PROGRAMMING FOR PROBLEM SOLVING

Subject Code : CS203ES
Regulation : R18 – JNTUH
Class : I B.Tech II Semester
(Common for CSE, IT, EEE Branches)

Department of Computer Science and Engineering

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY
Ibrahimpatnam - 501 510, Hyderabad

PROGRAMMING FOR PROBLEM SOLVING (CS203ES)
COURSE PLANNER

I. COURSE OVERVIEW:

This course emphasizes solving problems using the language, and introduces standard programming techniques like alternation, iteration and recursion. It will briefly glimpse the basics of software engineering practices like modularization, commenting, and naming conventions which help in collaborating and programming in teams. This course is enabled the students to formulate algorithms for arithmetic and logical problems, convert these algorithms to C language programs. It also aims on using arrays, pointers and structures to formulate algorithms and programs. In addition to that, apply programming to solve matrix addition and multiplication problems and searching and sorting problems.

II. PREREQUISITE(S):

- Mathematics knowledge, Analytical and Logical skills.

III. COURSE OBJECTIVES:

- To learn the fundamentals of computers.
- To understand the various steps in program development.
- To learn the syntax and semantics of C programming language.
- To learn the usage of structured programming approach in solving problems.

IV. COURSE OUTCOMES: The student will learn

Sno	Course Outcomes (CO)	Knowledge Level (Blooms Level)
CO1	To write algorithms and to draw flowcharts for solving problems.	L5: Evaluate
CO2	To convert the algorithms/flowcharts to C programs	L6: Create, L2:Understand
CO3	To code and test a given logic in C programming language	L6: Create, L4: Analyze
CO4	To decompose a problem into functions and to develop modular reusable code	L5: Evaluate, L3:Apply
CO5	To use arrays, pointers, strings and structures to write C programs	L5: Evaluate
CO6	Searching and sorting problems	L3:Apply

V. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes (POs)		Level	Proficiency assessed by
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems	3	Assignments
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	2	Assignments
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations	3	Experiments
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to	2	Experiments

	provide valid conclusions.		
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	-	-
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	-	-
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	-	-
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	-	-
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	-	-
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation make effective presentations, and give and receive clear instructions	-	-
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments	1	Mini Projects
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life long learning in the broadest context of technological change.	2	Research

VI. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes (PSOs)		Level	Proficiency assessed by
PSO1	**Software Development and Research Ability:** Ability to understand the structure and development methodologies of software systems. Possess professional skills and knowledge of software design process. Familiarity and practical competence with a broad range of programming language and open source platforms. Use knowledge in various domains to identify research gaps and hence to provide solution to new ideas and innovations.	3	Assignments
PSO2	**Foundation of mathematical concepts:** Ability to apply the acquired knowledge of basic skills, principles of computing, mathematical foundations, algorithmic principles, modeling and design of computer-based systems in solving real world engineering Problems.	2	Experiments
PSO3	**Successful Career:** Ability to update knowledge continuously in the tools like Rational Rose, MATLAB, Argo UML, R Language and technologies like Storage Computing, Communication to meet the industry requirements in creating innovative career paths for Immediate employment and for higher studies.	2	Experiments / Tools

1: Slight (Low) **2: Moderate (Medium)** **3: Substantial (High)** **- : None**

VII. SYLLABUS:

UNIT – I :

Introduction to components of a computer system: disks, primary and secondary memory, processor, operating system, compilers, creating, compiling and executing a program etc., Number systems
Introduction to Algorithms: steps to solve logical and numerical problems. Representation of Algorithm, Flowchart/Pseudo code with examples, Program design and structured programming
Introduction to C Programming Language: variables (with data types and space requirements), Syntax and Logical Errors in compilation, object and executable code , Operators, expressions and precedence, Expression evaluation, Storage classes (auto, extern, static and register), type conversion, The main method and command line arguments Bitwise operations: Bitwise AND, OR, XOR and NOT operators
Conditional Branching and Loops: Writing and evaluation of conditionals and consequent branching with if, if-else, switch-case, ternary operator, goto, Iteration with for, while, do-while loops
I/O: Simple input and output with scanf and printf, formatted I/O, Introduction to stdin, stdout and stderr. Command line arguments

UNIT—II:

