

EC8393-FUNDAMENTALS OF DATASTRUCTURES IN C

UNIT - I BASICS OF C PROGRAMMING

PART – A

1. Define compiler.

It is a program used to convert the high level language program into machine language.

2. What are the types of programming language? ·

Machine language · Assembly language · High level language

3. What are the different data types available in „C“?

There are four basic data types available in „C“. 1. int 2. float 3. char 4. Double

4. What are Keywords?

Keywords are certain reserved words that have standard and pre-defined meaning in „C“. These keywords can be used only for their intended purpose.

5. What is an Operator and Operand?

An operator is a symbol that specifies an operation to be performed on operands. **Example:** *, +, -, / are called arithmetic operators. The data items that operators act upon are called operands. **Example:** a+b; In this statement a and b are called operands.

6. What is Ternary operators or Conditional operators?

Ternary operators is a conditional operator with symbols ? and : **Syntax:** variable = exp1 ? exp2 : exp3.

If the exp1 is true variable takes value of exp2. If the exp2 is false, variable takes the value of exp3.

7. What are the Bitwise operators available in „C“?

& - Bitwise AND | - Bitwise OR ~ - One's Complement >> - Right shift << - Left shift ^ - Bitwise XOR are called bit field operators

8. What are the logical operators available in „C“?

The logical operators available in „C“ are && - Logical AND || - Logical OR ! - Logical NOT

9. What is the difference between Logical AND and Bitwise AND?

Logical AND (&&): Only used in conjunction with two expressions, to test more than one condition. If both the conditions are true the returns 1. If false then return 0. AND (&): Only used in Bitwise manipulation. It is a unary operator.

10. What is the difference between „=“ and „==“ operator?

Where = is an assignment operator and == is a relational operator. **Example:** while (i=5) is an infinite loop because it is a non zero value and while (i==5) is true only when i=5.

11. What is type casting?

Type casting is the process of converting the value of an expression to a particular data type. **Example:** `int x,y; c = (float) x/y;` where a and y are defined as integers. Then the result of x/y is converted into float.

12. What is the difference between „a“ and “a”?

„a“ is a character constant and “a” is a string.

13. What is the difference between while loop and do...while loop?

In the while loop the condition is first executed. If the condition is true then it executes the body of the loop. When the condition is false it comes out of the loop. In the do...while loop first the statement is executed and then the condition is checked. The do...while loop will execute at least one time even though the condition is false at the very first time.

14. What is a Modulo Operator?

“%” is modulo operator. It gives the remainder of an integer division

Example: `a=17, b=6. Then c=%b` gives 5.

15. How many bytes are occupied by the int, char, float, long int and double?

int - 4 Bytes char - 1 Byte float - 4 Bytes long int - 8 Bytes double - 8 Bytes

16. What are the types of I/O statements available in „C“?

There are two types of I/O statements available in „C“. Formatted I/O Statements Unformatted I/O Statements

17. What is the difference between ++a and a++?

++a means do the increment before the operation (pre increment) a++ means do the increment after the operation (post increment)

Example: `a=5; x=a++; /* assign x=5 */ y=a; /*now y assigns y=6 */ x=++a; /*assigns x=7 */`

18. What is a String?

String is an array of characters.

19. What is a global variable?

The global variable is a variable that is declared outside of all the functions. The global variable is stored in memory, the default value is zero. Scope of this variable is available in all the functions. Life as long as the program's execution doesn't come to an end.

20. What are the Escape Sequences present in „C“

\n - New Line \b - Backspace \t - Form feed \" - Single quote \\ - Backspace \t - Tab \r - Carriage return \a - Alert \" - Double quotes

21. Construct an infinite loop using while?

while (1) { } Here 1 is a non zero, value so the condition is always true. So it is an infinite loop.

22. Write the limitations of getchar() and scanf() functions for reading strings getchar()

To read a single character from stdin, then getchar() is the appropriate. **scanf()** scanf() allows to read more than just a single character at a time.

23. What is the difference between scanf() and gets() function?

In scanf() when there is a blank was typed, the scanf() assumes that it is an end. gets() assumes the enter key as end. That is gets() gets a new line (\n) terminated string of characters from the keyboard and replaces the „\n“ with „\0“.

24. What is meant by Control String in Input/Output Statements?

Control Statements contains the format code characters, specifies the type of data that the user accessed within the Input/Output statements.

25. What is the output of the programs given below?

```
main() { { float a; float a; int x=6, y=4; int x=6, y=4; a=x\y; a=(float) x\y; printf("Value of a=%f", a); printf("Value of a=%f",a); } }
```

Output 1. 1.500000

26. What is the output of the following program when, the name given with spaces?

```
main() { char name[50]; printf("\n name\n"); scanf("%s, name); printf("%s",name); }
```

Output: Lachi (It only accepts the data upto the spaces)

27. What is the difference between while(a) and while(!a)?

while(a) means while(a!=0) while(!a) means while(a==0)

28. Why we don't use the symbol „&“ symbol, while reading a String through scanf()?

The „&“ is not used in scanf() while reading string, because the character variable itself specifies as a base address.

