Computer Programming Set2
End Semester Exam

Instructions

1. This is a **closed book online proctored** exam.
    a. You should not refer to books, notes or online resources.
    b. You should not discuss questions or answers with anyone (including outsiders)
    c. You should have your camera and microphone **ON** at at all times and no headphones
2. Write the solutions clearly and legibly in A4 sheets, using pen (NOT pencil) and at the end of the exam you should submit the scanned copy of your solutions as explained by the faculty
3. **The name of the scanned copy should be the Roll No + '_' + Set No. (e.g., S20200010XYZ_Set2).**
4. **Write the name and the roll no. on each page of the answer sheets.**
5. Follow all other instructions given by the faculty during the exam

Subjective Questions (10 marks each)

1. Answer the following questions.

a) What is a tail recursion? Explain with an example. State the difference between simple recursion and tail recursion?

b) Figure out the output of the below code. Assume Necessary libraries have been included.

```
#include <stdio.h>
char fun();
int main(void)
{
  printf("%d %c\n", fun(), fun());
  return 0;
}
char fun()
{
  return 'G';
}
```

c) The following fact() function is expected to compute the factorial of a positive number **n**.

```
long fact(int n)
{
  long k=1;
  do
    k*=fact(i--);
  while(n>0);
  return k;
}
```

Is the above code error free? If not, state your reason why it will not work and also give the corrected code.

2. Write a program to do the following
    a. In 'main' create a string (char array) of size 5 in the **heap using dynamic allocation**.
    b. Initialize the string as "IIIT"
    c. Create a **function 'vowelfilter'** which takes a string as input and creates a **new string in the heap** with only the vowels('A','E','I','O','U') from the input string. (For example: if we pass the input string "IIIT", the resultant string is "III".
    d. The **'vowelfilter' function returns a pointer** to the new string as the output.
    e. Call the function from main and print the output
    f. At the end of main, make sure to **free all memory** allocated in the heap.

[Note: you should **not** use any utility functions from **string.h**. You can easily achieve the necessary result by treating the strings as character arrays which terminates with '\0']

3. Please read the below programs carefully, analyze what it does and write the correct output in the blank space.
    a. Program - 1
    In the below code, if the contents of the test.txt files are as below, what is the output of the program?
    Total number of lines are: _____

    The contents of the file *test.txt* are as below

Hello friends, how are you? This is a sample file to get line numbers from the file. a

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

```
#define FILENAME "test.txt"
int main()
{
	FILE *fp;
	char ch;
	int linesCount=0;

	//open file in read more
	fp=fopen(FILENAME,"r");
	if(fp==NULL)
	{
		printf("File \"%s\" does not exist!!!\n",FILENAME);
		return -1;
	}

	//read character by character and check for new line
	while((ch=fgetc(fp))!=EOF)
	{
		if(ch=='\n')
			linesCount++;
	}

	//close the file
	fclose(fp);

	//print number of lines
	printf("Total number of lines are: %d\n",linesCount);

	return 0;
}
```

b. Program - 2

In the program given for question (a), if we want to open the file for both reading and writing, the mode we need to use is _____

c. Program - 3

In the below code, using the comments given in the code, fill in the blanks in the code?

```
#include< stdio.h >
int main()
```

```
{

   FILE *fp;   /* file pointer*/
   char fName[20];

   printf("Enter file name to create :");
   scanf("%s",fName);

   /*creating (open) a file, in “w”: write mode*/
   fp=fopen(fName,"w");
   /*check file created or not*/
   if(fp==____NULL_____)
   {
      printf("File does not created!!!");
      exit(0); /*exit from program*/
   }

   printf("File created successfully.");
   return 0;
}
```

d. Program - 4

The below C Program writes all the members of an array of structures to a file using, and it reads the array from the file and displays it on the screen.

```
#include <stdio.h>
struct student
{
  char name[50];
  int height;
};
int main(){
    struct student stud1[5], stud2[5];
    FILE *fptr;
    int i;
    fptr = fopen("file.txt","wb");
    for(i = 0; i < 5; ++i)
    {
        fflush(stdin);
        printf("Enter name: ");
        gets(stud1[i].name);
```

```
        printf("Enter height: ");
        scanf("%d", &stud1[i].height);
    }
    ____(____, ____, 1, fptr);
    fclose(fptr);
    fptr = fopen("file.txt", "rb");
    ____(____, ____, 1, fptr);
    for(i = 0; i < 5; ++i)
    {
        printf("Name: %s\nHeight: %d", stud2[i].name, stud2[i].height);
    }
    fclose(fptr);
}
```