

## **Lab-1: Programming in C**

(6 Hours - 4 credits)

### **Section A**

1. Write a C Program to find the sum of digits.
2. Write a C Program to check whether a given number is Armstrong or not
3. Write a C Program to check whether a given number is Prime or not
4. Write a C Program to generate the Fibonacci series
5. Write a C Program to display the given number is Adam number or not
6. Write a C Program to print reverse of the given number and string
7. Write a C Program to find minimum and maximum of 'n' numbers using array
8. Write a C Program to arrange the given number in ascending order
9. Write a C Program to add, subtract and multiply two matrices
10. Write a C Program to calculate NCR and NPR

### **Section B**

1. Write a C Program to find the grade of a student using else if ladder
2. Write a C Program to implement the various string handling function
3. Write a C Program to create an integer file and displaying the even numbers only
4. Write a C Program to calculate quadratic equation using switch-case
5. Write a C Program to generate student mark list using array of structures
6. Write a C Program to create and process the student mark list using file
7. Write a C Program to create and process pay bill using file
8. Write a C Program to create and process inventory control using file
9. Write a C Program to create and process electricity bill using file

//1. **Sum of digits**

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

```
int main()
```

```
{  
    int n, t, sum = 0, remainder;  
    printf("Enter an integer\n");  
    scanf("%d", &n);  
    t = n;  
    while (t != 0)  
    {  
        remainder = t % 10;  
        sum = sum + remainder;  
        t = t / 10;  
    }  
    printf("Sum of digits of %d = %d\n", n, sum);  
    return 0;  
}
```

## // 2. Armstrong number

```
#include <stdio.h>
int main()
{
    int number, originalNumber, remainder, result = 0;
    printf("Enter a three digit integer: ");
    scanf("%d", &number);
    originalNumber = number;
    while (originalNumber != 0)
    {
        remainder = originalNumber%10;
        result += remainder*remainder*remainder;
        originalNumber /= 10;
    }
    if(result == number)
        printf("%d is an Armstrong number.",number);
    else
        printf("%d is not an Armstrong number.",number);
    return 0;
}
```

### //3. Prime number

```
#include <stdio.h>
int main()
{
    int n, i, flag = 0;

    printf("Enter a positive integer: ");
    scanf("%d",&n);

    for(i=2; i<=n/2; ++i)
    {
        // condition for nonprime number
        if(n%i==0)
        {
            flag=1;
            break;
        }
    }

    if (flag==0)
        printf("%d is a prime number.",n);
    else
        printf("%d is not a prime number.",n);

    return 0;
}
```

#### // 4. **Fibonacci Series**

```
#include <stdio.h>
int main()
{
    int i, n, t1 = 0, t2 = 1, nextTerm;

    printf("Enter the number of terms: ");
    scanf("%d", &n);

    printf("Fibonacci Series: ");

    for (i = 1; i <= n; ++i)
    {
        printf("%d, ", t1);
        nextTerm = t1 + t2;
        t1 = t2;
        t2 = nextTerm;
    }
    return 0;
}
```

```
//5. Adam number
#include<stdio.h>
#include<math.h>
using namespace std;
int main ()
{
    int num, temp, r1, r2, sq, rev1 = 0, rev2 = 0;
    printf("Enter a number : ");
    scanf("%d", &num);
    temp = num * num;
    while (temp != 0)
    {
        r1 = temp % 10;
        rev1 = rev1 * 10 + r1;
        temp = temp / 10;
    }
    sq = sqrt(rev1);
    while (sq != 0)
    {
        r2 = sq % 10;
        rev2 = rev2 * 10 + r2;
        sq = sq / 10;
    }
    if (rev2 == num)
    printf ("\n is an Adam number.");
    else
        printf ("\n is an not Adam number.");
    return 0;
}
```

## //6. Reversed Number

```
#include <stdio.h>
int main()
{
    int n, reversedNumber = 0, remainder;

    printf("Enter an integer: ");
    scanf("%d", &n);

    while(n != 0)
    {
        remainder = n%10;
        reversedNumber = reversedNumber*10 + remainder;
        n /= 10;
    }

    printf("Reversed Number = %d", reversedNumber);
    return 0;
}
```

