COMPUTER PROGRAMMING PRACTICAL FILE

FCCS002

NAME : SUMIT

ROLL NO. : 2022UIT3121

BRANCH : IT-2

INDEX

S.No.	PROGRAMMES	SIGNATURE
1.	Write a C program to input 3 numbers and print their average.	
2.	Write a C program to enter the radius of circle/sphere and compute its (i) Perimeter (ii) Area and (iii) Volume	
3.	Write a program in C to show that Right shift effectively divides a number by 2 and a left shift effectively multiplies a number by 2	
4.	Write a C program to find the roots of an quadratic equation.	
5.	Write down a function in C to implement bitwise AND, OR, XOR and NOT operations	
6.	Given a n integer number write a program that displays the number as follows First line: All digits Second Line : All except first digit Third line: All except first two digits Last line : The last digit	
7.	Write a program to enter an integer and print the sum of the digits in the integer.	
8.	Write a C program to input an investment amount and compute its fixed deposit cumulative return after 10 years at the rate of interest of 7.75%.	
9.	Write A C program to compute the roots of a quadratic equation.	
10.	A company has categorized its employees at 4 different levels (from 1 to 4). For different employees at different levels the perks are as follows Level TA entertainment Allowance 1 7000 3000 2 6000 2000 3 5000 1500	

	4. 5000 1500 For Level 1 Basic salary is between Rs 40000 to 60000 and Tax rate is 10% For level 2 Basic Salary is between Rs 30000 to 40000 and Tax rate is 8% For level 3 Basic salary is between Rs 20000 to 30000 and Tax rate is 5% For Level 4 Basic Salary is between Rs 15000 to 20000 and tax rate is 0 Gross Salary is sum of Basic salary, Perks and HRA which is 25% of Basic Salary Tax is computed on Gross Salary. Net Salary is Gross salary- Income tax Write a Program that will read Employees name, Level and Basic pay and will print Gross salary, Tax and Net Salary. Use Switch-case statement and if statements	
11.	Given a number, write a program using while loop to reverse the digits of the number. For example number 12345 should be written as 54321.	
12.	Write a program to find the prime numbers between a range of numbers entered by the user.	
13.	Write a program to find the HCF of two integers entered by the user.	
14.	The numbers in the sequence 1 1 2 3 5 8 13 21..... are called Fibbonacci numbers. Write a program using do...while loop to calculate and print the first m Fibonacci numbers	
15.	Write a program to evaluate the following functions to 0.0001% accuracy $Sinx = x-x^3/3! +x5/5! - x7/7!+.....$	
16.	Write a C program to display following Pattern 1 121 12321 121 1	
17.	Given the two one dimensional arrays A and B of size 10 which are sorted in ascending order. Write a C program to merge them into single sorted array C that contains every item from arrays A and B in ascending order.	
18.	Write a program that will count the number of occurrences of a specified character in a given line of Text.	

19.	Write a program to enter two 3 x 3 matrices and find their a. sum b. Multiplication c. Transpose	
20.	Write a program that counts the number of vowels, consonants and digits in a given line of string.	
21.	Write a program that replaces a substring with another string in a given line of text.	
22.	Write a program that takes as input a line of text and counts the frequency of each digit and letter in the input. The program will treat an uppercase letter and its lowercase equivalent as the same letter; For example, E and e increment the same counter.	
23.	Write a program that takes as input maximum 100 numbers from user (+ve integers) and calculates (i)sum (ii) mean (iii)standard deviation and (iv) variance .	
24.	Write a C program to display following Pattern: 1 A B 2 3 4 C D E F 5 6 7 8 9	
25.	Write a program that reads the cost of an item in the form RRR.PP(Where RRRR represents the Rupees and PP represents Paise) and converts the value to a string of words. e.g. if we input 125.75, the output should be "RUPEES ONE HUNDRED TWENTY FIVE AND PAISE SEVENTY-FIVE".	
26.	Write a program using pointers to read an array of integers and print its elements in reverse order.	
27.	Write a function (using pointer parameters) that compares two integer arrays to see whether they are identical. The function returns 1 if they are identical else 0	
28.	Write a program that takes as input an integer and prints if the number is Prime or Fibonacci or both. Use Functions write the program	

29.	Write a function substring that, given two strings s1 and s2, returns the starting position of the first occurrence of s1 in s2. If s1 is not in s2, return -1. For example, substring ("mom", "thermometer") returns 4 but substring ("dad" ,"thermometer") returns -1.	
30.	Write a program in C using pointers to implement insertion and deletion in a queue. A queue is a data structure that follows a first in first out i.e. the element to go in first is the one to come out first	
31.	Define a structure data type called time_struct containing three members hour, minute and second. Develop a program that will input values from the user and assign values to the individual members and display the time in the following format 16:40:40.	
32.	A start-up owner is interested to maintain the dataset of the newly recruited employees. She is interested in storing the Emp_Name (Str), Emp_Mobile(int), Emp_Age (int), Emp_Degree (Str), Emp_Exp (Float), Emp_add (Structure). Emp_add need one user defined data to store street no, city, district and state for the employee address. You have to design a database where we can store all the information for at least 20employees. The program should be interactive program to input the employee details and also the program should be able to retrieve the data of an employee based on the mobile number.	
33.	Implement the problem 22 using Files. Read the text from a file and store the count of letters and digits to another file.	
34.	Write a function using pointers to add two matrices and to return the resultant matrix to the calling function.	
35.	The prime numbers from 1 to 2500 can be obtained as follows. From a list of the numbers 1 to 2500, cross out all multiples of 2 (but not 2 itself). then, find the next number (n, say) that is not crossed out and cross out all multiples of n (but not n). repeat this last step provided that n has not exceeded 50 (the square root of 2500). The numbers remaining in the list (except 1) are prime. Write a program that uses this method to print all primes from 1to2500. storey our output in a file called primes .out. This method is called the sieve of eratosthenes.	

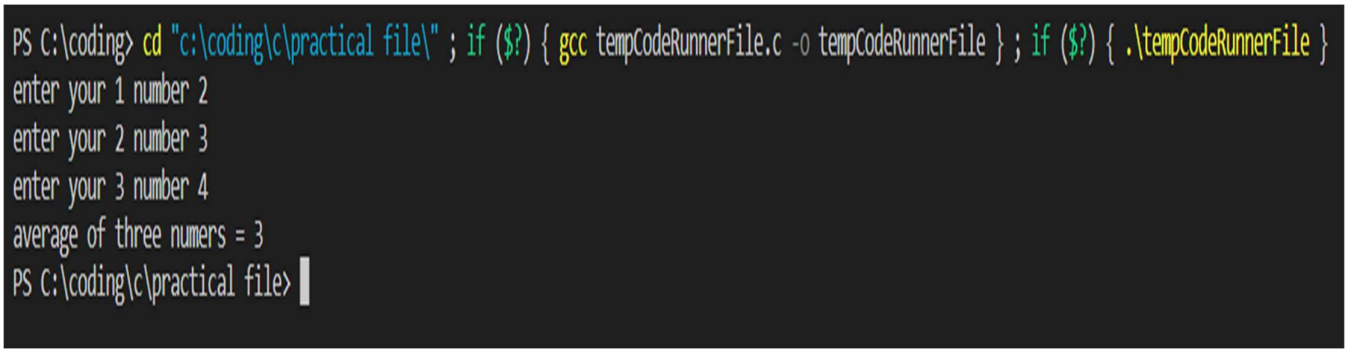
PROGRAM 1

Write a C program to input 3 numbers and print their average.

