Lab 1

Q. Check the output of program below

```
#include<stdio.h>
int add(int, int);                    //function declaration
void main()
{
   int a, b, sum;
   printf("Enter two different numbers: ");
   scanf("%d%d",&a, &b);
   sum = add(a, b);                   //function call with arguments and return value
   printf("The sum of %d and %d is %d", a, b, sum);
}
//function definition
int add(int x, int y)
{
   return (x+y);
}
```

Q. Check the output of program below

```
#include<stdio.h>
//function definition
float centigradeToFahrenheit(float centi)
{
   float f;
   f = 9.0/5*centi+32;
   return f;
}
void main()
{
   float c, f;
   printf("Enter temperature in centigrade: ");
   scanf("%f",&c);
   f = centigradeToFahrenheit(c);   //function call with argument and return type
   printf("The equivalent temperature in Fahrenheit is : %.2f",f);
}
```

Q. Check the output of program below

```
#include<stdio.h>
#define PI 3.1428
void areaOfCircle(float r)
{
   float area;
   area = PI * r * r;
   printf("Area of the circle is : %.4f\n", area);
```

```
}
void main()
{
   float r1, r2;
   printf("Enter radius of first circle: ");
   scanf("%f",&r1);
   areaOfCircle(r1);
   printf("Enter radius of second circle: ");
   scanf("%f",&r2);
   areaOfCircle(r2);
}
```

Q. Check the output of program below

```
#include<stdio.h>
void swap()
{
   int num1, num2, temp;
   printf("Enter first number : ");
   scanf("%d", &num1);
   printf("Enter second number : ");
   scanf("%d", &num2);
   printf("Before swap\nNumber 1 : %d\nNumber 2 : %d\n",num1, num2);
   temp = num1;
   num1 = num2;
   num2 = temp;
   printf("After swap\nNumber 1 : %d\nNumber 2 : %d\n",num1, num2);
}
void main()
{
   swap();
}
```

Practice question

1. WAP which receive **float** and **int** number as argument from **main()** function, find their product and return result.
2. WAP to calculate simple interest and compound interest making two different functions.
3. WAP to find cubes and squares of first 10 natural numbers using function.
4. WAP to reverse digits of number entered by user using function.
5. WAP to display all prime numbers between n1 and n2 where n1<n2, using function.

Lab 2

Q. Check the output of program below (swap using call by reference)

```
#include<stdio.h>
void swap(int *num1, int *num2)
{
   int temp;
   temp = *num1;
   *num1 = *num2;
   *num2 = temp;
}
void main()
{
   int num1, num2;
   printf("Enter first number : ");
   scanf("%d", &num1);
   printf("Enter second number : ");
   scanf("%d", &num2);
   printf("Before swap\nNumber 1 : %d\nNumber 2 : %d\n",num1, num2);
   swap(&num1, &num2);
   printf("After swap\nNumber 1 : %d\nNumber 2 : %d\n",num1, num2);
}
```

Q. Check the output of program below (factorial using recursive function)

```
#include<stdio.h>
long int factorial(int n)
{
   if(n==1)
   {
      return (1);
   }
   else
   {
      return (n*factorial(n-1));
   }
}
void main()
{
   int num;
   printf("Enter  number : ");
   scanf("%d", &num);
   printf("Factorial of %d is %ld",num, factorial(num));
}
```

Q. Check the output of program below (Fibonacci series using recursive function)

```
#include<stdio.h>
int fibo(int n)
{
   if(n==1)
   {
      return 0;
   }
   else if(n==2)
   {
      return 1;
   }
   else
   {
      return (fibo(n-1) + fibo(n-2));
   }
}
void main()
{
   int terms,i;
   printf("How many terms do you need?\t");
   scanf("%d", &terms);
   for(i=1; i<=terms; i++)
   {
      printf("%d\t",fibo(i));
   }
}
```

Q. Check the output of program below (Automatic and External)

```
#include<stdio.h>
int a = 10;
int b = 3;
void fun()
{
   printf("Global variable a from user-define function: %d\n", a);
   printf("Global variable b from user-define function: %d\n", b);
   a = 20;
   printf("local variable from user-define function: %d\n", a);
}
void main()
{
   printf("Global variable a from main function: %d\n", a);
   printf("Global variable b from main function: %d\n", b);
   a = 25;
   printf("local variable from main function: %d\n", a);
   fun();
}
```

Practice question

1. WAP to find out largest number among three number using call by reference.
2. WAP to read number from user and calculate sum first n natural number using recursive function.
3. WAP to find nth term Fibonacci series using recursive function.
4. WAP to calculate sum of digits of number using recursion.