Example: name, &name[0] both the declarations are same.

29. What is the output of the program?

```
main() increment() { { increment(); static inti=1; increment(); printf(“%d\n”,i) increment();  
i=i+1; } } Output: 1 2 3
```

30. Why header files are included in „C“ programming?

This section is used to include the function definitions used in the program. Each header file has „h“ extension and include using “# include” directive at the beginning of a program.

All statements should be written in lower case letters. Upper case letters are only for symbolic constants. Blank spaces may be inserted between the words. This improves the readability of statements. It is a free-form language; we can write statements anywhere between „{,„ and „}“
a = b + c; d = b*c;

31. What do you mean by variables in „C“?

A variable is a data name used for storing a data value. Can be assigned different values at different times during program execution. Can be chosen by programmer in a meaningful way so as to reflect its function in the program. Some examples are: Sum percent_1 class_total.

32. List the types of operators.

Arithmetic operators Relational operators Logical operators Increment and Decrement operators Assignment operators Bitwise operators Comma operator Conditional operator

33. Give the syntax for the „for“ loop statement

```
for (Initialize counter; Test condition; Increment / Decrement) { statements; }
```

34. What is the use of sizeof () operator?

The sizeof () operator gives the bytes occupied by a variable. No of bytes occupied varies from variable to variable depending upon its data types. **Example:** int x,y; printf(“%d”,sizeof(x));
Output: 2

35. Write a program to swap the values of two variables (without temporary variable).

```
#include <stdio.h> #include <conio.h> void main( ) { int a =5; b = 10; clrscr( ); printf(“Before  
swapping a = %d b = %d “, a , b); a = a + b; B = a – b; a = a – b; printf(“After swapping a = %d  
b = %d”, a,b); getch( ); } Output: Before swapping a = 5 b = 10 After swapping a = 10 b = 5
```

36. Write short notes about main () function in "C" program. (MAY 2009)

i) Every C program must have main () function. ii) All functions in C, has to end with „()“ parenthesis. iii) It is a starting point of all „C“ programs. iv) The program execution starts from the opening brace „{“, and ends with closing brace „}“, within which executable part of the program exists.

37. What is an array?

An array is a group of similar data types stored under a common name. `int a[10];` Here `a[10]` is an array with 10 values.

38. List the characteristics of Arrays.

All elements of an array share the same name, and they are distinguished from one another with help of an element number. Any particular element of an array can be modified separately without disturbing other elements.

3. What are the types of Arrays?

1. One-Dimensional Array 2. Two-Dimensional Array 3. Multi-Dimensional Array

39. What is the use of „\0“ character?

When declaring character arrays (strings), „\0“ (NULL) character is automatically added at end. The „\0“ character acts as an end of character array.

40. Define Strings.

The group of characters, digit and symbols enclosed within quotes is called as String (or) character Arrays. Strings are always terminated with „\0“ (NULL) character. The compiler automatically adds „\0“ at the end of the strings.

41. Write an example for declaring ARRAYS.

Arrays may consist of any of the valid data types.

Arrays are declared along with all other variables in the declaration section of the program. /*

Introducing array's */ #include <stdio.h>

main()

{

int numbers[100];

float averages[20]; numbers[2] = 10; --numbers[2];

printf("The 3rd element of array numbers is %d\n", numbers[2]); }

42. Write an example for assigning values to arrays.

The declaration is preceded by the word static. The initial values are enclosed in braces

Example

#include <stdio.h>

```

main()
{
int x;
static int values[] = { 1,2,3,4,5,6,7,8,9 };
static char word[] = { 'H','e','l','l','o' };
for( x = 0; x < 9; ++x )
printf("Values [%d] is %d\n", x, values[x]); }

```

43. Define Multi-dimensional array.

Multi-dimensioned arrays have two or more index values which specify the element in the array. multi[i][j];

Declaration: intm1[10][10]; static int m2[2][2] = { {0,1}, {2,3} };

44. Example for character arrays [strings].

```

#include <stdio.h>
main()
{
static char name1[] = { 'H','e','l','l','o' }; static char name2[] = "Hello"; printf("%s\n", name1);
printf("%s\n", name2);
}

```

45. With syntax and example mention the method declaring an array.

Syntax: Example: introllno[5]={4001,4002,4003,4004,4005};

46. Given an array intsalary[5]={10000,8500,15000,7500,9000}. Calculate the address of salary[4] if the base address=1000.

Here, the size of the datatype int is 2.