CODE:

```
#include<stdio.h>
int main(){
   int x,sum=0,avg;
   for(int i=1; i<4; i++){
      printf("enter your %d number ",i);
      scanf("%d",&x);
      sum=sum+x;
   }
   avg=sum/3;
   printf("average of three numers = %d",avg);
   return 0;
}
```

OUTPUT:



PROGRAM 2

Write a C program to enter the radius of circle/sphere and compute its (i) Perimeter (ii) Area and (iii) Volume

CODE:

```
#include<stdio.h>
int main(){
   float r,area,prmt,vol;
   printf("enter the value of radius=");
   scanf("%f",&r);
   prmt=2*3.14*r;
   area=3.14*r*r;
   vol=(4/3)*3.14*r*r*r;
   printf("perimeter =%f \n area=%f \n volume=%f ",prmt,area,vol);
   return 0;
}
```

OUTPUT:

```
                    > cd "c:\coding\c\practical file\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeR
unnerFile }
enter the value of radius=2.15
perimeter =13.502001
 area=14.514651
 volume=31.206501
```

PROGRAM 3

Write a program in C to show that Right shift effectively divides a number by 2 and a left shift effectively multiplies a number by 2.

CODE:

```
#include<stdio.h>
int main()
{
   unsigned char a ;
   printf("enter the number = ");
   scanf("%d",&a);
   printf("%d<<1 = %d\n",a,a<<1);
   printf("%dx2 = %d\n",a,a*2);
   printf("%d>>1 = %d\n",a,a>>1);
   printf("%d/2= %d\n",a,a/2);
   return 0;
}
```

OUTPUT:

```
> cd "c:\coding\c\practical file\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeR
unnerFile }
enter the number = 23
23<<1 = 46
23x2 = 46
23>>1 = 11
23/2= 11
```

PROGRAM 4

Write a C program to find the roots of an quadratic equation.

CODE:

```
#include<stdio.h>
#include<math.h>
int main(){
   int a,b,c;
   float d,r1,r2;
   printf("enter the quadratic equation : ax2+bx+c \n ");
   scanf("%d %d %d",&a,&b,&c);
   d=pow(b,2)-4*a*c;
   if(d<0){
      printf("roots are not real");
   }
   else if(d==0){
      r1=r2=(-b)/2*a;
      printf("roots are equal");
      printf("roots of %dx2+%dx+%d=0 \n %f \n %f",a,b,c,r1,r2);
   }
   else{
      r1=(-b+sqrt(d))/2*a;
      r2=(-b-sqrt(d))/2*a;
      printf("roots of %dx2+%dx+%d=0 \n %f \n %f",a,b,c,r1,r2);
   }
   return 0;
}
```

OUTPUT:

PROGRAM 5

Write down a function in C to implement bitwise AND, OR, XOR and NOT operations

CODE:

```
#include<stdio.h>
void main(){
   int n1, n2;
   printf("Enter number 1 and 2: \n");
   scanf("%d%d", &n1, &n2);
   printf("Bitwise AND of %d and %d = %d\n", n1, n2, n1 & n2);
   printf("Bitwise OR of %d and %d = %d\n", n1, n2, n1 | n2);
   printf("Bitwise NOT of %d = %d and %d = %d\n", n1, ~n1, n2, ~n2);
   printf("Bitwise XOR of %d and %d = %d\n", n1, n2, n1 ^ n2);
}
```

OUTPUT:

```
> cd "c:\coding\c\practical file\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeR
unnerFile }
Enter number 1 and 2:
2
5
Bitwise AND of 2 and 5 = 0
Bitwise OR of 2 and 5 = 7
Bitwise NOT of 2 = -3 and 5 = -6
Bitwise XOR of 2 and 5 = 7
```

PROGRAM 6

Given a n integer number write a program that displays the number as follows
First line: All digits
Second Line : All except first digit
Third line: All except first two digits
Last line : The last digit

CODE:

```
#include<stdio.h>

int main(){

   int n,rem,sum=0;

   printf("enter the number: ");

   scanf("%d",&n);

   int arr[n],i=0,j,k;

   while(n>0){

      rem=n%10;

      arr[i]=rem;

      n=n/10;

      i=i+1;

   }

   for (k = i; k >= 1; --k) {

      for (j = 1; j <= k; ++j){

         printf("%d ", arr[k-j]);

      }

      printf("\n");

   }

   return 0;

}
```

OUTPUT:

```
                    > cd "c:\coding\c\practical file\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeR
unnerFile }
enter the number: 1234
1 2 3 4
2 3 4
3 4
4
```

PROGRAM 7

Write a program to enter an integer and print the sum of the digits in the integer.

CODE:

```
#include<stdio.h>
int main(){
   int n,rem,sum=0;
   printf("enter the number: ");
   scanf("%d",&n);
   while(n>0){
      rem=n%10;
      sum=sum+rem;
      n=n/10;
   }
   printf("sum of the digits of numbers=%d",sum);
   return 0;
}
```

OUTPUT:

PROGRAM 8

Write a C program to input an investment amount and compute its fixed deposit cumulative return after 10 years at the rate of interest of 7.75%.

CODE:

```
#include<stdio.h>
#include<math.h>
int main(){
   float amount,principle;
   printf("enter the amount you want to invest= ");
   scanf("%f",&principle);
   amount=principle*pow((1+ 7.75/100),10);
   printf("your cummalative return after 10 years is %f",amount);
   return 0;
}
```

OUTPUT:

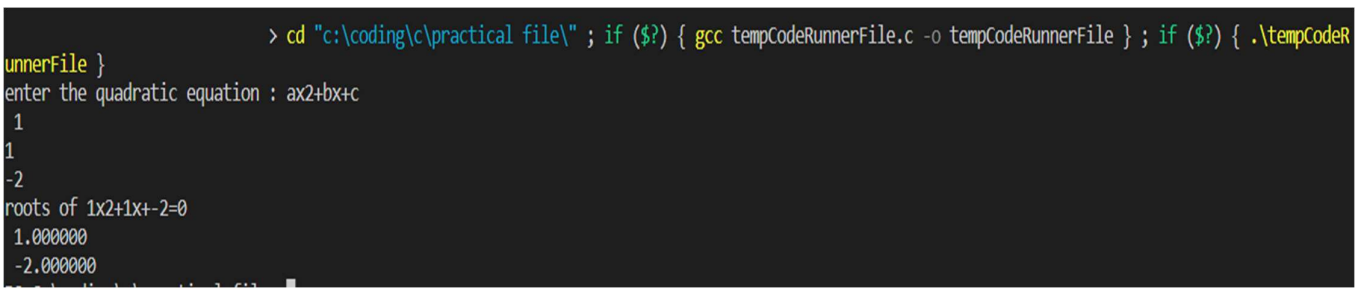
PROGRAM 9

Write A C program to compute the roots of a quadratic equation.

CODE:

```
#include<stdio.h>
#include<math.h>
int main(){
   int a,b,c;
   float d,r1,r2;
   printf("enter the quadratic equation : ax2+bx+c \n ");
   scanf("%d %d %d",&a,&b,&c);
   d=pow(b,2)-4*a*c;
   if(d<0){
      printf("roots are not real");
   }
   else if(d==0){
      r1=r2=(-b)/2*a;
      printf("roots are equal");
      printf("roots of %dx2+%dx+%d=0 \n %f \n %f",a,b,c,r1,r2);
   }
   else{
      r1=(-b+sqrt(d))/2*a;
      r2=(-b-sqrt(d))/2*a;
      printf("roots of %dx2+%dx+%d=0 \n %f \n %f",a,b,c,r1,r2);
   }
   return 0;
}
```

OUTPUT:



PROGRAM 10

A company has categorized its employees at 4 different levels (from 1 to 4). For different employees at different levels the perks are as follows

Level	TA	entertainment Allowance
1	7000	3000
2	6000	2000
3	5000	1500
4.	5000	1500

For Level 1 Basic salary is between Rs 40000 to 60000 and Tax rate is 10%

For level 2 Basic Salary is between Rs 30000 to 40000 and Tax rate is 8%

For level 3 Basic salary is between Rs 20000 to 30000 and Tax rate is 5%

For Level 4 Basic Salary is between Rs 15000 to 20000 and tax rate is 0.

