

PROGRAMMING FOR PROBLEM SOLVING
(CSEN 1001)

Time Allotted : 3 hrs

Full Marks : 70

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 5 (five) from Group B to E, taking at least one from each group.

Candidates are required to give answer in their own words as far as practicable.

Group - A
(Multiple Choice Type Questions)

1. Choose the correct alternative for the following: **10 × 1 = 10**

- (i) Which of the following statements is FALSE?
(a) Array bound is not checked in C
(b) Local variable gets precedence over global variable
(c) Garbage collection is not automatic in C
(d) main() is a function with no default return type.
- (ii) In which step of the program compilation process, macros are dealt with?
(a) Assembly
(b) Pre-Processing
(c) Linking
(d) Loading.
- (iii) What will be the output of the program snippet?
int main(void)
{
constint a = 10;
int *p = &a;
*p = 20;
printf("%d",a);
return 0;
}
(a) Compiler error (b) 10 (c) 20 (d) None of the above.
- (iv) What will be the output of the program snippet?
int main(void)
{
int a = 10;
float b = 2.5;
if(b = 0)
{
printf("%d, ",printf("%d",b));
}

```
    }
    else
    {
printf("%d, ",printf("%d",b));
    }
    return 0;
}
```

- (a) 0, 1 (b) 1, 0 (c) 2, 1 (d) 1, 2.

(v) What will be the output of the program snippet?

```
int main(void)
{
int a, x, count = 0;
for(a = 1; a >= 1 ; a = a << 2)
{
count++;
}
printf("%d\n",count);
return 0;
}
```

- (a) 16 (b) 15 (c) 1 (d) Cannot be said, Infinite loop.

(vi) Given a if statement with three compounded conditions, which of the below would ensure that a statement inside the if block can be reached when **condition1** of the three conditions is **FALSE**.

- (a) if(condition1 && condition2 && condition3)
(b) if(condition1 || condition2 && condition3)
(c) if(condition1 && condition2 || condition3)
(d) None of the above.

(vii) What will be the output of the program snippet?

```
int a = 15;
int main(void)
{
int a = 10;
{
printf("%d ", a);
int a = 20;
}
printf("%d ", a);
return 0;
}
```

- (a) 10 10 (b) 10 20 (c) 15 20 (d) 15 10

(viii) What will be the output of the following C code segment?

```
#define CUBE(x) x * x * x
int main ( )
{
```

```
printf ("%d", CUBE (2 + 3));
return 0;
}
```

- (a) 17 (b) 125 (c) 35 (d) 18.

- (ix) Programs are executed in
 (a) ROM (b) Flash Drive (c) RAM (d) Registers.
- (x) Assume that x is a short integer variable. Then, $x \gg 3$ is same as,
 (a) $x/3$ (b) $x*8$ (c) $x*3$ (d) $x/8$.

Group- B

2. (a) (i) What is $(48A6.B2)_{16}$ in decimal? Show the conversion steps.
 (ii) Compute $(-234.25)_{10} + (75.125)_{10}$ using 1's complement method.
 (iii) $(56.125)_{10} = (?)_8 = (?)_2$. Show the conversion steps.
[(CO1)(Understand/LOCQ)]
- (b) Draw a flowchart of an optimal algorithm to find whether a positive integer entered by user is prime or not.
[(CO2)(Analyze/IOCQ)]
 $[2 + 3 + (1.5 \times 2)] + 4 = 12$
3. (a) Convert the decimal number $(145.625)_{10}$ in IEEE-754 32-bit single precision format.
[(CO1)(Understand/IOCQ)]
- (b) Compute $(101101101)_2 - (110111)_2$ using 2's complement method.
[(CO1)(Understand/LOCQ)]
- (c) How the Compilation Process works for C Programs? Explain briefly.
[(CO2)(Remember/LOCQ)]
 $5 + 3 + 4 = 12$

Group - C

4. (a) (i) Explain explicit and implicit type casting with an example
 (ii) Can we use these techniques to round off a floating point number? If yes then write a program that will round off a decimal number to its approximate integer number note that a decimal number can be +ve as well as -ve.
[(CO2, CO3, CO4)(Remember, Apply/LOCQ, IOCQ)]
- (b) What will be the output of the given programme segments? *(Please provide suitable explanation)*

```
(i) int main(void)
    {
        int a = (int) (pow(2.0, 31.0) - 1);
        while(a > 0)
        {
            a++;
            printf("%d", a);
        }
        return 0;
```

```

    }
(ii) int main(void)
    {
        int a, b = 2, c = 1, z;
        if(a = 0)
            z = ++ a || b-- && c-- ;
        else
            z = ++ a && b-- || c-- ;
        printf("A: %d, B: %d, C: %d, Z: %d\n", a, b, c,
z);
        return 0;
    }