// 7. **Find maximum and minimum in all array elements.**

```
#include <stdio.h>
#define MAX_SIZE 100 // Maximum array size
int main()
{
    int arr[MAX_SIZE];
    int i, max, min, size;
    /* Input size of the array */
    printf("Enter size of the array: ");
    scanf("%d", &size);

    /* Input array elements */
    printf("Enter elements in the array: ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }
    /* Assume first element as maximum and minimum */
    max = arr[0];
    min = arr[0];

    /*
     * Find maximum and minimum in all array elements.
     */
    for(i=1; i<size; i++)
    {
        /* If current element is greater than max */
        if(arr[i] > max)
        {
            max = arr[i];
        }
        /* If current element is smaller than min */
        if(arr[i] < min)
        {
            min = arr[i];
        }
    }
    /* Print maximum and minimum element */
    printf("Maximum element = %d\n", max);
    printf("Minimum element = %d", min);
    return 0;
}
```



```

/*
 * 8. C program to accept N numbers and arrange them in an ascending order
 */

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

    int i, j, a, n, number[30];
    printf("Enter the value of N \n");
    scanf("%d", &n);

    printf("Enter the numbers \n");
    for (i = 0; i < n; ++i)
        scanf("%d", &number[i]);

    for (i = 0; i < n; ++i)
    {
        for (j = i + 1; j < n; ++j)
        {
            if (number[i] > number[j])
            {
                a = number[i];
                number[i] = number[j];
                number[j] = a;
            }
        }
    }
    printf("The numbers arranged in ascending order are given below \n");
    for (i = 0; i < n; ++i)
        printf("%d\n", number[i]);
}

```

```
/* 9. MATRIX ADDITION, SUBTRACTION AND MULTIPLICATION */
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,j,c,r,k;
    int a[20][20],b[20][20],ma[20][20],ms[20][20];
    int mm[20][20];
    clrscr();
    printf("\n\t\tINPUT:");
    printf("\n\t\t-----");
    printf("\n\t\tEnter the value for row and column: ");
    scanf("%d%d",&c,&r);
    printf("\n\t\tEnter the value for matrix A\n");
    for(i=0;i<c;i++)
    {
        for(j=0;j<r;j++)
        {
            scanf("%d",&a[i][j]);
        }
        printf("\n");
    }
    printf("\n\t\tEnter the value for matrix B\n");
    for(i=0;i<c;i++)
    {
        for(j=0;j<r;j++)
        {
            scanf("%d",&b[i][j]);
        }
        printf("\n");
    }
    for(i=0;i<c;i++)
    {
        for(j=0;j<r;j++)
        {
            ma[i][j]=a[i][j]+b[i][j];
            ms[i][j]=a[i][j]-b[i][j];
        }
    }
    for(i=0;i<c;i++)
    {
        for(j=0;j<r;j++)
        {
            mm[i][j]=0;
            for(k=0;k<c;k++)
            {
                mm[i][j] +=a[i][k]*b[k][j];
            }
        }
    }
    printf("\n\t\tOUTPUT:");
```

```

printf("\n\t\tt-----");
printf("\n\t\tThe addition matrix is:\n");
for(i=0;i<c;i++)
{
    for(j=0;j<r;j++)
    {
        printf("\t\t%d",ma[i][j]);
    }
    printf("\n");
}
printf("\n\t\tThe subtraction matrix is:\n");
for(i=0;i<c;i++)
{
    for(j=0;j<r;j++)
    {
        printf("\t\t%d",ms[i][j]);
    }
    printf("\n");
}
printf("\n\t\tThe multiplication matrix is:\n");
for(i=0;i<c;i++)
{
    for(j=0;j<r;j++)
    {
        printf("\t\t%d",mm[i][j]);
    }
    printf("\n");
}
getch();
}

```

## //10. NCR - NPR

```
#include <stdio.h>
long factorial(int);
long find_ncr(int, int);
long find_npr(int, int);
int main()
{
    int n, r;
    long ncr, npr;
    printf("Enter the value of n and r\n");
    scanf("%d%d",&n,&r);
    ncr = find_ncr(n, r);
    npr = find_npr(n, r);
    printf("%dC%d = %ld\n", n, r, ncr);
    printf("%dP%d = %ld\n", n, r, npr);
    return 0;
}
long find_ncr(int n, int r) {
    long result;
    result = factorial(n)/(factorial(r)*factorial(n-r));
    return result;
}
long find_npr(int n, int r) {
    long result;
    result = factorial(n)/factorial(n-r);
    return result;
}
long factorial(int n) {
    int c;
    long result = 1;
    for (c = 1; c <= n; c++)
        result = result*c;
    return result;
}
```