Gross Salary is sum of Basic salary, Perks and HRA which is 25% of Basic Salary Tax is computed on Gross Salary. Net Salary is Gross salary- Income tax

Write a Program that will read Employees name, Level and Basic pay and will print Gross salary, Tax and Net Salary. Use Switch-case statement and if statements

CODE:

```
#include<stdio.h>
int main(){
   float ta,ea,bs,hra,gross,tax,net;
   int lvl;
   char s[100];
   printf("enter employee name ");
   scanf("%s",&s[100]);
   printf("enter the level of employee ");
   scanf("%d",&lvl);
   switch(lvl){
      case 1:
         printf("enter the basic pay ");
         scanf("%f",&bs);
         if(bs>=40000 && bs<=60000){
```

```
      gross=bs+0.4*bs+7000+3000;
      tax=0.1*bs;
      net=gross-tax;
      printf("  \n your gross salary=%f \n tax deductions=%f \n net salary=%f",gross,tax,net);
   }
   else{
      printf("enter valid basic pay");
   }
   break;
case 2:
   printf("enter the basic pay ");
   scanf("%f",&bs);
   if(bs>=30000 && bs<=40000){
      gross=bs+0.4*bs+6000+2000;
      tax=0.08*bs;
      net=gross-tax;
      printf("  \n your gross salary=%f \n tax deductions=%f \n net salary=%f",gross,tax,net);
   }
   else{
      printf("enter valid basic pay");
   }
   break;
case 3:
   printf("enter the basic pay ");
   scanf("%f",&bs);
   if(bs>=20000 && bs<=30000){
      gross=bs+0.4*bs+5000+1500;
      tax=0.05*bs;
      net=gross-tax;
      printf(" \n your gross salary=%f \n tax deductions=%f \n net salary=%f",gross,tax,net);
   }
   else{
```

```
            printf("enter valid basic pay");
        }
        break;
    case 4:
        printf("enter the basic pay ");
        scanf("%f",&bs);
        if(bs>=15000 && bs<=20000){
            gross=bs+0.4*bs+5000+1500;
            tax=0;
            net=gross-tax;
            printf("  \n your gross salary=%f \n tax deductions=%f \n net salary=%f",gross,tax,net);
        }
        else{
            printf("enter valid basic pay");
        }
        break;
    default:
        printf("enter valid level !");
  }
  return 0;
}
```

OUTPUT:

```
                        > cd "c:\coding\c\practical file\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeR
unnerFile }
enter employee name SUMIT
enter the level of employee 1
enter the basic pay 45000

 your gross salary=73000.000000
 tax deductions=4500.000000
 net salary=68500.000000
```

PROGRAM 11

Given a number, write a program using while loop to reverse the digits of the number. For example number 12345 should be written as 54321.

CODE:

```
#include<stdio.h>
int main(){
   int n,rem;
   printf("enter the number: ");
   scanf("%d",&n);
   while(n>0){
      rem=n%10;
      printf("%d ",rem);
      n=n/10;
   }
   return 0;
}
```

OUTPUT:

```
                    > cd "c:\coding\c\practical file\" ; if ($?) { gcc 11.c -o 11 } ; if ($?) { .\11 }
enter the number: 4321
1 2 3 4
```

PROGRAM 12

Write a program to find the prime numbers between a range of numbers entered by the user.

CODE:

```
#include<stdio.h>
void main(){
   int num1,num2,f;
   printf("enter the range of number between which u want the prime numbers \n");
   scanf("%d %d",&num1,&num2);
   printf("prime numbers between %d and %d \n",num1,num2);
   for(int i=num1+1;i<num2;++i ){
      f=0;
      for(int j=2; j<=i/2; ++j){
         if(i%j==0){
            f=1;
            break;
         }
      }
      if(f==0){
         printf("%d\n",i);
      }
   }
}
```

OUTPUT:

```
> cd "c:\coding\c\practical file\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeR
unnerFile }
enter the range of number between which u want the prime numbers
10
20
prime numbers between 10 and 20
11
13
17
19
```

PROGRAM 13

Write a program to find the HCF of two integers entered by the user.

CODE:

```
#include<stdio.h>
void main(){
   int num1,num2,f;
   printf("enter the range of number between which u want the prime numbers \n");
   scanf("%d %d",&num1,&num2);
   printf("prime numbers between %d and %d \n",num1,num2);
   for(int i=num1+1;i<num2;++i ){
      f=0;
      for(int j=2; j<=i/2; ++j){
         if(i%j==0){
            f=1;
            break;
         }
      }
      if(f==0){
         printf("%d\n",i);
      }
   }
}
```

OUTPUT:

```
> cd "c:\coding\c\practical file\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeR
unnerFile }
enter the first number:7
enter the second number:21
hcf of 7 and 21 = 7
```

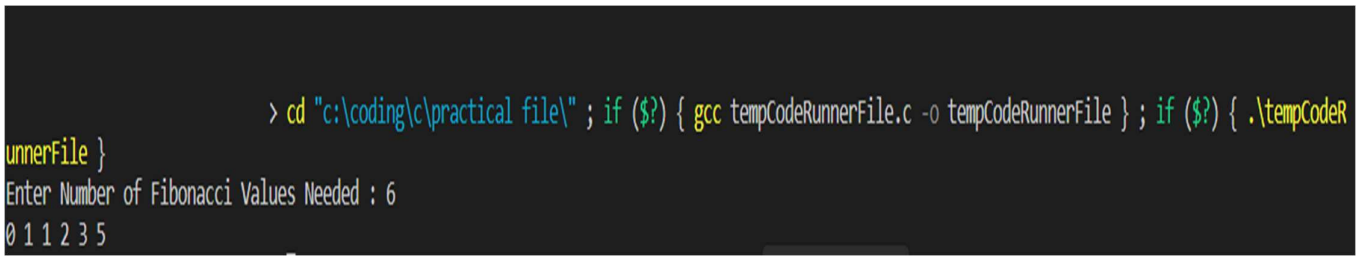
PROGRAM 14

The numbers in the sequence 1 1 2 3 5 8 13 21..... are called Fibbonacci numbers. Write a program using do...while loop to calculate and print the first m Fibonacci numbers

CODE:

```
#include<stdio.h>
void main()
{
   int i=1,n,f,f1,f2;
   printf("Enter Number of Fibonacci Values Needed : ");
   scanf("%d",&n);
   f=0;
   f1=1;
   f2=1;
   do
   {
      i++;
      printf("%d ",f);
      f1=f2;
      f2=f;
      f=f1+f2;
   }
   while(i<=n);
}
```

OUTPUT:



PROGRAM 15

Write a program to evaluate the following functions to 0.0001% accuracy

Sinx = $x-x^3/3! + x5/5! - x7/7! + \ldots\ldots$

CODE:

```
#include<stdio.h>
#include<math.h>
void cal_sin(float n)
{
   float accuracy = 0.0001, denominator, sinx, sinval;
   n = n * (3.142 / 180.0);
   float x1 = n;
   sinx = n;
   sinval = sin(n);
   int i = 1;
   do{
      denominator = 2 * i * (2 * i + 1);
      x1 = -x1 * n * n / denominator;
      sinx = sinx + x1;
      i = i + 1;
   } while (accuracy <= fabs(sinval - sinx));
   printf("VALUE OF CALCULATED sin(%f) = %f \n",n,sinx);
   printf("actual value of sin(%f)= %f",n,sinval);
}
void main(){
   float n;
   printf("enter the value of x in degrees ");
   scanf("%f",&n);
   cal_sin(n);
}
```

