** as long as the program execution doesn't come to an end.

Q9. What is the difference between 'break' and 'continue' statements?

A9. 1. break is a keyword used to terminate the loop or exit from the block. The control jumps to next statement after the loop or block. And continue is a keyword used for skipping the current iteration and go to next iteration of the loop

10. What is the difference between 'for' and 'while' loops?

A10. for loop: When it is desired to do initialization, condition check and increment/decrement in a single statement of an iterative loop, it is recommended to use 'for' loop. **Syntax:** for(initialization;condition;increment/decrement)

while loop: When it is not necessary to do initialization, condition check and increment/decrement in a single statement of an iterative loop, while loop could be used. In while loop statement, only condition statement is present.

Title of the Practical: Program based on one dimensional array and two dimensional arrays

Q1: What are arrays?

A1: An array is similar to an ordinary variable except that it can store multiple elements of similar type.

Q2: Does compiler perform bound checking on an arrays?

A2: No, It doesnot perform bound checking on arrays.

Q3: How are array elements stored?

A3: Array elements are stored in contiguous locations and so they can be accessed using pointers.

Q4: What type of arithmetic can be done on pointer?

A4: Only limited arithmetic can be done on pointers.

Q5: How is individual element in an array can be accessed?

A5: Individual element in an array can be accessed by using subscript, the number in the bracket following the array name.

Q6: How can we enter data on arrays?

A6: Data in an array element can be inserted by using loops.

Q7: What happens if the size of array exceeds?.

A7: Data entered with subscript exceeding the array size will simply be placed in memory outside the array.

Q8: How are array elements can be passed to function?

A8: By calling the function by value or by reference.

Q9: What is important is initalizing 2-d arrays ?

A9: It is necessary to mention the second (coulum) dimension,whereas the first dimension(row) is optional.

Q10: What is the syntax of array declaration?

A10: `data_type array _name[size of array];`

Title of the Practical: Programs based on string operations

Q1:What are strings?

A1: Character arrays are many atime called strings.

Q2: What is a string constant?

A2: A string constant is a one dimensional array of characters terminated by a null ('\0').

Q3: Why is null character important?

A3: It is the only way the function that work with a string can know where the string ends.

Q4: How is null character inserted ?

A4: C inserts null character automatically.

Q5:Explain the purpose of strlen().

A5: This function counts the number of characters present in a string.

Q6: What is strcat()?

A6: Appends one string at the end of another.

Q7: What is strcpy?

A7: Copies a string into another.

Q8: What is the use of strcmp?

A8: Compares two string.

Q9: How strcmp() works?

A9: It would return 0 if two string matches otherwise return a non-zero value.

Q10: Which functions can handle multi-word strings.

A10: printf() and puts().

Title of the Practical: Two programs based on functions

Q1: What is the purpose of main() function?

A1: In C, program execution starts from the main() function. Every C program must contain a main() function. The main function may contain any number of statements. These statements are executed sequentially in the order which they are written. The main function can in-turn call other functions. When main calls a function, it passes the execution control to that function. The function returns control to main when a return statement is executed or when end of function is reached.

Q2. Explain command line arguments of main function?

A2: In C, we can supply arguments to 'main' function. The arguments that we pass to main () at command prompt are called command line arguments. These arguments are supplied at the time of invoking the program.

The main () function can take arguments as: main(int argc, char *argv[]) { }

The first argument argc is known as 'argument counter'. It represents the number of arguments in the command line. The second argument argv is known as 'argument vector'. It is an array of char type pointers that points to the command line arguments. Size of this array will be equal to the value of argc.

Q3. What are header files? Are functions declared or defined in header files ?

A3: Functions and macros are declared in header files. Header files would be included in source files by the compiler at the time of compilation. Header files are included in source code using #include directive. #include <some.h> includes all the declarations present in the header file 'some.h'. A header file may contain declarations of sub-routines, functions, macros and also variables which we may want to use in our program. Header files help in reduction of repetitive code

Q4. What are the differences between formal arguments and actual arguments of a function?

A4: **Actual arguments:** The arguments that are passed in a function call are called actual arguments. These arguments are defined in the calling function.

Formal arguments: The formal arguments are the parameters/arguments in a function declaration. The scope of formal arguments is local to the function definition in which they are used. Formal arguments belong to the called function. Formal arguments are a copy of the actual arguments. A change in formal arguments would not be reflected in the actual arguments.

Q5. What is pass by value in functions?

A5: **Pass by Value:** In this method, the value of each of the actual arguments in the calling function is copied into corresponding formal arguments of the called function. In pass by value, the changes made to formal arguments in the called function have no effect on the values of actual arguments in the calling function.

Q6. What is pass by reference in functions?

A6: **Pass by Reference:** In this method, the addresses of actual arguments in the calling function are copied into formal arguments of the called function. This means that using these addresses, we would have an access to the actual arguments and hence we would be able to manipulate them. C does not support Call by reference. But it can be simulated using pointers

7. What are the differences between getchar() and scanf() functions for reading strings?

A7: **scanf();**-Entering of each character should be followed by return key. Continuous stream of characters cannot be directly supplied using scanf function.

getchar();- 1. Need not type return key. 2. Continuous stream of characters can be directly supplied using getchar function

Q8. Out of the functions fgets() and gets(), which one is safer to use and why?

A8: Out of functions fgets () and gets (), fgets () is safer to use. gets () receives a string from the keyboard and it is terminated only when the enter key is hit. There is no limit for the input string. The string can be too long and may lead to buffer overflow.

Q9. What is the difference between the functions strdup() and strcpy()?

A9: **strcpy function:** copies a source string to a destination defined by user. In strcpy function both source and destination strings are passed as arguments. User should make sure that destination has enough space to accommodate the string to be copied. 'strcpy' sounds like short form of "string copy".

strdup function: duplicates a string to a location that will be decided by the function itself. Function will copy the contents of string to certain memory location and returns the address to that location. 'strdup' sounds like short form of "string duplicate"

Q10: What is the purpose gets()?

A10: gets() receives a string from a keyboard.