Arrays: one and two dimensional arrays, creating, accessing and manipulating elements of arrays
Strings: Introduction to strings, handling strings as array of characters, basic string functions available in C (strlen, strcat, strcpy, strstr etc.), arrays of strings
Structures: Defining structures, initializing structures, unions, Array of structures
Pointers: Idea of pointers, Defining pointers, Pointers to Arrays and Structures, Use of Pointers in self-referential structures, usage of self referential structures in linked list (no implementation) Enumeration data type

UNIT- III:

Preprocessor: Commonly used Preprocessor commands like include, define, undef, if, ifdef, ifndef
Files: Text and Binary files, Creating and Reading and writing text and binary files, Appending data to existing files, Writing and reading structures using binary files, Random access using fseek, ftell and rewind functions.

UNIT- IV:

Functions: Designing structured programs, Declaring a function, Signature of a function, Parameters and return type of a function, passing parameters to functions, call by value, Passing arrays to functions, passing pointers to functions, idea of call by reference, Some C standard functions and libraries
Recursion: Simple programs, such as Finding Factorial, Fibonacci series etc., Limitations of Recursive functions Dynamic memory allocation: Allocating and freeing memory, Allocating memory for arrays of different data types

UNIT V:

Algorithms for finding roots of a quadratic equations, finding minimum and maximum numbers of a given set, finding if a number is prime number, etc.Basic searching in an array of elements (linear and binary search techniques), Basic algorithms to sort array of elements (Bubble, Insertion and Selection sort algorithms), Basic concept of order of complexity through the example programs

SUGGESTED BOOKS:

TEXT BOOKS:

1. **T1.** Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill.
2. **T2.** B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition).

REFERENCE BOOKS:

1. **R1**. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India
2. **R2**. R.G. Dromey, How to solve it by Computer, Pearson (16th Impression)

3. **R3.** Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
4. **R4.** Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition

NPTEL Web Course:

- nptel.ac.in/courses/106105085/4
- nptel.ac.in/courses/106105085/2

NPTEL Video Course:

- **https://onlinecourses.nptel.ac.in/noc18_cs31/preview**
- **https://onlinecourses.nptel.ac.in/noc18_cs33/preview**

GATE SYLLABUS: Programming in C. Recursion. Arrays, Searching, sorting

IES SYLLABUS: NA

VIII. COURSE PLAN:

Lecture	Week	Topics to be covered	Course Learning Outcomes	References
		UNIT-I		
1.	1	Introduction to components of a computer system: disks, primary and secondary memory, processor, operating system,	Illustrates different components	T2
2.		compilers, creating, compiling and executing a program		
3.		Number systems	Identify the steps in algorithm	
4.		Introduction to Algorithms: steps to solve logical and numerical problems. Representation of Algorithm		
5.	2	Flowchart/Pseudo code with examples, Program design and structured programming	Discuss the types of flowcharts and Understand the programming design	
6.		Introduction to C Programming Language: variables (with data types and space requirements), Syntax and Logical Errors in compilation, object and executable code ,		
7.		Operators	Describes about operators and expressions	
8.		expressions and precedence, Expression evaluation		
		Mock Test #1		
9.	3	Storage classes (auto, extern, static and register), type conversion	Describes various storage classes	T2
10.		The main method and command line arguments Bitwise operations: Bitwise AND, OR, XOR and NOT operators	Compute various methods	
11.		Conditional Branching and Loops: Writing and evaluation of conditionals and consequent branching with if, if-else, switch-case, ternary operator,		
12.		goto, Iteration with for, while, do-while loops	Creating different branching statements and loops	
		Tutorial / Bridge Class # 1		
13.	4	I/O: Simple input and output with scanf and printf, formatted I/O, Introduction to stdin, stdout and stderr. Command line arguments	Apply I/O operations	
		UNIT-II		