The value for w=2

Address of salary[4] = 1000 + 2(4-0) = 1000 + 8 = 1008

47. Given an array inta[10]={101,102,103,104,105,106,107,108,109,110}. Show the memory representation and calculate its length.

Memory Representation:

101	102	103	104	105	106	107	108	109	110
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]

48. List the different methods for reading and writing a string.

The different methods for reading a string are,

- scanf()
- gets()
- getchar()
- getch() or getche()

The different methods for writing a string are,

- printf()
- puts()
- putchar()

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

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

50. What is the use of gets() function? The gets() function allows a full line entry from the user. When the user presses the enter key to end the input, the entire line of characters is stored to a string variable.

51. Mention the various string manipulation functions in C.

S.No	Function	Purpose
1	strcpy(s1,s2)	Copies string s2 into string s1.
2	strcat(s1,s2)	Concatenates string s2 onto the end of string s1.
3	strlen(s1)	Returns the length of string s1
4	strcmp(s1,s2)	Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.
5	strchr(s1,ch)	Returns a pointer to the first occurrence of character ch in string s1.
6	strst(s1,s2)	Returns a pointer to the first occurrence of sting s2 in string s1.

52. List the primary data types in C (Nov/Dec 18)

Int, char, float, double

PART B

1. Illustrate the structure of C program with example
2. Explain about the various decision making statements in “C” language.
3. Explain operators and expressions in C
4. Explain the control statements in c.
5. Explain Input output operations in C
6. Explain string operations in detail with example.
7. Explain string arrays with example
8. Define arrays. Explain the array types with an example program for each type
9. Explain two dimensional arrays with a sample program-Matrix multiplication

10. Write a c program to perform factorial operation using recursive function(Nov/Dec 2018)

UNIT - II

FUNCTIONS, POINTERS, STRUCTURES & UNION

PART A

1. What is function?

To break a program into segments are commonly known as functions. Each of which can be written more or less independently of the others. Every function in the program is supposed to perform a well-defined task.

2. Why are functions needed?

- Dividing the program into separate well defined functions facilitates each function to be written and tested separately. This simplifies the process of getting the total program to work.
- Understanding, coding and testing multiple separate functions is easier than doing the same for one big function.
- If a big problem has to be developed without using any function other than main(), then there will be countless lines in the main() function and maintaining that program will be difficult task.

3. What is function prototype or function interface?

A function definition specifies how the function does what it does (implementation), a function prototype merely specifies its interface, (i.e) what data types go in and come out of it. The term function prototype is particularly used in forward declarations of functions in header files allows for splitting a program into translation units.

4. What is function declaration?

Before using a function, the compiler must know the number of parameters and the type of parameters that the function expects to receive and the datatype of value that it will return to the calling program. Placing the function declaration statement prior to its use enables the compiler to make a check on the arguments used while calling that function.

5. Write the function parts.

The functions have two parts are,

- Function header
- Function body

6. Write the syntax of function.

Syntax:

```
Return_datatypefunction_name(datatype var1, datatype var2,...)
{ ..... Statements .....
return (variable);
}
```

7. What is function call?

The function call statement invokes the function. When a function is invoked, the compiler jumps to the called function to execute the statements that are a part of that function. Once the called function is executed, the program control passes back to the calling function.

8. What are the points are to be noted while calling a function?

Function name and number of arguments, type of arguments in the function call must be same as that given in the function declaration and the function header of the function definition.

Names (not types) of variables in the function declaration, function call and header of function definition may vary.

Arguments may be passes in the form of expressions to the called function. In such a case, arguments are first evaluated and converted to the type of formal parameter and then the body of the function gets executed.

10. Differentiate call by value and call by reference.

Call by value: The values of the variables are passed by the calling function to the called function.

Call by reference: The addresses of the variables are passed by the calling function to the called function.

11. Write the advantages of call by reference technique.

The advantages of call by reference technique are,

- Arguments are not copied into the new variables, it provides greater time and space efficiency.
- Function can change the value of the argument and the change is reflected in the calling function.

12. Write the disadvantages of call by reference technique.

However, the drawback of using this technique is that if inadvertent changes are caused to variables in the called function then these changes would be reflected in calling function as original values would have been overwritten.

13. What is recursion?

Recursion is the process of repeating items in a self similar way. In programming languages, if a program allows you to call a function inside the same function, then it is called a recursive call of the function.

PART B

- 1. Explain the functions fscanf, fprintf, fgets, fputs, getw, putw,getc, putc, fread, fwrite, feof, ferror,fseek,ftell,rewind with example.**
- 2. Explain call by value and call by reference with example**
- 3. Explain the concept of recursion with example**
- 4. What are functions? Explain the types of functions in detail with an example program for each type**
- 5. Explain function pointers with example.**
- 6. Explain about function with variable number of arguments.**
- 7. Design a c program to create the employee database and to perform the manipulations such as adding a record, deleting a record, updating a record.**