OUTPUT:

```
> cd "c:\coding\c\practical file\" ; if ($?) { gcc 15.c -o 15 } ; if ($?) { .\15 }
enter the value of x in degrees 45
VALUE OF CALCULATED sin(0.785500) = 0.707215
actual value of sin(0.785500)= 0.707179
```

PROGRAM 16

Write a C program to display following Pattern

```
   1
  121
12321
  121
   1
```

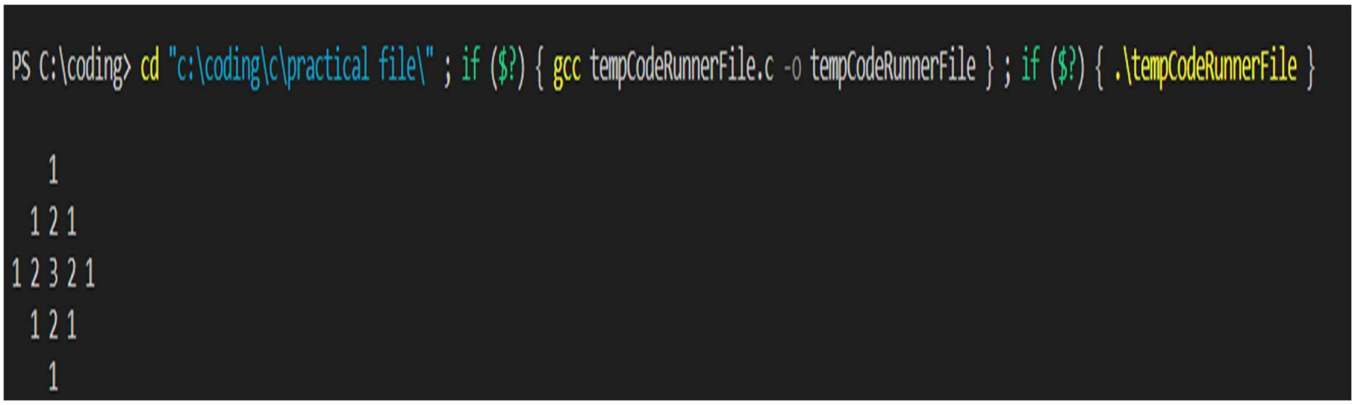
CODE:

```
#include<stdio.h>
void main(){
   int i, k, j, n=3;
   for (i = 0; i <= n; i++)
   {
      for (k = 0; k < n - i; k++)
      {
         printf("  ");
      }
      for (j = 1; j <= i; j++)
      {
         printf("%d ", j);
      }
      for (k = i - 1; k > 0; k--)
      {
         printf("%d ", k);
      }
      printf("\n");
   }

   for (i = n - 1; i > 0; i--)
```

```
    {
        for (k = 0; k < n - i; k++)
        {
            printf("  ");
        }
        for (j = 1; j <= i; j++)
        {
            printf("%d ", j);
        }
        for (k = i - 1; k > 0; k--)
        {
            printf("%d ", k);
        }
        printf("\n");
    }
}
```

OUTPUT:



PROGRAM 17

Given the two one dimensional arrays A and B of size 10 which are sorted in ascending order. Write a C program to merge them into single sorted array C that contains every item from arrays A and B in ascending order.

CODE:

```
#include<stdio.h>
void main(){
   int a[10];
   int b[10];
   int c[20];
   printf("enter 10 elements in array1 in ascencending order \n");
   int i, j = 0, m = 0, n = 0, size = 10;
   for(i=0; i<10; ++i){
      scanf("%d",&a[i]);
   }
   printf("enter 10 elements in array2 in ascencending order \n");
   for(i=0; i<10; ++i){
      scanf("%d",&b[i]);
   }

   for (i = 0; i < 2 * size; i++)
   {
      if (a[m] > b[n] && n != size)
      {
         c[j] = b[n];
         j++;
         n++;
      }
      else if (a[m] < b[n] && m != size)
```

```
    {
        c[j] = a[m];
        j++;
        m++;
    }
    else if (a[m] < b[n] && m == size)
    {
        c[j] = b[n];
        j++;
        n++;
    }
    else if (a[m] > b[n] && n == size)
    {
        c[j] = a[m];
        j++;
        n++;
    }
    else if (a[m] == b[n])
    {
        c[j] = b[n];
        n++;
        j++;
        c[j] = a[m];
        m++;
        j++;
        i++;
    }
}
for (i = 0; i < 2 * size; i++)
{
```

```
        printf("%d ", c[i]);
    }
}
```

OUTPUT:

```
enter 10 elements in array1 in ascencending order
1
2
3
4
5
6
7
8
9
10
enter 10 elements in array2 in ascencending order
2
3
45
46
47
48
49
50
51
52
1 2 2 3 3 4 5 6 7 8 9 10 45 46 47 48 49 50 51 52
```

PROGRAM 18

Write a program that will count the number of occurrences of a specified character in a given line of Text.

CODE:

```
#include<stdio.h>
#include<string.h>
void main(){
   char str[100],ch;
   int count=0,l,i,j;
   printf("enter the line: ");
   gets(str);
   printf("enter the character ");
   scanf("%c",&ch);
   l=strlen(str);
   for(i=0;i<=l;++i){
      if(str[i]==ch){
         count+=1;
      }
   }
   if(count==0){
      printf("character is not present in line");
   }
   else{
      printf("count of character %d",count);
   }
}
```

OUTPUT:

```
                          > cd "c:\coding\c\practical file\" ;
unnerFile }
enter the line: AMITISABADBOY
enter the character D
count of character 1
```

PROGRAM 19

Write a program to enter two 3 x 3 matrices and find their

a. sum

b. Multiplication

c. Transpose

CODE:

```
#include<stdio.h>
int main(){
    int mat1[3][3], mat2[3][3], mat3[3][3], i, j, x, k, sum;
    printf("Enter matrix 1 elements:\n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            scanf("%d", &mat1[i][j]);
        }
        // printf("\n");
    }
    printf("Enter matrix 2 elements:\n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            scanf("%d", &mat2[i][j]);
        }
        // printf("\n");
    }
    printf("Enter the operation: \n1.Addition\n2.Subtraction\n3.Transpose\n");
    scanf("%d", &x);
    switch (x)
    {
    case 1:
        for (i = 0; i < 3; i++)
```

```
    {
        for (j = 0; j < 3; j++)
        {
            mat3[i][j] = mat1[i][j] + mat2[i][j];
        }
    }
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            printf("%d ", mat3[i][j]);
        }
        printf("\n");
    }
    break;
case 2:
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            mat3[i][j] = mat1[i][j] - mat2[i][j];
        }
    }
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            printf("%d ", mat3[i][j]);
        }
        printf("\n");
    }
    break;
case 3:
    printf("Transpose of matrix 1:\n");
```

```
        for (i = 0; i < 3; i++)
        {
            for (j = 0; j < 3; j++)
            {
                printf("%d ", mat1[j][i]);
            }
            printf("\n");
        }
        printf("Transpose of matrix 2:\n");
        for (i = 0; i < 3; i++)
        {
            for (j = 0; j < 3; j++)
            {
                printf("%d ", mat2[j][i]);
            }
            printf("\n");
        }
        break;
    default:
        printf("Enter valid value!");
    }
    return 0;
}
```