Lab 3

Q. Check the output of program below

```
#include<stdio.h>
void main()
{
   int a=20;
   printf("The address of a is : %u", &a);
   printf("\nThe value of a is : %d", a);
}
```

Q. Check the output of program below

```
#include<stdio.h>
void main()
{
   int a=10, *p;
   p = &a;
   printf("Address of a is %u",&a);
   printf("Address of a is %u",p);
   printf("Value of a is %d",a);
   printf("Value of a is %d",*p);
   printf("Address of p is %u",&p);
}
```

Q. Check the output of program below

```
#include <stdio.h>
void main() {
   int a = 5, b = 15;
   int *p1, *p2;
   p1 = &a;
   p2 = &b;
   printf("Before assignment: *p1 = %d, *p2 = %d\n", *p1, *p2);
   p1 = p2;
   printf("After assignment: *p1 = %d, *p2 = %d\n", *p1, *p2);
}
```

Q. Check the output of program below

```
#include <stdio.h>
void addTen(int *num) {
   *num = *num + 10;
   printf("Inside function: %d\n", *num);
}
int main() {
   int a = 5;
   addTen(&a);
   printf("Outside function: %d\n", a);
   return 0;
}
```

Q. Check the output of program below (upper-lower alphabet)

```
#include <stdio.h>
void conversion(char *);
int main()
{
   char input;
   printf("Enter character of your choice: ");
   scanf("%c", &input);
   conversion(&input);
   printf("The corresponding character is : %c",input);
   return 0;
}
void conversion(char *c)
{
   if(*c >= 97 && *c <= 122)
   {
      *c = *c - 32;
   }
   else
   {
      *c = *c + 32;
   }
}
```

Practice question

1. Make the upper-lower alphabet program work only for alphabet input.
2. WAP to store resultant of two number addition using call by reference. Also display address of resultant.

Lab 4

Q1. Structure implementation

```
#include <stdio.h>
void main()
{
   struct student
   {
      char name[20];
      int rollno;
      float marks;
   };
   struct student st2 = {"John", 25, 89.90};
   printf("Accessing member of structure from structure variable.\n");
   printf("Name : %s\n",st2.name);
   printf("Roll no : %d\n",st2.rollno);
   printf("Marks : %.2f\n",st2.marks);
}
```

Q2. Structure implementation

```
#include <stdio.h>
void main()
{
   struct student
   {
      char name[20];
      int rollno;
      float marks;
   };
   struct student st;
   printf("Enter name: ");
   scanf("%s",st.name);
   printf("Enter roll no: ");
   scanf("%d", &st.rollno);
   printf("Enter marks: ");
   scanf("%f", &st.marks);
   printf("\nAccessing member of structure from structure variable.\n");
   printf("name : %s\n",st.name);
   printf("name : %d\n",st.rollno);
   printf("name : %.2f\n",st.marks);
}
```

Q3. Array of Structure implementation

```
#include <stdio.h>
void main()
{
   struct student
   {
      char name[20];
      int rollno;
      float marks;
   };
   struct student st[5];
   for(int i=0; i<5; i++)
   {
      printf("\nEnter information of student no %d\n", i+1);
      printf("Enter name: ");
      scanf("%s",st[i].name);
      printf("Enter roll no: ");
      scanf("%d", &st[i].rollno);
      printf("Enter marks: ");
      scanf("%f", &st[i].marks);
   }
   printf("\nDetail Information of Students\n");
   printf("Name\t\t\tRoll No\t\t\tMarks");
   printf("\n_____\n");
   for(int j=0; j<5; j++)
   {
      printf("%s\t\t%d\t\t%.2f\n", st[j].name, st[j].rollno, st[j].marks);
   }
}
```

