

LAB MANUAL

PROGRAMMING FOR PROBLEM SOLVING LABORATORY
COURSE CODE: 21ESPPL115

1. a. Find all the possible roots of Quadratic Equation.

```
#include<stdio.h>
#include<math.h>
int main () {
    float a,b,c,r1,r2,d,realPart,imgPart;
    printf ("Enter the values of a b c: ");
    scanf (" %f %f %f", &a, &b, &c);
    if(a==0)
    {
        printf("Enter a valid value for a");
    }
    else
    {
        d= (b*b) - (4*a*c);
        if (d>0) {
            r1 = (-b+sqrt (d)) / (2*a);
            r2 = (-b-sqrt (d)) / (2*a);
            printf ("The roots are real and distinct.\n");
            printf("r1= %f, r2= %f", r1, r2);

        }

        else if (d==0)
        {
            r1 = -b/(2*a);
            r2 = -b/(2*a);
            printf ("Roots are real and equal.\n");
            printf("r1= %f, r2= %f", r1, r2);

        }

        else
        {
            realPart=-b/(2*a);
            imgPart=sqrt(fabs(d))/(2*a);
            printf("Roots are real and imaginary.\n");
            printf("r1=%f+i%f\nr2=%f-i%f",realPart,imgPart,realPart,imgPart);
        }
    }
    return 0;
}
```

OUTPUT:

Case 1:

Enter the values of a b c: 1 4 4

Roots are real and equal.

r1= -2.000000, r2= -2.000000

Case 2:

Enter the values of a b c: 1 2 3

Roots are real and imaginary.

r1=-1.000000+i1.414214

r2=-1.000000-i1.414214

Case 3:

Enter the values of a b c: 1 3 2

The roots are real and distinct.

r1= -1.000000, r2= -2.000000

Case 4:

Enter the values of a b c: 0 1 1

Enter a valid value for a

Find roots of a quadratic equation, ax^2+bx+c .

There will be 2 roots for given quadratic equation.

Analysis

Input – a,b,c values

Output – r1, r2 values

$$r1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$r2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Start

Read a, b, c values

Compute $d = b^2 - 4ac$

if $d > 0$ then

$$r1 = \frac{-b + \sqrt{d}}{2a}$$

$$r2 = \frac{-b - \sqrt{d}}{2a}$$

Otherwise if $d = 0$ then

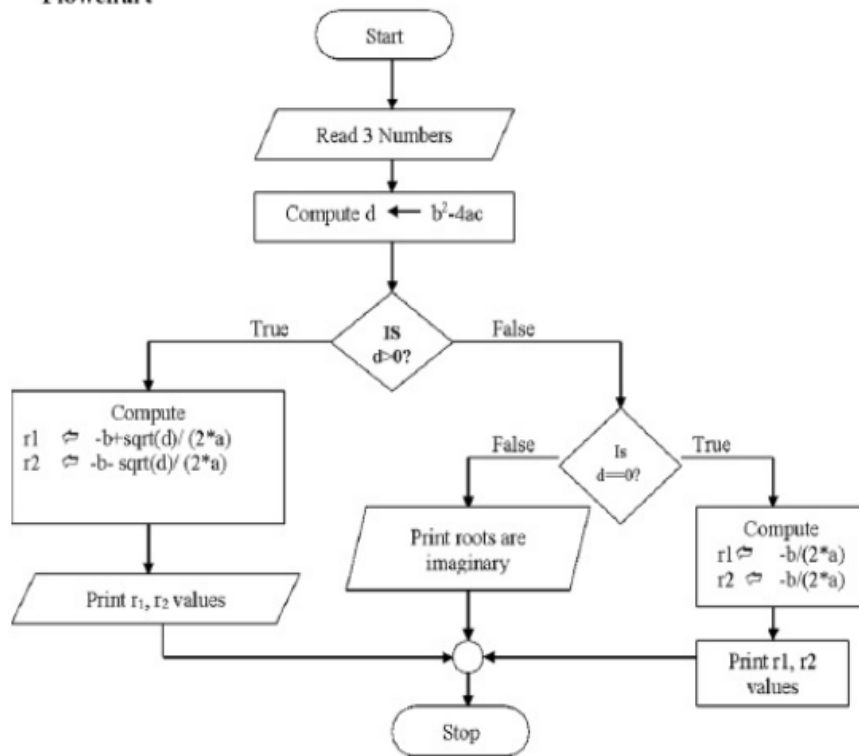
$$\text{compute } r1 = -b/2a, r2 = -b/2a$$