Lecture	Week	Topics to be covered	Course Learning Outcomes	References
14.	4	Arrays: one dimensional arrays, creating, accessing and manipulating elements of arrays	Applying the accessing techniques	
15.		two dimensional arrays		
16.		Strings: Introduction to strings, handling strings as array of characters	Defining strings and its characteristics	
		Tutorial / Bridge Class # 2		T2
17.	5	basic string functions available in C (strlen, strcat, strcpy, strstr etc.),	Describes various string methods	
18.		arrays of strings		
19.		Structures: Defining structures, initializing structures	Define a structure and its declaration	
20.		unions,		
		Tutorial / Bridge Class # 3		
21.	6	Array of structures		T2
22.		Pointers: Idea of pointers,	Understand the use of pointers	
23.		Defining pointers,	Defines a pointer	
24.		Pointers to Arrays and Structures	Types in it..	
		Tutorial / Bridge Class # 4		
25.	7	Use of Pointers in self-referential structures,	Examine in various methods	
26.		usage of self referential structures in linked list (no implementation)	Analyses the usages	
27.		Enumeration data type	Understanding the data types	
		Tutorial / Bridge Class # 5		
UNIT-III				
28.	8	Preprocessor: Commonly used Preprocessor commands like include, define, undef,	Demonstrate the preprocessor	
29.		if, ifdef, ifndef	Define the Loops	
30.		Files: Text and Binary files,	Understand the files	
31.		Creating and Reading and writing text and binary files	Evaluate text and binary files	
		Tutorial / Bridge Class # 6		
I-MID EXAMINATION (WEEK-9)				
32.	10	Appending data to existing files,	Evaluate the files	T2
33.		Writing and reading structures using binary files	Create the types in it	
34.		Random access using fseek, ftell and rewind functions.	Discuss about Random functions	

Lecture	Week	Topics to be covered	Course Learning Outcomes	References
35.		Example Programs on Random Access functions	Evaluate different examples	
		Tutorial / Bridge Class # 7		
UNIT-IV				
36.	11	Functions: Designing structured programs	Define a function	T2
37.		Declaring a function, Signature of a function,	Understand the declaration	
38.		Parameters and return type of a function	Classification of functions	
39.		passing parameters to functions, call by value	Evaluating types of function calls	
		Tutorial / Bridge Class # 8		
40.	12	Passing arrays to functions	Apply arrays	
41.		passing pointers to functions	Apply pointers with functions	
42.		idea of call by reference	Understand	
43.		Some C standard functions and libraries	Analyze some C functions	
		Tutorial / Bridge Class # 9		
44.	13	Recursion: Simple programs, such as Finding Factorial	Define a recursion	
45.		Fibonacci series etc.,	Create programs	
46.		Limitations of Recursive functions	Drawbacks of Recursion	
47.		Dynamic memory allocation: Allocating and freeing memory	Understanding about dynamic memory	
48.	14	**Mock Test - 2**		T2
49.		Allocating memory for arrays of different data types	Understanding its types	
UNIT-V				
50.	14	Algorithms for finding roots of a quadratic equations	Analyze various programming techniques	
51.		finding minimum and maximum numbers of a given set,	Analyze various programming techniques	
		Tutorial / Bridge Class # 10		
52.	15	finding if a number is prime number	Analyzing various programming techniques	

Lecture	Week	Topics to be covered	Course Learning Outcomes	References
53.		Basic searching in an array of elements - linear search technique	Analyzing various programming techniques	
54.		Basic searching in an array of elements - binary search technique	Analyzing various programming techniques	
55.		Basic algorithms to sort array of elements- Bubble sort algorithm	Analyzing various programming	
		Tutorial / Bridge Class # 11		
56.	16	Basic algorithms to sort array of elements - Selection sort algorithm	Analyzing various programming techniques	
57.		Basic algorithms to sort array of elements - Insertion sort algorithm	Analyzing various programming	
58.		Basic concept of order of complexity through the example programs	Compute various time complexities	
59.		UNIT-I Revision		
		Tutorial / Bridge Class # 12		
60.	17	UNIT-II Revision		
61.		UNIT-III Revision		
62.		UNIT-IV Revision		
63.		UNIT-V Revision		
		Tutorial / Bridge Class # 13		
		II MID EXAMINATIONS (WEEK 18)		

IX. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

	Program Outcomes (PO)												**Program Specific Outcomes (PSO)**		
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	3	2	-	-	-	-	-	-	1	2	3	2	2
CO2	3	2	3	2	-	-	-	-	-	-	1	2	3	2	2
CO3	3	2	3	2	-	-	-	-	-	-	1	2	3	2	2
CO4	2	2	3	2	-	-	-	-	-	-	-	1	2	2	1
CO5	2	1	3	1	-	-	-	-	-	-	-	1	2	1	-
CO6	2	2	3	2	-	-	-	-	-	-	1	1	3	2	2
AVG	2.5	1.8	3	1.8	-	-	-		-	-	0.66	1.5	2.66	1.8	1.5