Q4. Nested Structure implementation

```
#include <stdio.h>
void main()
{
   struct date
   {
      int day;
      int month;
      int year;
   };
   struct student
   {
      char name[20];
      int rollno;
      float marks;
```

```
    struct date dateOfBirth;
  };
  struct student st;
  struct date d;
  printf("\nEnter information of student\n");
  printf("Enter name: ");
  scanf("%s",st.name);
  printf("Enter roll no: ");
  scanf("%d", &st.rollno);
  printf("Enter marks: ");
  scanf("%f", &st.marks);
  printf("Enter day: ");
  scanf("%d", &st.dateOfBirth.day);
  printf("Enter month: ");
  scanf("%d", &st.dateOfBirth.month);
  printf("Enter year: ");
  scanf("%d", &st.dateOfBirth.year);
  printf("\nDetail Information of Students\n");
  printf("Name\t\t\tRoll No\t\t\tMarks\t\tDate of Birth");
  printf("\n_____\n");

  printf("%s\t\t\t%d\t\t\t%.2f\t\t%d-%d-%d", st.name, st.rollno, st.marks, st.dateOfBirth.day,
        st.dateOfBirth.month, st.dateOfBirth.year);
  printf("\n\nAccessing day, month and year from date structure variable\n");
  printf("Day : %d", d.day);
  printf("\nMonth : %d", d.month);
  printf("\nYear : %d", d.year);
}
```

Q5. Pointer to Structure implementation

```
#include <stdio.h>
void main()
{
  struct student
  {
    char name[20];
    int rollno;
    float marks;
  };
  struct student st, *stptr;
  printf("\nEnter information of student\n");
  printf("Enter name: ");
  scanf("%s",st.name);
  printf("Enter roll no: ");
  scanf("%d", &st.rollno);
  printf("Enter marks: ");
```

```
    scanf("%f", &st.marks);
    stptr = &st;
    printf("\n\nAccessing Student information using arrow operator\n");
    printf("Name\t\t\tRoll No\t\t\tMarks");
    printf("\n_____\n");
    printf("%s\t\t\t%d\t\t\t%.2f", stptr->name, stptr->rollno, stptr->marks);
    printf("\n\nAccessing Student information using * operator\n");
    printf("Name\t\t\tRoll No\t\t\tMarks");
    printf("\n_____\n");
    printf("%s\t\t\t%d\t\t\t%.2f", (*stptr).name, (*stptr).rollno, (*stptr).marks);
}
```

Q6. Passing structure member to a function

```
#include <stdio.h>
void display(char name[], int roll, float marks)
{
    printf("\nDetail Information of Students\n");
    printf("Name\t\t\tRoll No\t\t\tMarks");
    printf("\n_____\n");
   printf("%s\t\t\t%d\t\t\t%.2f", name, roll, marks);
}
void main()
{
    struct student
    {
        char name[20];
        int rollno;
        float marks;
    };
    struct student st;
    printf("\nEnter information of student\n");
    printf("Enter name: ");
    scanf("%s",st.name);
    printf("Enter roll no: ");
    scanf("%d", &st.rollno);
    printf("Enter marks: ");
    scanf("%f", &st.marks);
    display(st.name, st.rollno, st.marks);
}
```

Q7. Passing entire structure to a function

```
#include <stdio.h>
struct student
    {
        char name[20];
```

```
    int rollno;
    float marks;
  };
void display(struct student stf)
{
  printf("\nDetail Information of Students\n");
  printf("Name\t\t\tRoll No\t\t\tMarks");
  printf("\n_____\n");
  printf("%s\t\t\t%d\t\t\t%.2f", stf.name, stf.rollno, stf.marks);
}
void main()
{
  struct student st;
  printf("\nEnter information of student\n");
  printf("Enter name: ");
  scanf("%s",st.name);
  printf("Enter roll no: ");
  scanf("%d", &st.rollno);
  printf("Enter marks: ");
  scanf("%f", &st.marks);
  display(st);
}
```

Q8. Passing structure pointer to a function

```
#include <stdio.h>
struct student
  {
    char name[20];
    int rollno;
    float marks;
  };
void display(struct student *stf)
{
  printf("\nDetail Information of Students\n");
  printf("Name\t\t\tRoll No\t\t\tMarks");
  printf("\n_____\n");
  printf("%s\t\t\t%d\t\t\t%.2f", (*stf).name, (*stf).rollno, (*stf).marks);
}
void main()
{
  struct student st;
  printf("\nEnter information of student\n");
  printf("Enter name: ");
  scanf("%s",st.name);
  printf("Enter roll no: ");
  scanf("%d", &st.rollno);
```

```
   printf("Enter marks: ");
   scanf("%f", &st.marks);
   display(&st);
}
```