print r1, r2 values

Otherwise if $d < 0$ then print roots are imaginary

Stop

Flowchart



1. b. Find the reverse of an integer and check whether it is a palindrome or not.

```
#include <stdio.h>
int main()
{
    int num, temp, rem, rev = 0;
    printf("Enter a number:");
    scanf("%d", &num);
    temp = num;
    while ( temp > 0)
    {
        rem = temp % 10;
        rev = rev * 10 + rem;
        temp = temp / 10;
    }
    printf("reversed number is = %d\n", rev);
    if ( num == rev )
        printf("%d is a Palindrome.", num);
    else
        printf("%d is not a Palindrome.", num);
    return 0;
}
```

OUTPUT:

Enter a number:121
Reversed number is = 121
121 is a Palindrome.

Enter a number:12222
Reversed number is = 22221
12222 is not a Palindrome.

2. a. Find the GCD of two integers

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int m,n;
    printf("Input 2 numbers : ");
    scanf("%d%d",&m,&n);
    if(m<=0||n<=0)
        printf("Invalid Input");
    else{
        while(m!=n)
        {
            if(m>n)
                m=m-n;
            else
                n=n-m;
        }
        printf("GCD=%d\n",n);
    }
    getch();
}
```

OUTPUT:

```
Input two numbers
45
9
GCD = 9
```

```
Input two numbers
13
31
GCD = 1
```

2. b. Generate and print first N Fibonacci numbers using recursion.

```
#include<stdio.h>
#include<conio.h>
int fib (int n)
{
    if (n == 0 || n == 1)
        return n;
    else
        return (fib(n-1) + fib(n-2));
}
void main ()
{
    int n, i ;
    clrscr();
    printf("Please enter Limit for Fibonacci Series: ");
    scanf("%d", &n);
    printf("\nFibonacci series terms are:\n");
    for (i = 0; i < n; i++)
    {
        printf("%d\n", fib(i));
    }
}
```

If part of the recursive function is called as the base class and it sees that the function doesn't repeat infinity times. It also solves the problem

It is called as the general case

OUTPUT:

```
Please enter Limit for Fibonacci Series:8
Fibonacci series terms are:
0
1
1
2
3
5
8
13
```

3. a. Compute mean, variance and standard deviation of N real numbers.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define SIZE 100 //Symbolic Constant
void main()
{
    int n, i;
    float a[SIZE], sum=0, mean=0, variance=0, deviation=0;
    clrscr();
    printf("Enter the size of array ");
    scanf("%d",&n);
    printf("Enter the numbers in array \n");
    for(i=0;i<n;i++)
    {
        scanf ("%f",&a[i]);
    }
    // find mean value
    for(i=0;i<n;i++)
    {
        sum = sum+a[i];
    }
    mean = sum/n;
    printf("\nMean (Average)= %f", mean);
    // find variance value
    sum=0;
    for(i=0;i<n;i++)
    {
        sum = sum+(a[i]-mean)*(a[i]-mean);
    }
    variance = sum / n;
    printf("\nVariance = %f", variance);
    // find standard deviation
    deviation = sqrt (variance);
    printf("\nDeviation = %f \n", deviation);
}
```

Output:

```
Enter the size of array 5
Enter the numbers in array
1
5
10
15
20
Mean (Average)= 10.200000
Variance = 46.159996
Deviation = 6.794115
```