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)* **-** *: None*

X. QUESTION BANK: (JNTUH)

DESCRIPTIVE QUESTIONS: (WITH BLOOMS PHRASES)

UNIT I Short Answer Questions-

Sno	Questions	Blooms Taxonomy Level	Course Outcome
1.	Define a)Variable b)Identifier c)Constant	Knowledge	1
2.	Write short notes on number system with examples	Apply	1
3.	Write short notes on jump statements?	Apply	1
4.	Define Flowchart and Algorithm	Knowledge	2
5.	Write short notes on Continue statements?	Apply	1

Long Answer Questions-

Sno	Questions	Blooms Taxonomy Level	Course Outcome
1.	Write a Short notes on Creating and Running the program?	Apply	3
2.	Write about Different data types along with memory?	Apply	4
3.	Write about Decision making statements with example?	Apply	3
4.	Write a Program on sum of n numbers using While and for loop?	Apply	3
5.	Describe about computer system	Create	3

UNIT – II
Short answer questions

Sno	Questions	Blooms Taxonomy Level	Course Outcome
1.	Write short notes on pointer Arithmetic Operations?	Apply	5
2.	Define Array?	Knowledge	5
3.	Define 1) Array of pointers 2) Pointers to functions.	Knowledge	5
4.	Define a String with a example?	Knowledge	5
5.	Discuss Difference between Structure and Unions	Understand	5

Long answer questions

Sno	Questions	Blooms Taxonomy Level	Course Outcome
1.	Write about String Input / Output functions with example?	Apply	5
1.	Write a C Program to reverse a given string with and without using functions?	Apply	5
2.	What are the types of Arrays? Explain in detail	Knowledge	5
3.	Write a C program that reads 15 names each of up to 30 characters,	Apply	5

	stores them in an array, and uses an array of pointers to display them in ascending (ie. alphabetical) order		
4.	What is Self referral Structure? Explain with example?	Knowledge	5

UNIT – III
Short answer questions

Sno	Questions	Blooms Taxonomy Level	Course Outcome
1.	Write Short notes on Preprocessor command 'define'	Apply	4
2.	Difference between undef and ifndef.		4
3.	Write the Difference between Binary File and Text File?	Apply	4
4.	Define different types of file input output functions?	Knowledge	4
5.	Define different file modes of operation?	Knowledge	4

Long answer questions

Sno	Questions	Blooms Taxonomy Level	Course Outcome
1	What are Preprocessor commands? List some Preprocessor commands with the example?	Knowledge	4
2	Define a macro that finds the maximum of two numbers. Write a C program that uses the macro and prints the maximum of two numbers.	Knowledge	4
3	Write about Positioning functions (fseek ,rewind and ftell)	Apply	4
4	Write a C program to count the number of times a character occurs in a text file. The file name and the character are supplied as command-line arguments.	Apply	3
5	Write about file status functions (error handling) with example?	Apply	4

UNIT – IV
Short answer questions

Sno	Questions	Blooms Taxonomy Level	Course Outcome
1.	What is a Function? Write about signature of a function?	Knowledge	4
2.	Write about limitation of recursion?	Apply	4
3.	Explain passing parameters to functions.	Analyze	4
4.	Explain passing pointers to functions.	Analyze	5
5.	Describe the allocation of memory for arrays of different data types	Create	5

Long answer questions

Sno	Questions	Blooms Taxonomy Level	Course Outcome
1.	Write about standard functions with examples?	Apply	4
2.	Explain about memory allocation functions with example.	Evaluate	4
3.	What is recursion? Write a program for factorial using recursion?	Knowledge	4
4.	Difference between call by value and call by reference.	Analyze	4

5.	Write a recursive function for Fibonacci series.	Apply	4

UNIT – V

Short answer questions

Sno	Questions	BloomsTaxonomy Level	Course Outcome
1.	Define order of complexity.	Knowledge	4
2.	What is an algorithm and what are its advantages?	Knowledge	2
3.	Write an algorithm to check whether a given number is prime or not.	Apply	2
4.	Differences between linear search and binary search.	Analyze	6
5.	Compare different sorting algorithms.	Analyze	6