Q9. Passing array of structure to a function

```
#include <stdio.h>
struct student
   {
      char name[20];
      int rollno;
      float marks;
   };
void display(struct student stf[])
{
  printf("\nDetail Information of Students\n");
   printf("Name\t\t\tRoll No\t\t\tMarks");
   printf("\n_____\n");
   for(int j=0; j<5; j++)
   {
      printf("%s\t\t%d\t\t%.2f\n", stf[j].name, stf[j].rollno, stf[j].marks);
   }
}
void main()
{
   struct student st[5];
   for(int i=0; i<5; i++)
   {
      printf("\nEnter information of student no %d\n", i+1);
      printf("Enter name: ");
      scanf("%s",st[i].name);
      printf("Enter roll no: ");
      scanf("%d", &st[i].rollno);
      printf("Enter marks: ");
      scanf("%f", &st[i].marks);
   }
   display(st);
}
```

Q10. Union Implementation

```
#include <stdio.h>
void main()
{
   union student
   {
```

```
    int roll;
    float marks;
  };
  union student st;
  st.roll = 12;
  printf("\nRoll no: %d", st.roll);
  st.marks = 89.5;
  printf("\nMarks: %.2f", st.marks);

  st.roll = 8;
  st.marks = 60.5;
  printf("\nRoll no: %d", st.roll);
  printf("\nMarks: %.2f", st.marks);
}
```

Practice question

1. Add structure member “Remarks” in Q3 and display remarks too.
2. Implement array in Q4.
3. Create a structure named **book** which has members **name, pages, and price.** Write a program to read name, no of pages and price of 5 books. Display all information of books.
4. Create structure name **Employee** with members **name, address, salary, dateJoin.** WAP to read information of 5 employee and display employee information using function.
5. Create structure name **Employee** with members **name, address, salary, dateJoin.** WAP to read information of 5 employee and display employee detail with third highest salary.
6. Create structure name **Employee** with members **name, address, salary, dateJoin.** WAP to read information of 5 employee and display employee detail in ascending order of their name.
7. Explain output of Q10.

Lab 5 : File Handling in C

Q1. Implementation of gets() and puts()

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
   FILE *fptr;
   char s[100];
   fptr = fopen("abc.txt","w");
   if(fptr == NULL)
   {
      printf("File can not be created");
      exit(1);
   }
   else
   {
      printf("File has been successfully created");
   }

   fputs("File handling in C.", fptr);
   fclose(fptr);
   printf("\nOpen File closed\n");

   fptr = fopen("abc.txt","r");
   if (fptr == NULL)
   {
      printf("Error opening file for reading\n");
      exit(1);
   }
   fgets(s,100,fptr);
   printf("The text from file is:\t %s", s);
   fclose(fptr);
}
```

Q2. Implementation of fgetc() and fputc()

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
   FILE *fptr;
   char c, filename[20];
   printf("Enter file name with extension (.txt): ");
   scanf("%s", &filename);
   fptr = fopen(filename,"w");
   if(fptr == NULL)
```

```
    {
       printf("File can not be created");
       exit(1);
    }
    while ((c = getchar()) != EOF)
    {
       fputc(c, fptr);
    }
    fclose(fptr);
    printf("\nOpen File closed\n");

    fptr = fopen(filename,"r");
    if (fptr == NULL)
    {
       printf("Error opening file for reading\n");
       exit(1);
    }
    while ((c = fgetc(fptr)) != EOF)
    {
       putchar(c);
    }
    fclose(fptr);
}
```

Q3. Implementation of getw() and putw()

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
   FILE *fptr;
   int num, i;
   fptr = fopen("numbers.dat", "wb");
   if (fptr == NULL) {
      printf("Error opening file for writing\n");
      exit(1);
   }
   for (i = 1; i <= 5; i++) {
      printf("Writing %d to file\n", i);
      putw(i, fptr);
   }
   fclose(fptr);
   fptr = fopen("numbers.dat", "rb");
   if (fptr == NULL) {
      printf("Error opening file for reading\n");
      exit(1);
   }
```