3. b. Search an element using linear search method.

```
#include<stdio.h>
#include<conio.h>
#define SIZE 100
void main()
{
    int n, key, a[SIZE], i, found =0;
    printf("Enter the size of array : ");
    scanf("%d", &n);
    printf("Enter the numbers in array \n");
    for(i=0; i < n; i++)
        scanf ("%d", &a[i]);
    printf("Enter the elements to be searched : ");
    scanf("%d", &key);
    // linear search logic
    for(i=0; i < n; i++)
    {
        if (a[i] == key)
        {
            found=1;
            break;
        }
    }
    if (found == 1)
        printf("Key Element %d found at position= %d\n", key , i+1 );
    else
        printf("Key Element NOT found\n");
    getch();
}
```

OUTPUT:

```
Enter the size of array: 6
Enter the numbers in array
4532
15
23
59894
1543
100
Enter the elements to be searched: 1543
Key Element 1543 found at position= 5
```

4. a. Interchange the largest and smallest number in the array.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[10],i,n,min_Ele,max_Ele,min_Pos,mx_Pos,temp;
    clrscr();
    printf("Enter the array size:");
    scanf("%d",&n);
    printf("Enter the elements of array: ");
    for(i=0;i<n;i++) //Inputting Array Elements
        scanf("%d",&a[i]);
    printf("The array elements are : ");
    for(i=0;i<n;i++) //Printg Array Elements
        printf("%d ",a[i]);
    min_Ele=a[0];
    max_Ele=a[0];
    for(i=0;i<n;i++)
    {
        if(a[i]<min_Ele)
        {
            min_Ele=a[i];
            min_Pos=i;
        }
        if(a[i]>max_Ele)
        {
            max_Ele=a[i];
            mx_Pos=i;
        }
    }
    printf("\n smallest number:=%d \n largest number:=%d\n",min_Ele, max_Ele);
    temp=a[min_Pos];
    a[min_Pos]=a[mx_Pos];
    a[mx_Pos]=temp;
    printf("\nAfter interchanging largest & smallest values the array:\n");
    for(i=0;i<n;i++)
        printf("%d ",a[i]);
    getch();
}
```

OUTPUT:

```
Enter the array size: 8
Enter the elements of array:
16 0 45 98 5432 698 13 9999
The array elements are: 16 0 45 98 5432 698 13 9999
smallest number: =0
largest number: =9999
After interchanging largest & smallest values the array:
16 9999 45 98 5432 698 13 0
```

4.b. Search an element using binary search method.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[10],i,n,m,c=0,low,high,mid;
    printf("Enter the size of an array: ");
    scanf("%d",&n);
    printf("Enter the elements in ascending order:\n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    printf("Enter the number to be searched: ");
    scanf("%d",&m);
    low=0, high=n-1;
    while(low<=high)
    {
        mid=(low+high)/2;
        if(m==a[mid])
        {
            c=1;
            printf("\n %d found at %d position\n",m,mid+1);
            break;
        }
        else if(m<a[mid])
        {
            high=mid-1;
        }
        else
            low=mid+1;
    }
    if(c==0)
        printf("%d is not found\n",m);
    getch();
}
```

OUTPUT:

1.

```
Enter the size of an array: 10
Enter the elements in ascending order:
1 23 45 65 100 150 900 1505 8253 11000
Enter the number to be search: 8253
8253 found at 8 position
```

2.

```
Enter the size of an array: 5
Enter the elements in ascending order:
2 4 6 8 10
Enter the number to be search: 3
3 is not found
```

5. a.To check whether a given string is palindrome or not without using library functions

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,pal_len=0,len=0;
    char pal_str[100];
    printf("Enter a string to check whether it is palindrome or not: ");
    gets(pal_str);
    for(i=0;i<100;i++)
    {
        if(pal_str[i]=='\0')
        {
            break;
        }
        len++; // Calculating length of string
    }
    for(i=0;i<len;i++)
    {
        if(pal_str[i]==pal_str[len-1-i])
        {
            pal_len++;
        }
    }
    if(len==pal_len)
    {
        printf("Entered string is palindrome");
    }
    else
    {
        printf("Entered string is not palindrome");
    }
    getch();
}
```

OUTPUT:

1.

Enter a string to check whether it is palindrome or not: GADAG

Entered string is palindrome

2.

Enter a string to check whether it is palindrome or not: computer

Entered string is not palindrome

5.b. Find the number of vowels, consonants, digits and white spaces in a string.