OUTPUT:

```
PS C:\coding> cd "c:\coding\c\practical file\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter matrix 1 elements:
2
3
4
5
6
7
8
9
12
```

```
Enter matrix 2 elements:
1
2
3
-
4
5
6
7
8
```

```
Enter the operation:
1.Addition
2.Subtraction
3.Transpose
3
Transpose of matrix 1:
2 5 8
3 6 9
4 7 12
Transpose of matrix 2:
1 0 6
2 4 7
```

PROGRAM 20

Write a program that counts the number of vowels, consonants and digits in a given line of string

CODE:

```
#include<stdio.h>
#include<string.h>
int main()
{
   int i, vCount = 0, cCount = 0,dcount=0;
   char str[1000] ;
   printf("enter the line :");
   gets(str);
   for(i = 0; i < strlen(str); i++){
      str[i] = tolower(str[i]);
      if(str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u') {
         vCount++;
      }
      else if(str[i] >= 'a' && str[i] <= 'z'){
            cCount++;
      }
      else if(str[i]>='0' && str[i]<='9' ){
         dcount++;
      }
   }
   printf("Number of vowels : %d\n", vCount);
   printf("Number of consonant : %d\n", cCount);
   printf("number of digits %d", dcount);
   return 0;
}
```

OUTPUT:

```
PS C:\coding> cd "c:\coding\c\practical file\" ; if ($?) { gcc 20.c -o 20 } ; if ($?) { .\20 }
enter the line :amitisabadboy
Number of vowels : 6
Number of consonant : 7
number of digits 0
```

PROGRAM 21

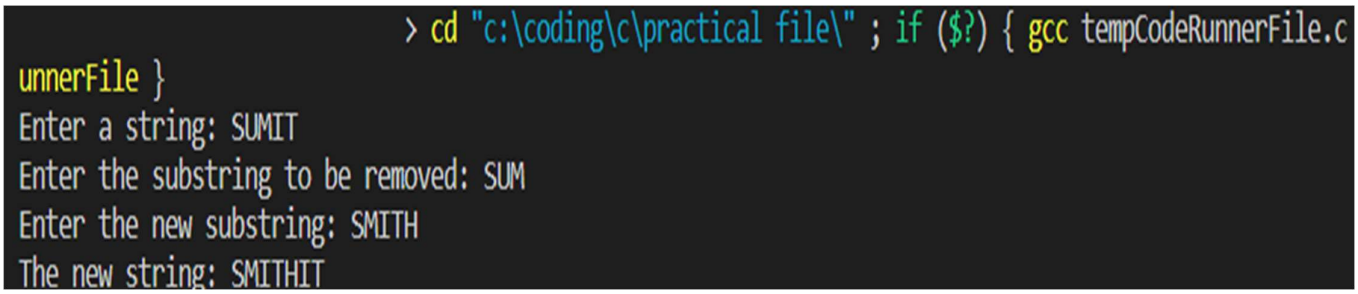
WAP that replaces a substring with another string in a given line of text.

CODE:

```
#include <stdio.h>
#include <string.h>
void replace(char str[],char sub[],char nstr[])
{
   int strLen,subLen,nstrLen;
   int i=0,j,k;
   int flag=0,start,end;
   strLen=strlen(str);
   subLen=strlen(sub);
   nstrLen=strlen(nstr);
   for(i=0;i<strLen;i++)
   {
      flag=0;
      start=i;
      for(j=0;str[i]==sub[j];j++,i++){
         if(j==subLen-1){
            flag=1;
         }
      }
      end=i;
      if(flag==0){
         i-=j;
      }
      else
      {
         for(j=start;j<end;j++)
         {
            for(k=start;k<strLen;k++){
               str[k]=str[k+1];
            }
            strLen--;
            i--;
```

```
            }
            for(j=start;j<start+nstrLen;j++)
            {
                for(k=strLen;k>=j;k--){
                    str[k+1]=str[k];
                }
                str[j]=nstr[j-start];
                strLen++;
                i++;
            }
        }
    }
}
int main()
{
    char str[20],sub[20],nstr[50];
    printf("Enter a string: ");
    scanf("%s",str);
    printf("Enter the substring to be removed: ");
    scanf("%s",sub);
    printf("Enter the new substring: ");
    scanf("%s",nstr);
    replace(str,sub,nstr);
    printf("The new string: %s",str);
    return 0;
}
```

OUTPUT:



PROGRAM 22

Write a program that takes as input a line of text and counts the frequency of each digit and letter in the input. The program will treat an uppercase letter and its lowercase equivalent as the same letter; For example, E and e increment the same counter.

CODE:

```
#include <stdio.h>
int main() {
    char line[100];
    int i=0, digit[10] = {0}, alphabet[26] = {0};
    printf("Enter a line of text: ");
    fgets(line,sizeof(line),stdin);
    while(line[i]!='\0'){
        if (line[i] >= '0' && line[i] <= '9') {
            digit[line[i] - '0']++;
        } else if (line[i] >= 'A' && line[i] <= 'Z') {
            alphabet[line[i] - 'A']++;
        } else if (line[i] >= 'a' && line[i] <= 'z') {
            alphabet[line[i] - 'a']++;
        }
        i++;
    }
    printf("Digit frequency: ");
    for (i = 0; i < 10; i++) {
        if (digit[i]>0) {
            printf("%d: %d, ", i, digit[i]);
        }
    }
    printf("\nAlphabet frequency: ");
    for (i = 0; i < 26; i++) {
        if (alphabet[i]>0) {
            printf("%c: %d, ", 'a' + i, alphabet[i]);
        }
    }
    return 0;
}
```

OUTPUT:

```
> cd "c:\coding\c\practical file\"
Enter a line of text: sumit12345
Digit frequency: 1: 1, 2: 1, 3: 1, 4: 1, 5: 1,
Alphabet frequency: i: 1, m: 1, s: 1, t: 1, u: 1,
```

PROGRAM 23

Write a program that takes as input maximum 100 numbers from user (+ve integers) and calculates (i)sum (ii) mean (iii)standard deviation and (iv) variance .

CODE:

```
#include <math.h>
#include <stdio.h>
void calculateSD(float data[],int n);
int main() {
    int i,n;
    float data[100];
    printf(" number of  elements: ");
    scanf("%d",&n);
    printf("enter the elements \n");
    for (i = 0; i < n; ++i){
        scanf("%f", &data[i]);
    }
    calculateSD(data,n);
    return 0;
}
void calculateSD(float data[],int n) {
    float sum = 0.0, mean, SD = 0.0;
    int i;
    for (i = 0; i < n; ++i) {
        sum += data[i];
    }
    mean = sum / 10;
    printf("mean of given numbers %f \n",mean);
    printf("sum of given numbers %f \n",mean);
    for (i = 0; i < n; ++i) {
        SD += pow(data[i] - mean, 2);
    }
    printf("variance of given numbers %f \n",SD/n);
    printf("standard deviation of given numbers %f \n" ,sqrt(SD/n));
}
```

OUTPUT:

```
 number of  elements: 10
enter the elements
23
23
24
34
54
65
76
89
12
23
mean of given numbers 42.299999
sum of given numbers 42.299999
variance of given numbers 640.809961
standard deviation of given numbers 25.314224
```

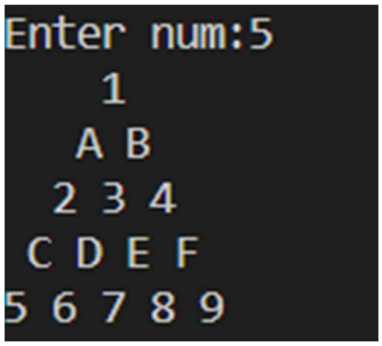
PROGRAM 24

Write a C program to display following Pattern:

CODE:

```
#include<stdio.h>
int main(){
    int i, j, alp = 65, num = 1, k, n;
    printf("Enter num:");
    scanf("%d", &n);
    for (i = 1; i <= n; i++)
    {
        for (k = 0; k < n - i; k++)
            printf(" ");
        if (i % 2 == 0)
        {
            for (j = 1; j <= i; j++)
            {
                printf("%c ", alp);
                alp++;
            }
        }
        else
        {
            for (j = 1; j <= i; j++)
            {
                printf("%d ", num);
                num++;
            }
        }
        printf("\n");
    }
}
```

OUTPUT:



PROGRAM 25

Write a program that reads the cost of an item in the form RRR.PP(Where RRRR represents the Rupees and PP represents Paise) and converts the value to a string of words. e.g. if we input 125.75, the output should be "RUPEES ONE HUNDRED TWENTY FIVE AND PAISE SEVENTY-FIVE".