```
    printf("Reading numbers from file:\n");
    while ((num = getw(fptr)) != EOF) {
        printf("%d\n", num);
    }
    fclose(fptr);
    return 0;
}
```

Q4. Implementation of fprintf()

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *fptr;
    int num = 42;
    float pi = 3.14159;
    char str[] = "Hello, World!";
    fptr = fopen("implement_fprintf.txt", "w");
    if (fptr == NULL) {
        printf("Error opening file in write mode\n");
        exit(1);
    }
    fprintf(fptr, "Integer: %d\n", num);
    fprintf(fptr, "Float: %.2f\n", pi);
    fprintf(fptr, "String: %s\n", str);
    fclose(fptr);
    fptr = fopen("abc.txt", "a");
    if (fptr == NULL) {
        printf("Error opening file in append mode\n");
        exit(1);
    }
    fprintf(fptr, "Appending more data...\n");
    fprintf(fptr, "New Integer: %d\n", num * 2);
    fprintf(fptr, "New Float: %.2f\n", pi * 2);
    fprintf(fptr, "New String: %s\n", "Appended Text");
    fclose(fptr);
    return 0;
}
```

Q5. Implementation of fscanf()

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

```
    FILE *fptr;
    int num;
    float pi;
    char str[50];
    fptr = fopen("implement_fprintf.txt", "r");
    if (fptr == NULL) {
        printf("Error opening file\n");
        exit(1);
    }
    // Read data from file using fscanf
    fscanf(fptr, "Integer: %d\n", &num);
    fscanf(fptr, "Float: %f\n", &pi);
    fscanf(fptr, "String: %[^\n]\n", str);
    fclose(fptr);
    printf("Read from file:\n");
    printf("Integer: %d\n", num);
    printf("Float: %.2f\n", pi);
    printf("String: %s\n", str);
    return 0;
}
```

Q6. Writing structure data into file

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    struct student
    {
        char name[30];
        int roll;
        float marks;
    };
    struct student s[3];
    int i;
    FILE *fp;
    fp = fopen("student.txt", "w");
    if(fp == NULL)
    {
        printf("File cannot be created...");
        exit(1);
    }
    for(i = 0; i < 3; i++)
    {
        printf("\nEnter information of student no %d\n", i+1);
        printf("Name: ");
        scanf("%s", s[i].name);
```

```
        printf("Roll No.: ");
        scanf("%d", &s[i].roll);
        printf("Marks: ");
        scanf("%f", &s[i].marks);
        fprintf(fp, "%s %d %.2f\n", s[i].name, s[i].roll, s[i].marks);
    }
    fclose(fp);
    fp = fopen("student.txt", "r");
    if(fp == NULL)
    {
        printf("Error opening file for reading...");
        exit(1);
    }
    struct student st;
    printf("\nReading student information from file...\n");
    printf("Name\t\tRoll\t\tMarks\n");
    printf("---------------------------------------\n");
    while(fscanf(fp, "%s %d %f", st.name, &st.roll, &st.marks) != EOF)
    {
        printf("%s\t\t%d\t\t%.2f\n", st.name, st.roll, st.marks);
    }
    fclose(fp);
    return 0;
}
```

Q7. Implementation of fwrite() and fread()

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    struct student
    {
        char name[30];
        int roll;
        float marks;
    };
    struct student s[3], st[3];
    int i;
    FILE *fp;
    fp = fopen("student1.txt", "wb+");
    if(fp == NULL)
    {
        printf("File cannot be created...");
        exit(1);
    }
    for(i = 0; i < 3; i++)
    {
```

```
        printf("\nEnter information of student no %d\n", i+1);
        printf("Name: ");
        scanf("%s", s[i].name);
        printf("Roll No.: ");
        scanf("%d", &s[i].roll);
        printf("Marks: ");
        scanf("%f", &s[i].marks);
    }
    printf("\nWriting information to file...\n");
    fwrite(s, sizeof(struct student), 3, fp);
    // Rewind the file pointer to the beginning for reading
    rewind(fp);
    printf("\nReading student information from file...\n");
    fread(st, sizeof(struct student), 3, fp);

    printf("Name\t\tRoll\t\tMarks\n");
    printf("---------------------------------------\n");
    for(i = 0; i < 3; i++)
    {
        printf("%s\t\t%d\t\t%.2f\n", st[i].name, st[i].roll, st[i].marks);
    }
    fclose(fp);
    return 0;
}
```

Practice question

1. Append “Class 12 computer science” in **abc.txt**.
2. WAP to enter name and address of students and store then in data file “student.dat”.
3. WAP to create structure named employee with member **empName, age** and **salaray.** Write information of 5 employee in data file “employee.dat”.
4. Append 5 more employee information in data file “employee.dat”.