## // 11. Program using else if ladder

```
#include<stdio.h>
#include<conio.h>
void main()
{
int s1,s2,s3,s4,s5,t,p;
clrscr();

printf("\n Enter marks of 5 subjects each out of 100 ");
printf("\n\n *****");
printf("\n\n Maths = ");
scanf("%d",&s1);
printf("\n Science = ");
scanf("%d",&s2);
printf("\n English = ");
scanf("%d",&s3);
printf("\n History = ");
scanf("%d",&s4);
printf("\n Geography = ");
scanf("%d",&s5);
printf("\n *****");
t=s1+s2+s3+s4+s5; //Total
printf("\n Total marks = %d/500",t);

p=t/5; //Percentage
printf("\n\n Percentage = %d%",p);
printf("\n *****");
////////// Ladder If Statement //////////
if(p>=80)
printf("\n\n Your Grade : A+");

else if(p>=75)
printf("\n\n Your Grade : A");

else if(p>=60)
printf("\n\n Your Grade : B");

else if(p>=45)
printf("\n\n Your Grade : C");

else if(p>=35)
printf("\n\n Your grade : D");

else
printf("\n\n You Are Fail");

////////// Ladder If Statement //////////

getch();
}
```

## // 12. String manipulation functions

```
#include <string.h>
main()
```

```

{ char s1[20], s2[20], s3[20];

int x, l1, l2, l3;
printf("\n\nEnter two string constants \n");
printf("?");
scanf("%s %s", s1, s2);

/* comparing s1 and s2 */

x = strcmp(s1, s2);
if(x != 0)
{ printf("\n\nStrings are not equal \n");
  strcat(s1, s2); /* joining s1 and s2 */
}
else
  printf("\n\nStrings are equal \n");
/* copying s1 to s3

  strcpy(s3, s1);

/* Finding length of strings */

l1 = strlen(s1);
l2 = strlen(s2);
l3 = strlen(s3);
/* output */

printf("\ns1 = %s\t length = %d characters\n", s1, l1);
printf("s2 = %s\t length = %d characters\n", s2, l2);
printf("s3 = %s\t length = %d characters\n", s3, l3);
}

```

### //13. HANDLING OF INTEGER DATA FILES

```
#include <stdio.h>
main()
{
    FILE *f1, *f2, *f3;
    int number, i;

    printf("Contents of DATA file\n\n");
    f1 = fopen("DATA", "w");    /* Create DATA file */
    for(i = 1; i <= 30; i++)
    {
        scanf("%d", &number);
        if(number == -1) break;
        putw(number, f1);
    }
    fclose(f1);

    f1 = fopen("DATA", "r");
    f2 = fopen("ODD", "w");
    f3 = fopen("EVEN", "w");

    /* Read from DATA file */
    while((number = getw(f1)) != EOF)
    {
        if(number %2 == 0)
            putw(number, f3); /* Write to EVEN file */
        else
            putw(number, f2); /* Write to ODD file */
    }
    fclose(f1);
    fclose(f2);
    fclose(f3);

    f2 = fopen("ODD", "r");
    f3 = fopen("EVEN", "r");
    printf("\n\nContents of ODD file\n\n");

    while((number = getw(f2)) != EOF)
        printf("%4d", number);
    printf("\n\nContents of EVEN file\n\n");

    while((number = getw(f3)) != EOF)
        printf("%4d", number);

    fclose(f2);
    fclose(f3);
}
```

