CHRIST DEEMED TO BE UNIVERSITY,BENGALURU - 560029

End Semester Examination September/October - 2019

Master of Computer Applications I SEMESTER

Code: MCA131 **Max.Marks: 100**

Course: PROGRAMMING IN C **Duration: 3Hrs**

SECTION A

Answer all the questions. **5X20=100**

1 **a)** Write a C program that accepts an employee's ID, total worked hours of a month and the amount he received per hour. Print the employee's ID and salary (with two decimal places) of a particular month. **(10)**

[OR]

2 **b)** Write C a program to display prime numbers between a lower limits (low) and upper limit (high). These limits are to be accepted from user. Include appropriate error handling. **(10)**

3 **a)** Discuss the output of the below program. You may assume that this program is executed on a 32-bit machine **(10)**

```
#include <stdio.h>
int main()
{
int num = 10;
float fnum = 10.0;
double dnum = 10.0;
char ch = 'H';
int *iptr = &num;
double *dptr = &dnum;
printf("size of num is : %d\n", sizeof(num));
printf("size of short num is : %d\n", sizeof(fnum));
printf("size of long num is : %d\n", sizeof(dnum));
printf("size of ch is : %d\n", sizeof(ch));
printf("size of iptr is : %d\n", sizeof(iptr));
printf("size of cptr is : %d\n", sizeof(dptr));
return 0;
}
```

[OR]

4 **b)** Write a program that accepts integers from the keyboard until we enter a zero or a negative number. The program will output the number of positive values entered, the minimum value, the maximum value and the average of all numbers. **(10)**

5 **a)** What is a recursive function? Explain the working of any sample recursive function by depicting the stack frames created for it. **(10)**

[OR]

6 **b)** Explain **extern** storage class. What are its applications? Discuss with the help of an example. **(10)**

7 **a)** Differentiate between call by value and call by reference through sample programs. Explain with the help of stack frames. **(10)**

[OR]

8 **b)** What is stack overflow error? Explain with the help of an example. **(10)**

9 **a)** Compare and contrast between ***scanf*** and ***fgets*** functions to accept strings. Give suitable examples. **(10)**

[OR]

10 **b)** Explain the output of below program. Clearly mention the assumptions made by you to predict the output. **(10)**

```
#include <stdio.h>
```

```
int main()

int num; char ch; float price; double value;
void *vptr = NULL;
printf("%d\n", sizeof(vptr));
vptr = &num;
printf("%d\n", sizeof(vptr));
vptr = &ch;
printf("%d\n", sizeof(vptr));
vptr = &price;
printf("%d\n", sizeof(vptr));
vptr = &value;
printf("%d\n", sizeof(vptr));
return 0;
}
```

11 a) ***gets()*** is a dangerous function to use. Justify this statement. Explain how ***fgets*** function is better off than *gets*. **(10)**

[OR]

12 b) What is an *array of pointer*? What are its applications? Implement a program to demonstrate array of pointers. **(10)**

13 a) Write a program to accept a maximum of 10 integers into an array and find out the largest element in the array. The exact number of integers is to be accepted from user. Include appropriate error handling. The program should follow necessary coding guidelines. **(10)**

[OR]

14 b) Explain array overflow error with respect to strings. Give sufficient examples to bring out possible scenarios. **(10)**

15 a) Explain how the bug "function returns the address of a local variable" can be solved using dynamic memory allocation functions. Give suitable examples. **(10)**

[OR]

16 b) Explain the output of below program. What are the bugs in it? **(10)**

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define STRING_SIZE 5
int main()
{
char s_name[STRING_SIZE];
char *p_name = NULL;
int i_strl = 0;
strcpy(s_name, "Divya");
i_strl = strlen(s_name);
p_name = (char *) malloc(i_strl * sizeof(char));
if(NULL == p_name)
{
printf("%s","malloc failure");
exit(0);
}
p_name = "Manju";
printf("%s%s\n", p_name, s_name);
strcpy(p_name, s_name);
printf("%s%s\n", p_name, s_name);
return 0;
```

}

17 **a)** What are the different ways to pass a structure to a function? Explain their merits and demerits with suitable examples. **(10)**

[OR]

18 **b)** Write a program to demonstrate read and write operations on a binary file. Do necessary validations. **(10)**

19 **a)** Explain **enum** and **typedef** with examples. **(10)**

[OR]

20 **b)** Write a C program to copy the contents of one file to another file. **(10)**