CODE:

```
#include<stdio.h>

#include<string.h>

int main(){

   char *ones[] = {"", "ONE ", "TWO ", "THREE ", "FOUR ", "FIVE ", "SIX ", "SEVEN ", "EIGHT ", "NINE "};

   char *tens[] = {"", "TEN ", "TWENTY ", "THIRTY ", "FOURTY ", "FIFTY ", "SIXTY ", "SEVENTY ", "EIGHTY ", "NINETY "};

   char *elevs[] = {"", "ELEVEN ", "TWELVE ", "THIRTEEN ", "FOURTEEN ", "FIFTEEN ", "SIXTEEN ", "SEVENTEEN ", "EIGHTEEN ", "NINETEEN "};

   char *hundreds[] = {"", "ONE HUNDRED ", "TWO HUNDRED ", "THREE HUNDRED ", "FOUR HUNDRED ", "FIVE HUNDRED ", "SIX HUNDRED ", "SEVEN HUNDRED ", "EIGHT HUNDRED ", "NINE HUNDERED "};

   int n, o, t, h;

   float num;

   printf("enter the amount in RRR.PP \n");

   scanf("%f", &num);

   n = (int)num;

   o = n % 10;

   h = n / 100;

   t = n / 10 - (h * 10);

   char str[500] = "RUPEES ";

   strcat(str, hundreds[h]);

   if (n % 100 > 20)

   {
```

```
        strcat(str, tens[t]);
        strcat(str, ones[o]);
    }
    else if (n % 100 > 10)
    {
        strcat(str, elevs[n % 10]);
    }
    else
    {
        strcat(str, ones[o]);
    }
    strcat(str, "AND PAISE ");
    n = (num - n + 0.01) * 100;
    o = n % 10;
    t = n / 10;
    strcat(str, tens[t]);
    strcat(str, ones[o]);
    puts(str);
    return 0;
}
```

OUTPUT:

PROGRAM 26

Write a program using pointers to read an array of integers and print its elements in reverse order.

CODE:

```
#include<stdio.h>
int main(){
   int arr[100];
   int *p, i, n,k ;
   printf("number of elements u want to enter ");
   scanf("%d",&n);
   printf("enter the elements in array \n");
   for(i=0; i<n; i++){
      scanf("%d",&k);
      arr[i]=k;
   }
   p = &arr[n - 1];
   while (n--)
   {
      printf("%d ", *p);
      p--;
   }
}
```

OUTPUT:

```
                        > cd "c:\coding\c\practical file\" ; if ($?) { gcc 26.c -o 26 } ; if ($?) { .\26 }
number of elements u want to enter 6
enter the elements in array
1
2
3
4
5
6
6 5 4 3 2 1
```

PROGRAM 27

Write a function (using pointer parameters) that compares two integer arrays to see whether they are identical. The function returns 1 if they are identical else 0.

CODE:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>
int compare(int *a1, int *a2, int n)
{
	int i, flag = 0;
	for(i = 0; i < n; i++)
	{
		if(*a1 != *a2)
		{
			flag = 1;
			break;
		}
		a1++;
		a2++;
	}
	if(flag == 1)
		return 0;
	else
		return 1;
}

int main(int argc, char **argv)
{
	int a1[10], a2[10];
```

```
        int n, i;
        printf("Enter a number between 1 and 10: ");
        scanf("%d", &n);
        printf("Enter %d numbers for array 1: ", n);
        for(i = 0; i < n; i++)
                scanf("%d", &a1[i]);
        printf("Enter %d numbers for array 2: ", n);
        for(i = 0; i < n; i++)
                scanf("%d", &a2[i]);
        int a=compare(a1, a2, n);
   printf("%d",a);
   getch();
   return 0;
}
```

OUTPUT:

```
Enter a number between 1 and 10: 5
Enter 5 numbers for array 1: 1
3
4
5
6
Enter 5 numbers for array 2: 2
3
4
5
6
0
```

PROGRAM 28

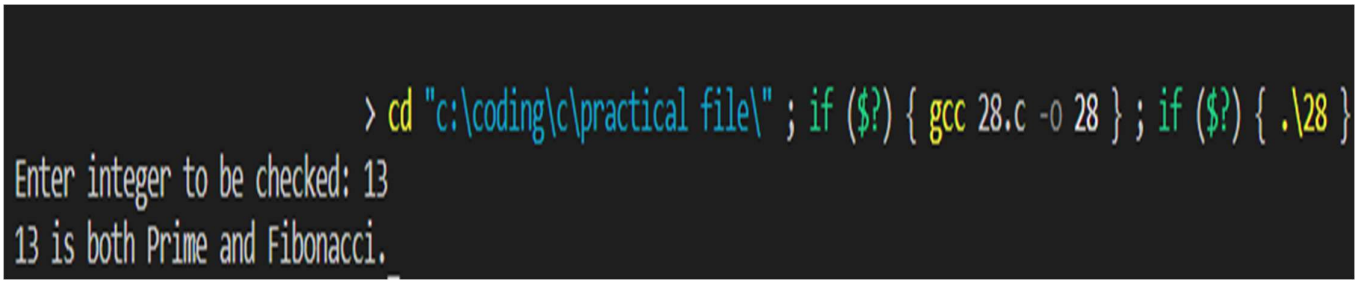
Write a program that takes as input an integer and prints if the number is Prime or Fibonacci or both. Use Functions write the program.

CODE:

```
#include <stdio.h>
#include <math.h>
int is_prime(int n)
{
   int i;
   if (n <= 1)
      return 0;
   for (i = 2; i <= sqrt(n); i++)
   {
      if (n % i == 0)
         return 0;
   }
   return 1;
}
int is_fibonacci(int n)
{
   int a = 0, b = 1, c = 0;
   while (c < n)
   {
      c = a + b;
      a = b;
      b = c;
      if (c == n)
         return 1;
   }
   return 0;
}
int main()
{
```

```
    int n;
    printf("Enter integer to be checked: ");
    scanf("%d", &n)
    if (is_prime(n) && is_fibonacci(n))
        printf("%d is both Prime and Fibonacci.\n", n);
    else if (is_prime(n))
        printf("%d is Prime not Fibonacci", n);
    else if (is_fibonacci(n))
        printf("%d is Fibonacci not Prime", n);
    else
        printf("%d is neither Prime nor Fibonacci.\n", n);
    return 0;
}
```

OUTPUT:



PROGRAM 29

Write a function substring that, given two strings s1 and s2, returns the starting position of the first occurrence of s1 in s2. If s1 is not in s2, return -1. For example, substring ("mom", "thermometer") returns 4 but substring ("dad" ,"thermometer") returns -1.

CODE:

```
#include<stdio.h>
#include<string.h>
void substring(char s[],char d[]){
   int l1=strlen(s);
   int l2=strlen(d);
   int i=0,flag,k,j,l;
   for(i=0; i<l2; i++){
      for(j=0; j<l1; j++){
         if(d[0]==s[j]){
            k=j;
            while(i+1<l2){
               l=k;
               flag=0;
               if(d[i+1]==s[j+1]){
                  flag=1;
               }
               else{
                  flag=0;
                  break;
               }
               i++;
               j++;
            }
         }
```

```
        }
    }
    if(flag==1){
        printf("subtring is found its first index value in string is= ");
        printf("%d",l);
    }
    else{
        printf("-1");
    }
}
int main(){
    char s[100],d[100];
    printf("enter the string \n");
    gets(s);
    printf("enter the substring \n");
    gets(d);
    substring(s,d);
    return 0;
}
```

OUTPUT:

```
> cd "c:\coding\c\practical file\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeR
unnerFile }
enter the string
AMITISABADBOY
enter the substring
BOY
subtring is found its first index value in string is= 10
```

PROGRAM 30

WAP in C using pointers to implement insertion and deletion in a queue. A queue is a data structure that follows a first in first out i.e. the element to go in first is the one to come out first.

CODE:

```
#include<stdio.h>
int main(){
   int i, num, x, index = 0, arr[100] = {0};
   while (1)
   {
      printf("Enter desired Operation :\n1.Insertion\n2.Deletion\n");
      scanf("%d", &x);
      if (x == -1)
      {
         break;
      }
      switch (x)
      {
      case 1:
         printf("Enter num:");
         scanf("%d", &num);
         arr[index] = num;
         index++;
         break;
      case 2:
         for (i = 0; i < index; i++)
         {
            arr[i] = arr[i + 1];
         }
         index--;
```

```
        printf("New queue is:");
        for (i = 0; i < index; i++)
        {
            printf("%d ", arr[i]);
        }
        printf("\n");
        break;
      default:
        printf("Invalid operation!\n");
        break;
      }
   }
}
```

OUTPUT:

```
Enter desired Operation :
1.Insertion
2.Deletion
1
Enter num:23
Enter desired Operation :
1.Insertion
2.Deletion
1
Enter num:24
```

```
Enter desired Operation :
1.Insertion
2.Deletion
125
Invalid operation!
Enter desired Operation :
1.Insertion
2.Deletion
2
New queue is:24
Enter desired Operation :
1.Insertion
2.Deletion
```

PROGRAM 31

Define a structure data type called time_struct containing three members hour, minute and second. Develop a program that will input values from the user and assign values to the individual members and display the time in the following format 16:40:40.

CODE:

```
#include<stdio.h>
struct time_struct
   {
      int hours, min, sec;
   };
int main(){
   struct time_struct time1;
   printf("Enter hours:");
   scanf("%d", &time1.hours);
   printf("Enter minutes:");
   scanf("%d", &time1.min);
   printf("Enter seconds:");
   scanf("%d", &time1.sec);
   printf("The time is : %d:%d:%d\n", time1.hours, time1.min, time1.sec);
}
```

OUTPUT:

```
PS C:\coding> cd "c:\coding\c\practical file\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter hours:2
Enter minutes:3
Enter seconds:34
The time is : 2:3:34
```

PROGRAM 32

A start-up owner is interested to maintain the dataset of the newly recruited employees. She is interested in storing the Emp_Name (Str), Emp_Mobile(int), Emp_Age (int), Emp_Degree (Str), Emp_Exp (Float), Emp_add (Structure). Emp_add need one user defined data to store street no, city, district and state for the employee address. You have to design a database where we can store all the information for at least 20employees. The program should be interactive program to input the employee details and also the program should be able to retrieve the data of an employee based on the mobile number.

CODE:

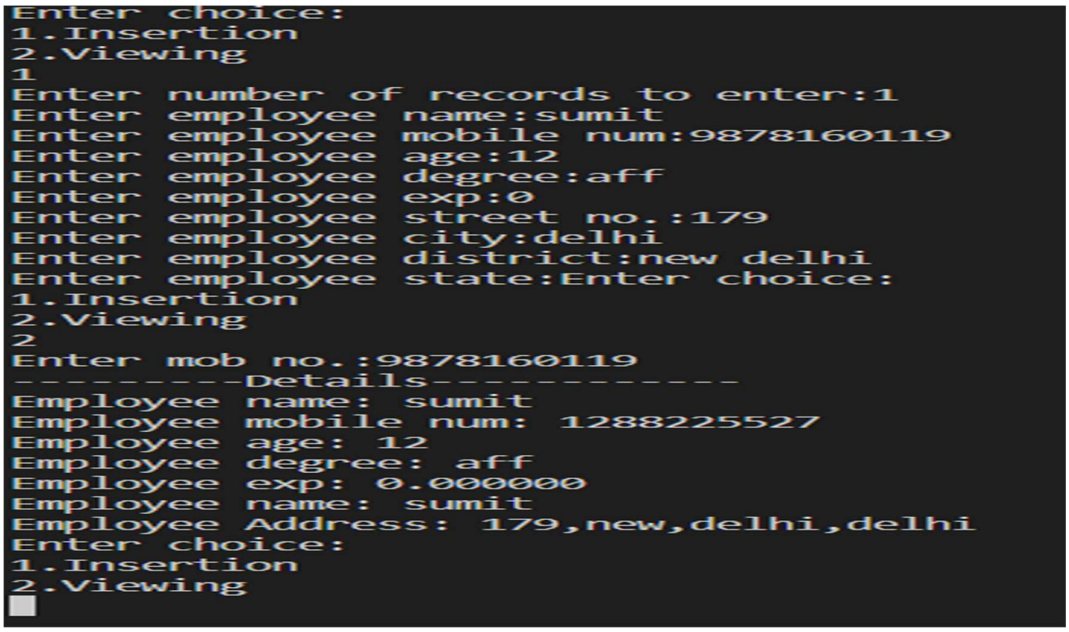
```
#include <stdio.h>
#include <string.h>
struct Emp_Add
{
    int streetNo;
    char city[100];
    char state[100];
    char district[100];
};
struct Employee
{
    char Emp_name[100];
    int Emp_mob;
    int Emp_age;
    char Emp_deg[100];
    float Emp_exp;
    struct Emp_Add emp;
};
int main()
{
    int x = 0, i, count = 0, n, num;
    struct Employee arr[30];
    for (int i = 0; i < 1; i++)
    {
        // scanf("%s", &arr[0].Emp_name);
```

```
}
while (x != -1)
{
    printf("Enter choice:\n1.Insertion\n2.Viewing\n");
    scanf("%d", &x);
    switch (x)
    {
    case 1:
        printf("Enter number of records to enter:");
        scanf("%d", &n);
        for (i = 0; i < n; i++)
        {
            printf("Enter employee name:");
            scanf("%s", &arr[i].Emp_name);
            printf("Enter employee mobile num:");
            scanf("%d", &arr[i].Emp_mob);
            printf("Enter employee age:");
            scanf("%d", &arr[i].Emp_age);
            printf("Enter employee degree:");
            scanf("%s", &arr[i].Emp_deg);
            printf("Enter employee exp:");
            scanf("%f", &arr[i].Emp_exp);
            printf("Enter employee street no.:");
            scanf("%d", &arr[i].emp.streetNo);
            printf("Enter employee city:");
            scanf("%s", &arr[i].emp.city);
            printf("Enter employee district:");
            scanf("%s", &arr[i].emp.district);
            printf("Enter employee state:");
            scanf("%s", &arr[i].emp.state);
        }
        count += n;
        break;
    case 2:
```