```
#include<stdio.h>
#include<conio.h>
void countCharType(char str[100])
{
    // Declare the variable vowels, consonant, digit, WhiteSpace
    int vowels = 0, consonant = 0, WhiteSpace = 0, SpecialChar=0,
    digit = 0,i;
    for (i = 0; str[i]!='\0'; i++)
    {
        char ch = str[i];
        if ( (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z') )
        {
            // To handle upper case letters
            ch = tolower(ch);
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
                vowels++;
            else
                consonant++;
        }
        else if (ch >= '0' && ch <= '9')
            digit++;
        else if(ch == ' ')
        {
            WhiteSpace++;
        }
        else
        {
            SpecialChar++;
        }
    }
    printf("\n Vowels:%d\n ",vowels);
    printf("Consonant:%d\n ", consonant);
    printf("Digit:%d\n ", digit);
    printf("WhiteSpace:%d\n ", WhiteSpace);
    printf("SpecialChar:%d\n ", SpecialChar);
}
int main()
{
    char str[100];
    printf("Enter input : ");
    gets(str);
    countCharType(str);// Calling another function
    return 0;
    getch();
}
```

OUTPUT:

Enter input : Programming for problem solving

Vowels:8

Consonant:20

Digit:0

WhiteSpace:3

SpecialChar:0

6. a. Delete an element from an array.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int array[100], position, i, n;
    clrscr();
    printf("Enter number of elements in array\n");
    scanf("%d", &n);
    printf("Enter %d elements\n", n);
    for (i = 0; i < n; i++)
    {
        scanf("%d", &array[i]);
    }
    printf("Enter the location where you wish to delete element\n");
    scanf("%d", &position);
    if (position >= n+1)
    {
        printf("Deletion not possible.\n");
    }
    else
    {
        for (i = position - 1; i < n - 1; i++)
            array[i] = array[i+1];
        printf("Resultant array:\n");
        for (i = 0; i < n - 1; i++)
            printf("%d\n", array[i]);
    }
    getch();
    return 0;
}
```

OUTPUT:

```
Enter number of elements in array
10
Enter 10 elements
8 16 32 64 128 256 512 1024 2048 4096
Enter the location where you wish to delete element
5
Resultant array:
8 16 32 64 256 512 1024 2048 4096
```

6.b. Sort N elements of an array in ascending order using bubble sort technique.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int array[100], num, c, i, swap;
    clrscr();
    printf("Enter number of elements\n");
    scanf("%d", &num);
    printf("Enter %d integers\n", num);
    for (c = 0; c < num; c++)
        scanf("%d", &array[c]);
    for (c = 0; c < num - 1; c++)
    {
        for (i = 0; i < num - c - 1; i++)
        {
            if (array[i] > array[i+1])
            {
                swap = array[i];
                array[i] = array[i+1];
                array[i+1] = swap;
            }
        }
    }
    printf("Sorted list in ascending order:\n");
    for (c = 0; c < num; c++)
        printf("%d\n", array[c]);
    getch();
    return 0;
}
```

OUTPUT:

```
Enter number of elements
5
Enter 5 integers
56 77 3 45 1
Sorted list in ascending order:
1
3
45
56
77
```

7. Read a matrix A of size MxN and find the following.

(i) Sum of the elements of the row.

(ii) Sum of the elements of the column.

(iii) Sum of all the elements of the matrix.

(iv) Sum of both diagonal elements of a matrix.

Output the computed results with suitable headings.

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    static int array[10][10];
    int i, j, m, n, sum = 0;
    printf(" Enter the order of the matrix: ");
    scanf("%d %d", &m, &n);
    printf(" Enter the elements of the matrix\n");
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            scanf("%d", &array[i][j]);
        }
    }
    // Performing Addition of elements in each rows.
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            sum = sum + array[i][j];
        }
        printf(" Sum of the %d row is = %d\n", i, sum);
        sum = 0;
    }
    // Performing Addition of elements in each column.
    sum = 0;
    for (j = 0; j < n; j++)
    {
        for (i = 0; i < m; i++)
        {
            sum = sum + array[i][j];
        }
        printf(" Sum of the %d column is = %d\n", j, sum);
        sum = 0;
    }
    // Performing Addition of all the elements of a matrix.
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            sum = sum + array[i][j];
        }
    }
    printf("\n Sum of All the elements of a matrix is = %d\n", sum);
    // Performing Addition of diagonal elements.
    sum=0;
    for(i = 0; i < n; i++)
    {
```



```
        sum = sum + array[i][i];
    }
    printf("\n The Sum of Diagonal Elements of a Matrix = %d", sum );
    getch();
}
```

OUTPUT:

Enter the order of the matrix: **3 3**

Enter the elements of the matrix

1 1 1

2 2 2

3 3 3

Sum of the **0 row** is = **3**

Sum of the **1 row** is = **6**

Sum of the **2 row** is = **9**

Sum of the **0 column** is = **6**

Sum of the **1 column** is = **6**

Sum of the **2 column** is = **6**

Sum of All the elements of a matrix is = 18

The Sum of Diagonal Elements of a Matrix = **6**

8. Input 2 matrices of size M x N and P x Q. Perform
a. Multiplication if they are compatible.
b. Transpose of the resultant matrix.
Print the result in matrix form with suitable headings.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int Matrix_A[10][10], Matrix_B[10][10], Matrix_Mul[10][10]={0}, Matrix_Trans[10][10]={0};
    int i,j,k,m,n,p,q;
    printf("Enter no. of rows and columns in matrix A: ");
    scanf("%d%d",&m,&n);
    printf("Enter no. of rows and columns in matrix B: ");
    scanf("%d%d",&p,&q);
    if(n!=p)
    {
        printf("Matrix Multiplication is not possible");
        return;
    }
    else
    {
        printf("Enter elements of matrix A: ");
        for(i=0;i<m;i++)
            for(j=0;j<n;j++)
                scanf("%d", &Matrix_A[i][j]);
        printf("Enter elements of matrix B: ");
        for(i=0;i<p;i++)
            for(j=0;j<q;j++)
                scanf("%d", &Matrix_B[i][j]);
        //Performing Multiplication of Matrices
        for(i=0;i<m;i++)
            for(j=0;j<q;j++)
                for(k=0;k<p;k++)
                    Matrix_Mul[i][j] += Matrix_A[i][k]*Matrix_B[k][j];
        printf("\nResult of Matirx Multiplication:\n");
        // Displaying Matrix_Mul
        for(i=0;i<m;i++)
        {
            for(j=0;j<q;j++)
                printf("%d ", Matrix_Mul[i][j]);
            printf("\n");
        }
        // Finding the transpose of Matrix_Mul
        for(i=0; i<m; ++i)
            for(j=0; j<q; ++j)
            {
                Matrix_Trans[j][i] = Matrix_Mul[i][j];
            }
        // Displaying the transpose of Matrix_Mul
        printf("\nTranspose of Matrix:\n");
        for(i=0; i<q; ++i)
        {
            for(j=0; j<m; ++j)
                printf("%d ",Matrix_Trans[i][j]);
        }
    }
}
```

```
        printf("\n\n");
    }

    } //End of if Statement
    getch();
}
```

OUTPUT:

Enter no. of rows and columns in matrix A: 3 3

Enter no. of rows and columns in matrix B: 3 2

Enter elements of matrix A:

1 1 1

2 2 2

3 3 3

Enter elements of matrix B:

1 1

1 1

1 1

Result of Matirx Multiplication:

3 3

6 6

9 9

Transpose of Matrix:

3 6 9

3 6 9

9. a. Swap the contents of two variables using pointers.