**/\*14. Program to find the roots of a quadratic equation using switch statement \*/**

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

{
    int flag;
    float x, x1, x2;
    float a, b, c, d;
    float rpart, ipart;
    clrscr();
    printf("\n Enter 3 numbers: ");
    scanf("%f %f %f", &a, &b, &c);
    if(a==0)
    {
        x=-b/c;
        printf("\n Only root x : %7.3f", x);
        exit(); }
    d=b*b-4*a*c;
    if(d>0)
        flag=1;
    else if(d==0)
        flag=2;
    else
        flag=3;
    switch(flag)
    {
        case 1:
            printf("\n Real & Distinct roots are: ");
            x1=(-b+sqrt(d))/(2*a);
            x2=(-b-sqrt(d))/(2*a);
            printf("\n x1=%7.3f \n x2=%7.3f", x1, x2);
            break;
        case 2:
            printf("\n Repeated roots are: ");
            x1=-b/(2*a);
            x2=x1;
            printf("\n x1 & x1 : %7.3f", x1);
            break;
        case 3:
            d=sqrt(abs(d));
            rpart=-b/(2*a);
            ipart=d/(2*a);
            printf("\n Complex Roots are: ");
            printf("\n x1=%7.3f+i%7.3f", rpart, ipart);
            printf("\n x2=%7.3f-i%7.3f", rpart, ipart);
    }
    getch();
    return; }
```



### //15. students marks list using array of structure

```
#include<stdio.h>
#include<conio.h>
int k=0;
struct stud
{
    int rn;
    char name[30];
    int m1,m2,m3,total;
    float avg;
    char grade,result;
}s[30];
void main()
{
    int no,roll=101,i;
    clrscr();
    printf("Enter No of Students : ");
    scanf("%d",&no);
    for(i=0;i<no;i++)
    {
        clrscr();
        s[k].rn=roll;
        printf("\nEnter the Student Roll Number : %d ",s[k].rn);
        printf("\nEnter the Student Name :");
        fflush(stdin);
        gets(s[k].name);
        printf("\nEnter the Three Marks : ");
        scanf("%d",&s[k].m1);
        scanf("%d",&s[k].m2);
        scanf("%d",&s[k].m3);
        if(s[k].m1>=35 && s[k].m2>=35 && s[k].m3>=35)
        {
            s[k].result='P';
        }
        else
        {
            s[k].result = 'F';
        }
        s[k].total = s[k].m1+s[k].m2+s[k].m3;
        printf("The Total is : %d",s[k].total);
        s[k].avg=s[k].total/3;
        if(s[k].avg>=60)
        {
            if(s[k].result == 'P')
            {
                s[k].grade = 'A';
            }
            else
            {
                s[k].grade = 'N';
            }
        }
        else if(s[k].avg>=50)
        {
            if(s[k].result == 'P')
```

```

        {
            s[k].grade = 'B';
        }
        else
        {
            s[k].grade = 'N';
        }
    }
else if(s[k].avg>=35)
{
    if(s[k].result == 'P')
    {
        s[k].grade = 'C';
    }
    else
    {
        s[k].grade = 'N';
    }
}

    getch();
    k++;
    roll++;
}

printf("\n*****");
printf("\n      STUDENT MARKLIST ");
printf("\n*****");
printf("\nROLL \tNAME  MARK1 MARK2 MARK3 TOTAL RESULT Average GRADE");
for(i=0;i<no;i++)
{
    printf("\n%d\t%s  %d  %d  %d  %d  %c  %0.1f%c",
s[i].rn,s[i].name,s[i].m1,s[i].m2,s[i].m3,s[i].total,s[i].result,s[i].avg,s[i].grade);
}
getch();
}

```

## // 16. Student data file creation

```
#include <stdio.h>
int main() {
    char name[50];
    int marks,i,n;
    printf("Enter number of students: ");
    scanf("%d",&n);
    FILE *fptr;
    fptr=(fopen("C:\\student.txt","w"));
    if(fptr==NULL) {
        printf("Error!");
        exit(1);
    }
    for (i=0;i<n;++i) {
        printf("For student%d\nEnter name: ",i+1);
        scanf("%s",name);
        printf("Enter marks: ");
        scanf("%d",&marks);
        fprintf(fptr,"\nName: %s \nMarks=%d \n",name,marks);
    }
    fclose(fptr);
    return 0;
}
```

## // 17. EMPLOYEE FILE PROGRAM

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