Long answer questions

Sno	Questions	Blooms Taxonomy Level	Course Outcome
1)	Write algorithm to demonstrate binary search.	Apply	6
2)	Explain the algorithm for finding roots of a quadratic equations	Evaluate	2
3)	Define algorithm for finding minimum and maximum numbers of a given set	Knowledge	2
4)	Explain Bubble sort algorithm with its order of complexity	Evaluate	6
5)	Explain the techniques of Selection sort and Insertion sort with the help of an example array.	Evaluate	6

XI. OBJECTIVE QUESTIONS: JNTUH

UNIT: I

1. The format identifier ‘%i’ is also used for _____ data type?
a) char b) int c) float d) double

2. 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

3. What is the output of the following code. int n=0, m=1; do { printf("%d", m); m++; } while (m<=n);
A) 0 B) 2 C) 1 D) 4

4. Which of the following is the correct usage of conditional operators used in C?
A) a>b?c=30:c=40; B) a>b?c=30;
C) max=a>b?a>c?a:c:b>c?b:c D) return (a>b)?(a:b)

5. Which symbol is used as a statement terminator in C?
A) , B) “ C) ; D):

6. If a=8 and b=15 then the statement x=(a>b) ? a:b;
A) assigns a value 8 to x B) gives an error message
C) assigns a value 15 to x D) assigns a value 7 to x

7. In expression i = g() + f(), first function called depends on
a) Compiler b) Associativiy of () operator
c) Precedence of () and + operator d) Left to write of the expression

8. How many times the following loop be executed?
{ .. ch='b'; while (ch>='a' && ch<=='z') ch++; }
A) 0 B) 25 C) 26 D) 1

9. function tolower(c) defined in library works for
a) Ascii character set b) Unicode character set
c) Ascii and utf-8 but not EBSIDIC character set d) Any character set

10. Which type 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

UNIT: II

1. 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);

2.The elements in the array of the following code are
int array[5] = {5};
a) 5, 5, 5, 5, 5 b) 5, 0, 0, 0, 0 c) 5, (garbage), (garbage), (garbage), (garbage)
d) (garbage), (garbage), (garbage), (garbage), 5

3. Different ways to initialize an array with all elements as zero are
a) int array[5] = {}; b) int array[5] = {0};
c) int a = 0, b = 0, c = 0; d) int array[5] = {a, b, c};

4. Which of the following operand can be applied to pointers p and q?
(Assuming initialization as int *a = (int *)2; int *b = (int *)3;)
a) a + b b) a – b c) a * bd) a / b

5. Size of a union is determined by size of the.
a) First member in the union b) Last member in the union
c) Biggest member in the union d) Sum of the sizes of all members

6. Members of a union are accessed as________________.
a) union-name.member b) union-pointer->member
c) Both a & b d) None of the mentioned

7. The correct syntax to access the member of the ith structure in the array of structures is?

```
Assuming: struct temp
{
   int b;
}s[50];
```

a) s.b.[i]; b) s.[i].b; c) s.b[i]; d) s[i].b;

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

UNIT: III

1. For binary files, a ___ must be appended to the mode string.
a) Nothing b) “b” c) “binary” d) “01”

2. If there is any error while opening a file, fopen will return
a) Nothing b) EOFc) NULL d) Depends on compiler

3. What is the meant by ‘a’ in the following operation?
fp = fopen(“Random.txt”, “a”);
a) Attach b) Append c) Apprehend d) Add

4. putchar(c) function/macro always outputs character c to the
a) screen b) standard output
c) depends on the compiler d) Depends on the standard

5. stderr is similar to?
a) stdinb) stdout c) Both stdout and stdin d) None of the mentioned

6. What is the output of this C code?

```
#include <stdio.h>
#define foo(x, y) #x #y
int main()
{
   printf("%s\n", foo(k, l));
   return 0;
}
```

a)kl b)xy c) Compile time error d) k l

7. 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

8. What is the purpose of the function?
int ferror(FILE *fp)
a) They check for input errors b) They check for output errors
c) They check for all types of errors d) They check for error in accessing the file

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

10. 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

UNIT: IV

1. In C, if you pass an array as an argument to a function, what actually gets passed?
A) Value of elements in array **B)** First element of the array
C) Base address of the array **D)** Address of the last element of array