```
#include <stdio.h>
int main()
{
    int x, y, *a, *b, temp;
    printf("Enter the value of x and y\n");
    scanf("%d%d", &x, &y);
    printf("Before Swapping\nx = %d\nny = %d\n", x, y);
    a = &x;
    b = &y;
    //swapping values between a and b
    temp = *b;
    *b = *a;
    *a = temp;
    printf("After Swapping\nx = %d\nny = %d\n", x, y);
    return 0;
}
```

OUTPUT:

```
Enter the value of x and y
54
99
Before Swapping
x = 54
y = 99
After Swapping
x = 99
y = 54
```

9. b. Concatenate the contents of two files.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    // Open two files to be merged
    FILE *fp1 = fopen("file1.txt", "a");
    FILE *fp2 = fopen("file2.txt", "r");
    char c;
    if (fp1 == NULL || fp2 == NULL)
    {
        puts("Could not open files");
        exit(0);
    }
    fputc(' ', fp1);
    while ((c = fgetc(fp2)) != EOF)
        fputc(c, fp1);
    printf("File1 and File2 Merged please check file1");
    fclose(fp1);
    fclose(fp2);
    getch();
    return 0;
}
```

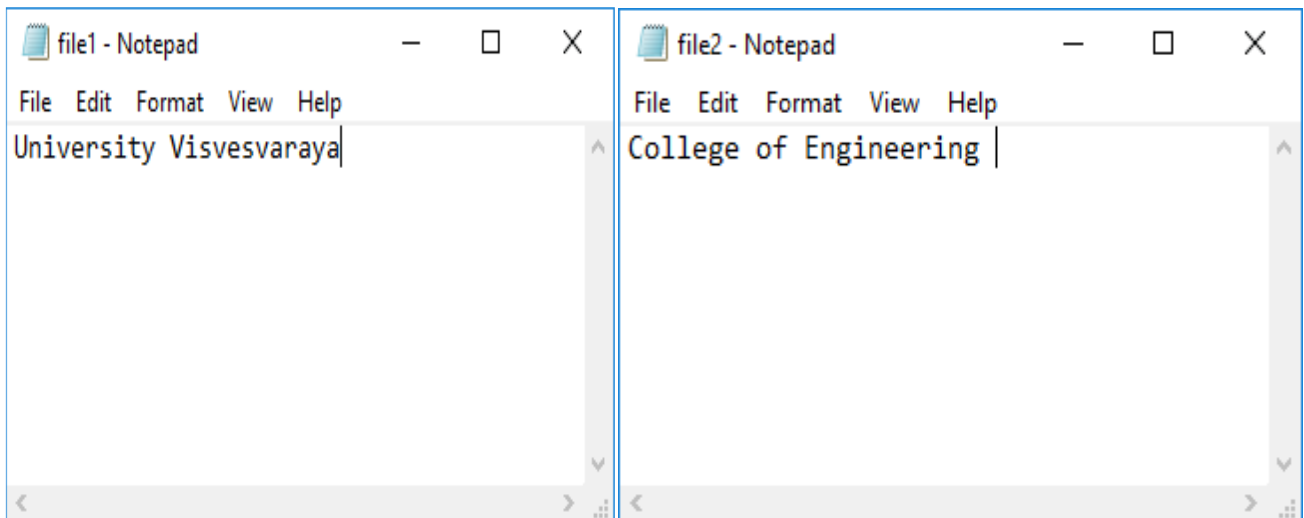
Create .txt files in TURBO C++
or
Paste path with // (2 slashes instead of one)

OUTPUT:

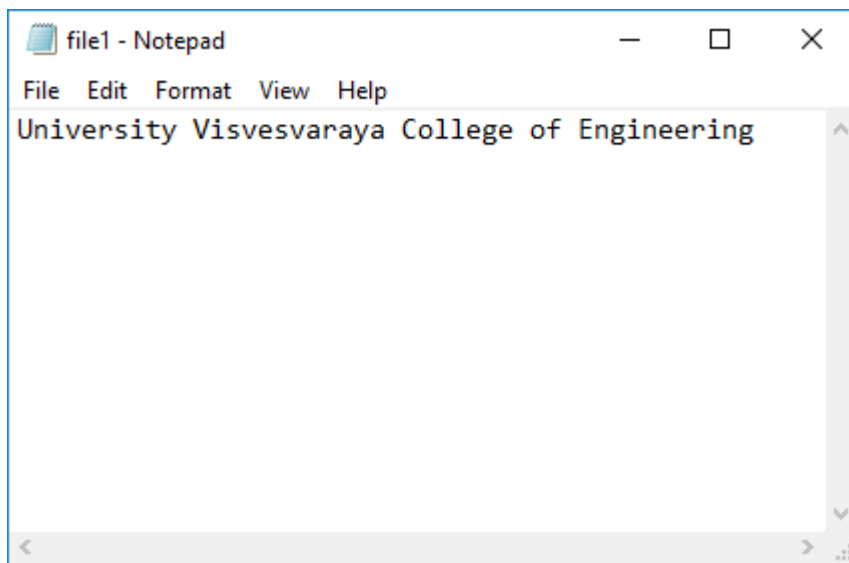
Before Running Program:

File1.txt

File2.txt



After Running Program:



```

/* 10. Define a Structure called Employee with Emp ID, Emp-name and Salary as
its data members.
Read details of N Employees and display the details of employees whose salary
is greater than ₹15000.*/
#include <stdio.h>
#include <conio.h>
struct employee
{
    char Ename[50];
    int Emp_ID;
    int Salary;
};

int display(struct employee Emp[], int n)
{
    int i;
    printf("\nThe Employee Details are as follows\n");
    for (i = 0; i < n; i++)
    {
        printf("\nEmployee %d details are:\n", i + 1);
        printf("Employee Name: %s\n", Emp[i].Ename);
        printf("Employee ID: %d\n", Emp[i].Emp_ID);
        printf("Employee Salary: %d\n", Emp[i].Salary);
    }
    return 0;
}

int main()
{
    int n, i;
    struct employee Emp1[5];
    printf("Enter the number of Employees: ");
    scanf("%d", &n);
    printf("\nEnter the details of %d Employees:", n);
    for (i = 0; i < n; i++)
    {
        printf("\nEnter Employee %d Name: ", i + 1);
        scanf("%s", Emp1[i].Ename); // Correct format specifier for string
        printf("Enter Employee %d ID: ", i + 1);
        scanf("%d", &Emp1[i].Emp_ID);
        printf("Enter Employee %d Salary: ", i + 1);
        scanf("%d", &Emp1[i].Salary);
    }

    display(Emp1, n);

    printf("\nThe Employee Details whose salary is greater than 15000\n");
    for (i = 0; i < n; i++)

```

```
{  
    if (Emp1[i].Salary > 15000)  
    {  
        printf("\n");  
        printf("Employee Name: %s\n", Emp1[i].Ename);  
        printf("Employee ID: %d\n", Emp1[i].Emp_ID);  
        printf("Employee Salary: %d\n", Emp1[i].Salary);  
    }  
}  
  
getch();  
return 0;  
}
```



```

/* 11. Create a structure called student with student name, roll-no, marks in
three tests. Write a C
program to create N records and
(i) Search on roll-no and display all the records
(ii) Average marks in each test
(iii) Highest in each test. */
#include <stdio.h>
// #include<conio.h>
struct student
{
    char name[50];
    int roll_no;
    int Test1,Test2,Test3, highest;
    float avg;
} s[50];
void display(int n)
{
    int i;
    printf("\nDisplaying Information:\n");
    printf("\nName\tRoll_No\tTest1\tTest2\tTest3\tAverage\tHighest Marks\n");
    printf("-----\n");
    for(i=0;i<n;i++)
    {
        printf("%s\t",s[i].name);
        printf("%d\t",s[i].roll_no);
        printf("%d\t",s[i].Test1);
        printf("%d\t",s[i].Test2);
        printf("%d\t",s[i].Test3);
        printf("%2f\t",s[i].avg);
        printf("%d\t",s[i].highest);
        printf("\n");
    }
}
void Highest(int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        if(s[i].Test1>s[i].Test2 && s[i].Test1>s[i].Test3)
        {
            s[i].highest=s[i].Test1;
        }
        else if(s[i].Test2>s[i].Test3)
        {
            s[i].highest=s[i].Test2;
        }
        else
    }
}