```

[(CO3)(Analyze/IOCQ)]
(3 + 4) + (2.5 + 2.5) = 12

5. (a) Write a c program to find the result of the given series up to the nth term, where user will provide the value of n and x.

$$x - x^3 / 3 + x^5 / 5 - x^7 / 7 + x^9 / 9 + \dots$$

[(CO5)(Create/IOCQ)]

- (b) Write a c program to print this pattern, where the number of rows will be taken as an input from the user. For example if number of rows = 3 then the pattern would look like this.

```

    1           1
    2 2       2 2
    3 3 3 3 3 3

```

[(CO5)(Create/IOCQ)]
6 + 6 = 12

Group - D

6. (a) Write a recursive C function to check whether a given number is an Armstrong number or not. The recursive function would return 1 if the passed number is an Armstrong number else it will return 0. The signature of the function is given below:

```
int recurArmstrong(int n);
```

(Armstrong number: It is a number that is the sum of its own digits each raised to the power of the number of digits gives the number itself.

For example: 153 = 1³ + 5³ + 3³ or 1634 = 1⁴ + 6⁴ + 3⁴ + 4⁴)

[(CO6) (Create/HOCQ)]

- (b) State the difference between:

(i) Macro vs. Function

(ii) malloc vs. Calloc.

[(CO3)(Remember,Understand/IOCQ)]

7 + (2.5 + 2.5) = 12

7. (a) Write C functions to compare two strings, and concatenate two strings without using String library functions. The functions should have the following prototypes

```
int strCompare( const char *,const char *);
char *strConcat ( char *, const char *);
```

[[CO5) (Analyze/HOCQ)]

- (b) Consider the following recursive function. Assume that both n, k are positive.

```
int S( int n, int k ){
    if (k > n) return 0;
    if ( (k == 1) || (k == n) ) return 1;
    return S(n-1,k-1) + k * S(n-1,k);
}
```

What value will be returned by S (5, 3)? Show step by step execution of the function.

[[CO2) (Analyze/HOCQ)]

$$(3 + 3) + 6 = 12$$

Group - E

8. (a) Write a C function that will take only an integer array and its size as two input parameters and will return the minimum, the maximum, and the average of the integer array. *Note that the average of an integer array can be decimal and usage of global data types is discouraged!!!.* [[CO5, CO6)(Create/HOCQ)]

- (b) Write a C program that takes two strings as command line arguments and compares them to see whether they are same or not. [[CO5)(Create/HOCQ)]

$$7 + 5 = 12$$

9. (a) Distinguish between the file open modes “w+” and “a+”. [[CO3)(Remember/LOCQ)]

- (b) Explain the function feof() with an example. [[CO3)(Remember/LOCQ)]

- (c) Write a C program to read last 5 characters from a text file. [[CO5)(Understand/LOCQ)]

- (d) Create a structure that has two elements, one is name and the other is marks. Create an array of structure of size 5. Enter data in the array using some input method. Print the name who secured minimum marks. [[CO2)(Analyse/IOCQ)]

$$2 + 2 + 2 + 6 = 12$$

Cognition Level	LOCQ	IOCQ	HOCQ
Percentage distribution	29	38.5	32.5

Course Outcome (CO) :

After the completion of the course students will be able to

(CO1) Understand and remember functions of the different parts of a computer.

(CO2) Understand and remember how a high-level language (C programming language, in this course) works, different stages a program goes through.

- (C03)** Understand and remember syntax and semantics of a high-level language (C programming language, in this course).
- (C04)** Understand how code can be optimized in high-level languages.
- (C05)** Apply high-level language to automate the solution to a problem.
- (C06)** Apply high-level language to implement different solutions for the same problem and analyze why one solution is better than the other.

*LOCQ: Lower Order Cognitive Question; IOCQ: Intermediate Order Cognitive Question;
HOCQ: Higher Order Cognitive Question