2. What is the return-type of the function sqrt()
a) int b) float c) double d) Depends on the data type of the parameter

3. What is the problem in the following declarations?

```
   int func(int);
   double func(int);
   int func(float);
```

a) A function with same name cannot have different signatures
b) A function with same name cannot have different return types
c) A function with same name cannot have different number of parameters
d) All of the mentioned

4. Which type of variables can have same name in different function:
a) global variables b) static variables
c) Function arguments d) Both (b) and (c)

5. The maximum number of arguments that can be passed in a single function are_____________
a) 127 b) 253 c) 361 d) No limits in number of arguments

6. What is the output of the following code?

```
#include <stdio.h>
    int main()
    {   printf("%d", main);    return 0; }
```

a) Address of main function c) Compiler Error
b) Runtime Error d) Some random value
Ans) A

7. A function has a ____ , a _____ and a __ Ans) Name, data type and list of arguments
8. A function with no return type is declared as ________ Ans) void
9. When a function calls itself again and again, it is called _________ Ans) Recursion

UNIT: V

1. Before searching, the list of items should be sorted in ascending order can be done in [ ]
a) Binary searching b) Linear searching c) Both A&B d)None

2. Binary search is effective only when the elements are in

a) ascending order b) descending order c) a& b d) jumbled order

3) In bubble sort ___________ element is settled first in its position in first pass
4) A node contains ___________ and _________.
5) In binary search key is compared with __________ element.
a) First b) last c) middle d) none
6) Time Complexity of linear Search is ________________
7) Time Complexity of Binary Search is ________________
8) Time complexity of Bubble sort is ___________________
9) Time Complexity of Selection sort is____________________
10) Time Complexity of Insertion sort is____________________

XII. GATE QUESTIONS:

1. The number of tokens in the following C statement.

`printf("i = %d, &i = %x", i, &i);` is

(a) 3 (b) 26 (c) 10 (d) 21 **Ans) C**

2. Consider the following C function definition:

```
int Trial (int a, int b, int c)
{
   if ((a > = b) && (c < b)) return b;
   else if (a > = b) return Trial (a,c,b);
   else return Trial (b,a,c);
}
```

The function Trial:

(a) Finds the maximum of a, b, and c

(b) Finds the minimum of a, b and c

(c) Finds the middle number of a, b, c Ans) C

(d) None of the above

3. Consider the following declaration of a ‘two-dimensional array in C:

```
char a[100][100];
```

Assuming that the main memory is byte-addressable and that the array is stored starting from memory address 0, the address of a[40][50] is

(a) 4040 (b) 4050 (c) 5040 (d) 5050 Ans) B

4. Consider the following C function:

```
int f(int n)
{
static int i = 1;
if (n >= 5)
return n;
   n = n+i;
   i++;
   return f(n);
}
```

The value returned by f(1) is Ans) C

a) 5 b) 6 c) 7 d) 8

And Find many more questions from the following links

- https://www.geeksforgeeks.org/tag/gate-cs-c-language/
- https://www.geeksforgeeks.org/quiz-corner-gq/
- http://www.btechonline.org/2015/12/gate-questions-c-programming.html
- http://www.techvyom.com/c-programming-solved-questions-from-previous-years-gate-papers.html

XIII. WEBSITES:

1. http://www.kciti.edu/wp-content/uploads/2017/07/cprogramming_tutorial.pdf
2. https://www.codewithc.com/programming-with-c-pdf-byron-gottfried/
3. https://phy.ntnu.edu.tw/~cchen/pdf/ctutor.pdf
4. https://www.codewithc.com/c-projects-with-source-code/

XIV. EXPERT DETAILS: NA

XV. JOURNALS:

INTERNATIONAL:

- **https://www.cprogramming.com/codej.html**
- **https://ieeexplore.ieee.org/document/5387762/**

NATIONAL:

- **https://www.nationaljournal.com/**
- **https://www.sciencedirect.com/browse/journals-and-books**

XVI. LIST OF TOPICS FOR STUDENT SEMINARS:

1. Arrays and its Types
2. Self referential Structures
3. Dynamic memory Allocation Functions
4. Sorting Techniques
5. Searching Techniques

XVII. CASE STUDIES / SMALL PROJECTS:

1) Telecom Billing System
2) Cricket Score Sheet
3) Bank Management System
4) School Billing System
5) Hospital Management System