Roll No. | | | | | | | |

Paper Code: TCS-201

Even End Sem Back Paper Examination December 2022

Course - B.Tech *Semester –2nd*

Subject Name: Programming for Problem Solving

Subject Code: TCS 201

Time: Three Hours

MM: 100

Note:

i. This question paper contains five questions.
ii. All Questions are compulsory.
iii. Instructions on how to attempt a question are mentioned against it.
iv. Total marks assigned to each questions are **twenty.**

Q1. (Attempt any two questions of choice from **a, b and c**). (2*10=20 Marks)

a.

i. Design a C program to accept N real numbers from the user, compute the average and then store the numbers those are above the average in another array called **aboveAvg** and below average in the array **belowAvg**. Display the average and content of both the arrays. Write the corresponding algorithm for the same.
ii. Describe multi-dimensional arrays with an illustration. List the advantages of Arrays in real world applications.

b.

i. Define an Array. Explain & illustrate the significance of **Row Major Order** method of storage of array elements.
ii. Implement a C program by accepting an integer through the command line arguments and compute its factorial. Display the computed factorial to the output screen.

c. Write a C program to check that inputted string is palindrome or not. (without using string library function)

Q2. (Attempt any two questions of choice **from a, b and c**). (2*10=20 Marks)

a.

i. Explain the storage of strings in C by emphasizing on the significance of **null character**. Describe the working of **strncmp** and **strncpy** functions with an example.
ii. Accept a sentence from the user and capitalize the first letter of each of the words of the sentence. Assume the words are separated uniformly by a single space. Implement using a C program. Draw a flowchart for the same.

b.

i. "***Usage of dynamic memory allocation by a programmer leads to more efficient usage of memory than a static memory allocation***". Justify the statement with an illustration.

ii. Design a non-recursive C function that will test whether a pattern string is the prefix of a text string or not. Accept the text string and the pattern string in the calling program and display an appropriate message to the output screen if the pattern is present or not present.

Example:
"ABC" is a prefix of **"ABC Company"**, but **"AABC"** is not.

c. Write a program to accept name, age and basic salary of 5 employee and calculate the total salary of the employee as-
Total sal =Basic +TA+DA where TA is 15% of Basic Salary, DA is 5% of Basic Salary.
Display the information of the employees in the ascending order of their total sal.

Q3. (Attempt any two questions of choice **from a, b and c**). (2*10=20 Marks)

a. Define the following terms with an example for each: **2.5 x 4 = 10 M**

i. dangling pointer
ii. null pointer
iii. wild pointer
iv. segmentation fault

b.

i. **"Pointers can be used alternatively to the access the arrays"**. Justify & demonstrate using a C program.
ii. Implement a C program to accept numbers using pointers. Design a function **Sort** that sorts and returns the sorted numbers in descending order using pointers. Display the sorted numbers in the main program.

c. What is dereferencing of pointer variable? Explain how these expression will work assuming that p is a integer pointer and holding address 300-

1. x=*p++	2. p=p+2
3. x=*(p+2)	4. x=++*p

Q4. (Attempt any two questions of choice **from a, b and c**). (2*10=20 Marks)

a.

i. Illustrate different ways of declaring a structure. A certain grocery store would like to store ten different categories of items with their quantities and prices. Write a structure declaration for the same in C.
ii. Design a program in C that numbers each of the lines in a file. Accept N lines from the user and write to a file **"numbered.txt"** by pre-fixing a line number followed by a space to each of sentences of the file.

For example: 1 First Sentence.....
2 Second Sentence .....
3 Third Sentence..... & so on till
:

N Nth Sentence.

b.

i. Elaborate on working of files using **fscanf** and **fprintf** functions. Explain with a snippet of C code.

ii. A Nursery plantation house would like to automate to maintain different saplings breeds based on their name, number of saplings and the price per sapling. Write a C program using appropriate data types to store the details of N saplings. Display the total cost of maintaining the Nursery to the output screen

Sample Input:
For N=2
Enter the name of the sapling: **Red Rose**
Enter the quantity: 200
Enter the price per sapling: 50
Enter the name of the sapling: **Hibiscus**
Enter the quantity: 100
Enter the price per sapling: 35

Total Cost of maintaining the Nursery (Rs): **13500**

c. Write a C program to read records from a file and calculate grade of each student and display it.
Grade - A if marks>=75
Grade - B if marks>=60
Grade - C if marks>=50
FAIL if marks>=0

Q5. (Attempt any two questions of choice **from a, b and c**). (2*10=20 Marks)

a.

i. Describe the working of **range** function with a sample code illustrating the same. **3M**

ii. Write a Python code to accept a list of integers from the user and then display the same by sorting it in descending order. **4M**

iii. List the differences between list and a tuple. **3M**

b.

i. Dictionary in Python is more flexible data structure that allows storing a pair of items. Support the statement with an example.

ii. Design a program in Python to accept & store the name of items with their respective quantities to a dictionary. Then eliminate duplicate items from the dictionary and display it to the console. Write the corresponding algorithm for the same.

c. Find Output for the following code-

(Consider the following code for a 16 bit compiler. Ignore the punctuation error, if any and header files)

i)int main()
[2]
```
{
   char str[]={'H','E','L','P','\0','\n','M','E','\0', '\0'};
   printf("%s\n",str);
   printf("%d  %d\n",strlen(str),sizeof(str));
   return 0;
}
```

ii)int sum(int); [4]
```
int main()
{
        printf( " %d ", sum(146) );
        printf( " %d ", sum(681) );
        return 0;
}
int sum(int n)
{
        static int s;
        while(n!=0)
        {
                s=s+n%10;
                n=n/10;
        }
        return sum;
}
```

iii)int main() [2]
```
{
 enum value={val1=0,val2,val3,val4,val5}var;
printf("%d",sizeof(var));
return 0;
}
```

iv)int main() [2]
```
{
 struct emp
{
  char n[20];
  int age;
};
struct emp e1={"Dravid",23};
struct emp e2=e1;
if(e1==e2)
printf("The structure variable are equal ");
}
```