Graphic Era HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)
University under section 2(f) of UGC Act, 1956

Term Evaluation (Even) Semester Examination March 2025

Roll no. 2494056

Name of the Program: B.Tech — Semester: II
Name of the Course: Programming for Problem solving
Course Code: **TCS201**
Time: 90 Minutes — Maximum Marks: 50

Note:

(i) Answer **all the questions except Que(2)** by choosing ***any one of the sub-questions***.
(ii) All 5 sub-questions in **Que(2)** are compulsory.
(iii) Each question carries 10 marks.

Q1 — **(10 marks)**

(a) Describe the various ways to read a string in 'C' language with their suitable syntax. Write a 'C' program to read an alphanumeric string. Display the string by converting digits into words.

Sample Input:	**Sample Output:**
Input String: Pincode of my city is 248001	Output String: Pincode of my city is two four eight zero zero one

CO1

OR

(b) Write a code in 'C' to read a multiword string. Input two indexes a and b. Reverse the string starting from index a to index b. Print the final string.

Sample Input:	**Sample Output:**
Input String: Delhi is the capital of India Input index(a): 14 Input index(b): 20	Output String: Delhi is the latipac of India

Q2 — **(10 marks)**

Find the Output with an explanation. Considering *int* is getting 2 Bytes of space in memory.

CO1
CO3

1.
```
#include<stdio.h>
int main()
{
   char str[]= "how are you?";
   int i, j, t, len;
   len = strlen(str);
   for(i=0, j=len; i<j; i++, j--)
   {
      t = str[i];
      str[i] = str[j];
      str[j] = t;
   }
   printf("%s", str);
}
```

2.
```
#include<stdio.h>
int main()
{
   int arr[]={0, 1, 2, 3, 4, 5, 10};
   static int x, y, k;
   int *p = arr;
   for(k=0; k<7; k++)
   {
      if(*(p+k)%2==0 && *(p+k)%5==0)
         ++x;
      else
         ++y;
   }
   printf("%d %d %d", x, y, x-y);
```

		return 0; }	

	3.
	#include<stdio.h>
	int main()
	{
	int a = 12, b = 4, *p1, *p2;
	int x, y;
	p1 = &a;
	p2 = &b;
	x = *p1 * *p2 - 6;
	y = 4 * - (*p2) / *p1 + 10;
	printf("%d %d", x, y);
	return 0;
	}
		4.
		#include<stdio.h>
		int main()
		{
		char str[] = {'a', 'b', '\0', 'x', 'y', '\0'};
		printf("%d", strlen(str));
		printf("%d", sizeof(str));
		return 0;
		}	

	5.
	#include<stdio.h>
	int main()
	{
	char str1[20] = "Hello";
	char str2[20] = "World";
	printf("%s", strcpy(str2, strcat(str1, str2)));
	return 0;
	}		

Q3			**(10 marks)**
(a)	Draw a flowchart to read a multiword string. Copy this string to another string without spaces. Display the final copied string. **Sample Input:** / **Sample Output:** Input String (str1): my city is a beautiful valley Output String(str2): mycityisabeautifulvalley		CO2
	OR		
(b)	Draw a flowchart to input elements into a matrix of size *mXn*. Print maximum element of each column of matrix. **Sample Input:** / **Sample Output:** Number of rows: 3 Number of columns: 4 Matrix elements: 34 56 12 89 23 45 78 38 15 21 67 92 Maximum of column 1: 34 Maximum of column 2: 56 Maximum of column 3: 78 Maximum of column 4: 92		
Q4			**(10 marks)**
(a)	To solve a mathematical problem Aman inputted elements into a matrix of size *mXn*. Write a 'C' program to help him to check whether he created Sparse matrix or not. **Note**: Sparse matrix is a matrix where most of the elements are zero. **Sample Input:** / **Sample Output:** Inputted Matrix: 0 1 0 0 2 0 0 5 0 3 0 0 Sparse Matrix		CO1

	0 0 0 0 7 2 0 0		
	Inputted Matrix: 0 1 8 6 2 4 0 5 9 3 0 0 0 0 6 0 7 2 0 0	Not a Sparse Matrix	
	OR		
(b)	Write a 'C' program to input elements into a matrix of size *mXn*. Input two rows index and interchange element to both rows.		
	Sample Input:	**Sample Output:**	
	Number of rows: 5 Number of columns: 3 Elements of matrix: 2 5 7 **4 6 8** 1 7 9 **3 5 2** 8 9 3 Row Index1: 1 Row index2: 3	Final matrix after interchange: 2 5 7 **3 5 2** 1 7 9 **4 6 8** 8 8 3	
Q5		**(10 marks)**	
(a)	Describe various types of pointers available in 'C' programming language. How call by reference is different with call by value? Explain with an example.		
	OR		
(b)	Write a 'C' program to read two 1D array. Create a new array by finding the intersection of two inputted array. (access elements of array using pointer)		CO3
	Sample Input:	**Sample Output:**	
	Number of elements in arry1: 4 Elements of array1: 2 4 6 8 Number of elements in arry2: 6 Elements of array2: 3 4 6 8 9 5	Elements of output array(arry3): 4 6 8	