```
struct Employee
```

```
{  
    int Num;  
    char name[50];  
    float salary;  
};
```

```
int main()
```

```
{  
    struct Employee Emp;  
    FILE *fp;  
    int x,i,hra,da,cca,fpay;  
    printf("Enter the Number of Employees\n");  
    scanf("%d",&x);  
    fp = fopen("abc.txt","w");  
    fflush(stdin);  
    for(i=1;i<=x;i++)  
    {  
        printf("\nEnter Employee Number.\n");  
        scanf("%d",&Emp.Num);  
        printf("Enter Employee Name.\n");  
        scanf("%s",&Emp.name);  
        printf("Enter Employee Salary.\n");  
        scanf("%f",&Emp.salary);  
        fwrite(&Emp,sizeof(Emp),1,fp);  
    }  
    fclose(fp);  
    fp = fopen("abc.txt","r");  
    for(i=1;i<=x;i++)  
    {  
        fread(&Emp, sizeof(Emp),1,fp);  
        printf("\nEmployee Details\nNumber : %d\nName : %s\nSalary :  
%f\n",Emp.Num,Emp.name,Emp.salary);  
        hra = 0.15*(Emp.salary);  
        da = 0.55*(Emp.salary);  
        cca = 0.05*(Emp.salary);  
        fpay = Emp.salary + hra + da + cca;  
        printf("HRA : %d\nDA : %d\nCCA : %d\nFinal Salary : %d\n",hra,da,cca,fpay);  
    }  
    fclose(fp);  
}
```

## /\* 18. SAMPLE INVENTORY PROGRAM

\* C program to display the inventory of items in a store / shop  
\* The inventory maintains details such as name, price, quantity  
\* and manufacturing date of each item.  
\*/

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

```
void main()
```

```
{  
    struct date  
    {  
        int day;  
        int month;  
        int year;  
    };  
    struct details  
    {  
        char name[20];  
        int price;  
        int code;  
        int qty;  
        struct date mfg;  
    };  
    struct details item[50];  
    int n, i;  
  
    printf("Enter number of items:");  
    scanf("%d", &n);  
    fflush(stdin);  
    for (i = 0; i < n; i++)  
    {  
        fflush(stdin);  
        printf("Item name: \n");  
        scanf("%s", item[i].name);  
        fflush(stdin);  
        printf("Item code: \n");  
        scanf("%d", &item[i].code);  
        fflush(stdin);  
        printf("Quantity: \n");  
        scanf("%d", &item[i].qty);  
        fflush(stdin);  
        printf("price: \n");  
        scanf("%d", &item[i].price);  
        fflush(stdin);  
        printf("Manufacturing date(dd-mm-yyyy): \n");  
        scanf("%d-%d-%d", &item[i].mfg.day,  
            &item[i].mfg.month, &item[i].mfg.year);  
    }  
    printf("          ***** INVENTORY ***** \n");  
    printf("-----  
-----\n");  
    printf("S.N.|  NAME      |  CODE  |  QUANTITY  |  PRICE  
| MFG.DATE \n");  
    printf("-----  
-----\n");
```

```
for (i = 0; i < n; i++)
    printf("%d %-15s %-d %-5d %-5d
    %d/%d/%d \n", i + 1, item[i].name, item[i].code, item[i].qty,
    item[i].price, item[i].mfg.day, item[i].mfg.month,
    item[i].mfg.year);
printf("-----
-----\n");
}
```

```

/**
 * 19. SAMPLE ELECTRICITY BILL PROGRAM
 */

#include <stdio.h>

int main()
{
    int unit;
    float amt, total_amt, sur_charge;

    /* Input unit consumed from user */
    printf("Enter total units consumed: ");
    scanf("%d", &unit);

    /* Calculate electricity bill according to given conditions */
    if(unit <= 50)
    {
        amt = unit * 0.50;
    }
    else if(unit <= 150)
    {
        amt = 25 + ((unit-50) * 0.75);
    }
    else if(unit <= 250)
    {
        amt = 100 + ((unit-150) * 1.20);
    }
    else
    {
        amt = 220 + ((unit-250) * 1.50);
    }

    /*
     * Calculate total electricity bill
     * after adding surcharge
     */
    sur_charge = amt * 0.20;
    total_amt = amt + sur_charge;

    printf("Electricity Bill = Rs. %.2f", total_amt);

    return 0;
}

```