```

```

        {
            s[i].highest=s[i].Test3;
        }
        s[i].avg+=(s[i].Test1+s[i].Test2+s[i].Test3)/3;
    }
}

void Search(int n)
{
    int ser, notfound=0,i;
    printf("\nEnter the student roll_no to be searched : ");
    scanf("%d",&ser);
    for(i=0;i<n;i++)
    {
        if(s[i].roll_no==ser)
        {
            printf("\nDisplaying Search Information:\n");
            printf("\nName\tRoll_No\tTest1\tTest2\tTest3\tAverage\tHighest
Marks\n");
            printf("-----\n");
            printf("%s\t",s[i].name);
            printf("%d\t",s[i].roll_no);
            printf("%d\t",s[i].Test1);
            printf("%d\t",s[i].Test2);
            printf("%d\t",s[i].Test3);
            printf("%2f\t",s[i].avg);
            printf("%d\t",s[i].highest);
            printf("\n");
        }
        else
        {
            notfound=1;
        }
    }

    if(notfound==0)
    {
        printf("No student record found\n");
    }
    // getch();
}

void main()
{
    int i,n;
    // clrscr();
    printf("Enter number of students: ");
    scanf("%d",&n);
    printf("Enter information of %d students\n",n);
    printf("\nName \tRoll_No\tTest1\tTest2\tTest3\n");

```

```
printf("-----\n");
for(i=0;i<n;i++)
{
    scanf("%s",&s[i].name);
    scanf("%d",&s[i].roll_no);
    scanf("%d",&s[i].Test1);
    scanf("%d",&s[i].Test2);
    scanf("%d",&s[i].Test3);
}
Highest(n);
display(n);
Search(n);
}
```

// 12. a. *DMA*: Store a character string in a block of memory space created by malloc() and then modify it.

```
#include<stdio.h>
// #include<conio.h>
#include<string.h>
void main()
{
    char*buffer;
    // Initial Allocation for Variable called buffer
    buffer=(char*)malloc(7*sizeof(char));
    printf("Buffer of size %d is created\n",7*sizeof(char));
    strcpy(buffer,"MYSORE");
    printf("Buffer contains: %s\n",buffer);
    // Reallocation of buffer
    buffer=(char*)realloc(buffer,25*sizeof(char));
    // After reallocation say buffer size is now modified
    printf("Buffer size is now modified\n");
    printf("New Buffer size is: %d\n",25*sizeof(char));
    printf("Value of Buffer is: %s\n", buffer);
    strcpy(buffer,"BENGALURU");
    printf("Value of Buffer now is: %s\n",buffer);
    free (buffer);
    // getch();
}
```

```
// 12. b. Reverse the elements of an array using pointers.
```

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

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
int array[50],n,i,j,temp;
```

```
int *a;
```

```
a=array;
```

```
printf("Enter size of array:");
```

```
scanf("%d",&n);
```

```
printf("Enter %d Elements: \n",n);
```

```
for(i=0;i<n;i++)
```

```
{
```

```
scanf("%d",&a[i]);
```

```
}
```

```
for(i=0,j=n-1;i<j;i++,j--)
```

```
{
```

```
temp=*(a+i);
```

```
*(a+i)=*(a+j);
```

```
*(a+j)=temp;
```

```
}
```

```
printf("After reversing the array:\n");
```

```
for(i=0;i<n;i++)
```

```
{
```

```
printf("%d\t",a[i]);
```

```
}
```

```
return 0;
```

```
}
```