```
        printf("Enter mob no.:");
        scanf("%d", &num);
        for (i = 0; i < count; i++)
        {
           if (arr[i].Emp_mob == num)
           {
              printf("---------Details----------\n");
              printf("Employee name: %s\n", arr[i].Emp_name);
              printf("Employee mobile num: %d\n", arr[i].Emp_mob);
              printf("Employee age: %d\n", arr[i].Emp_age);
              printf("Employee degree: %s\n", arr[i].Emp_deg);
              printf("Employee exp: %f\n", arr[i].Emp_exp);
              printf("Employee name: %s\n", arr[i].Emp_name);
              printf("Employee Address: %d,%s,%s,%s\n", arr[i].emp.streetNo, arr[i].emp.district, arr[i].emp.city,
arr[i].emp.state);
           }
        }
     }
  }

  return 0;
}
```

OUTPUT:



PROGRAM 33

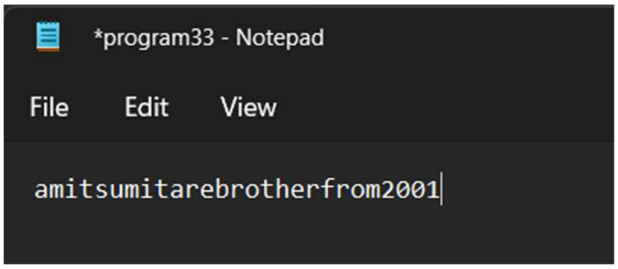
Implement the problem 22 using Files. Read the text from a file and store the count of letters and digits to another file.

CODE:

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
int main(){
   FILE *fptr;
   char line[10000];
   fptr= fopen("program33.txt","r");
   fgets(line,10000,fptr);
   int i=0, digit[10] = {0}, alphabet[26] = {0};
   while(line[i]!='\0'){
      if (line[i] >= '0' && line[i] <= '9') {
         digit[line[i] - '0']++;
      } else if (line[i] >= 'A' && line[i] <= 'Z') {
         alphabet[line[i] - 'A']++;
      } else if (line[i] >= 'a' && line[i] <= 'z') {
         alphabet[line[i] - 'a']++;
      }
      i++;
   }
   printf("Digit frequency: ");
   for (i = 0; i < 10; i++) {
      if (digit[i]>0) {
         printf("%d: %d, ", i, digit[i]);
      }
   }
   printf("\nAlphabet frequency: ");
```

```
    for (i = 0; i < 26; i++) {
        if (alphabet[i]>0) {
            printf("%c: %d, ", 'a' + i, alphabet[i]);
        }
    }
    fclose(fptr);
    return 0;
}
```

OUTPUT:




```
                        > cd "c:\coding\c\practical file\" ; if ($?) { gcc 33.c -o 33 } ; if ($?) { .\33 }
Digit frequency: 0: 2, 1: 1, 2: 1,
Alphabet frequency: a: 2, b: 1, e: 2, f: 1, h: 1, i: 2, m: 3, o: 2, r: 4, s: 1, t: 3, u: 1,
```

PROGRAM 34

Write a function using pointers to add two matrices and to return the resultant matrix to the calling function.

CODE:

```
#include<stdio.h>
int a[5][5],b[5][5],row,col;
void add(int(*)[5]);
int main(){
    int c[5][5],i,j;
    printf("Enter row : ");
    scanf("%d",&row);
    printf("Enter column : ");
    scanf("%d",&col);
    printf("Enter matrix A :\n");
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("Enter matrix B :\n");
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }
    add(c);
    printf("Addition :\n");
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
```

```
        {
            printf("%d\t",c[i][j]);
        }
        printf("\n");
    }
    return 0;
}
void add(int c[5][5]){
    int i,j;
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
        c[i][j]=a[i][j]+b[i][j];
        }
    }
}
```

OUTPUT:

```
Enter row : 3
Enter column : 3
Enter matrix A :
1
2
3
4
5
6
7
8
9
Enter matrix B :
12
13
14
15
16
17
18
19
10
Addition :
13      15      17
19      21      23
25      27      19
```

PROGRAM 35

The prime numbers from 1 to 2500 can be obtained as follows. From a list of the numbers 1 to 2500, cross out all multiples of 2 (but not 2 itself). then, find the next number (n, say) that is not crossed out and cross out all multiples of n (but not n). repeat this last step provided that n has not exceeded 50 (the square root of 2500). The numbers remaining in the list (except 1) are prime. Write a program that uses this method to print all primes from 1to2500. storey our output in a file called primes .out. This method is called the sieve of eratosthenes.

CODE:

```
#include <stdio.h>
#define N 2500
int main() {
   int is_prime[N+1];
   int i, j;
   for (i = 2; i <= N; i++) {
      is_prime[i] = 1;
   }
   for (i = 2; i <= 50; i++) {
      if (is_prime[i]) {
         for (j = i*i; j <= N; j += i) {
            is_prime[j] = 0;
         }
      }
   }
   FILE *fp = fopen("primes.out", "w");
   for (i = 2; i <= N; i++) {
      if (is_prime[i]) {
         printf("%d\n", i);
         fprintf(fp, "%d\n", i);
      }
   }
   fclose(fp);
   return 0;
}
```

OUTPUT:

```
2    3    5    7    11   13   17   19   23   29   31   37   41   43   47   53   59   61   67   71   73   7
9    83   89   97   101  103  107  109  113  127  131  137  139  149  151  157  163  167  173  179  181  1
91   193  197  199  211  223  227  229  233  239  241  251  257  263  269  271  277  281  283  293  307  3
11   313  317  331  337  347  349  353  359  367  373  379  383  389  397  401  409  419  421  431  433  4
39   443  449  457  461  463  467  479  487  491  499  503  509  521  523  541  547  557  563  569  571  5
77   587  593  599  601  607  613  617  619  631  641  643  647  653  659  661  673  677  683  691  701  7
09   719  727  733  739  743  751  757  761  769  773  787  797  809  811  821  823  827  829  839  853  8
57   859  863  877  881  883  887  907  911  919  929  937  941  947  953  967  971  977  983  991  997  1
009  1013 1019 1021 1031 1033 1039 1049 1051 1061 1063 1069 1087 1091 1093 1097 1103 1109 1117 1123 11291
151  1153 1163 1171 1181 1187 1193 1201 1213 1217 1223 1229 1231 1237 1249 1259 1277 1279 1283 1289 12911
297  1301 1303 1307 1319 1321 1327 1361 1367 1373 1381 1399 1409 1423 1427 1429 1433 1439 1447 1451 14531
459  1471 1481 1483 1487 1489 1493 1499 1511 1523 1531 1543 1549 1553 1559 1567 1571 1579 1583 1597 16011
607  1609 1613 1619 1621 1627 1637 1657 1663 1667 1669 1693 1697 1699 1709 1721 1723 1733 1741 1747 17531
759  1777 1783 1787 1789 1801 1811 1823 1831 1847 1861 1867 1871 1873 1877 1879 1889 1901 1907 1913 19311
933  1949 1951 1973 1979 1987 1993 1997 1999 2003 2011 2017 2027 2029 2039 2053 2063 2069 2081 2083 20872
089  2099 2111 2113 2129 2131 2137 2141 2143 2153 2161 2179 2203 2207 2213 2221 2237 2239 2243 2251 22672
269  2273 2281 2287 2293 2297 2309 2311 2333 2339 2341 2347 2351 2357 2371 2377 2381 2383 2389 2393 23992
411  2417 2423 2437 2441 2447 2459 2467 2473 2477
```