

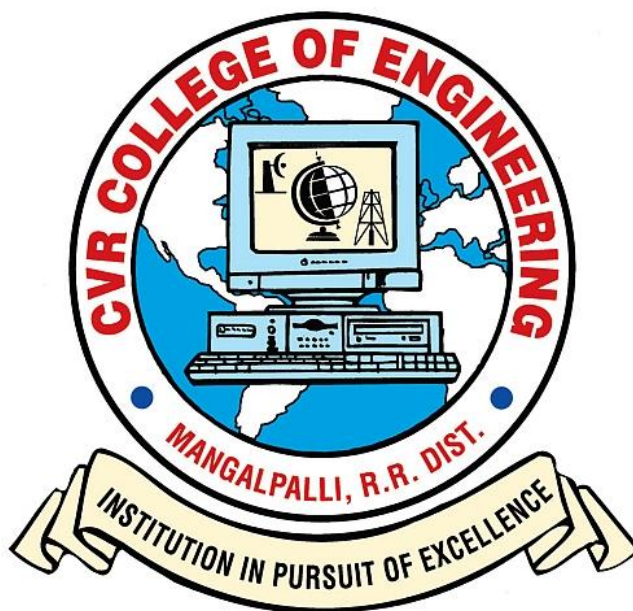
CVR COLLEGE OF ENGINEERING

In Pursuit of Excellence

(An Autonomous Institution, NAAC 'A' Grade)

Vastu Nagar, Mangalpalli (V), Ibrahimpatnam (M),

Ranga Reddy (D), Telangana – 501510



DEPARTMENT OF CSIT

I YEAR – I SEMESTER

65131 – COMPUTER PROGRAMMING LAB

1. Write the algorithm and draw the flow chart to find the roots of a quadratic equation

Algorithm to find all the roots of a quadratic equation:

Step 1. Start

Step 2. Read the coefficients of the equation, a, b and c from the user.

Step 3. Calculate discriminant = $(b * b) - (4 * a * c)$

Step 4. If discriminant > 0:

4.1: Calculate root1 = $(-b + \text{sqrt}(\text{discriminant})) / (2 * a)$

4.2: Calculate root2 = $(-b - \text{sqrt}(\text{discriminant})) / (2 * a)$

4.3: Display “Roots are real and different”

4.4: Display root1 and root2

Step 5: Else if discriminant = 0:

5.1: Calculate root1 = $-b / (2 * a)$

5.2: root2 = root1

5.3: Display “Root are real and equal”

5.4: Display root1 and root2

Step 6. Else:

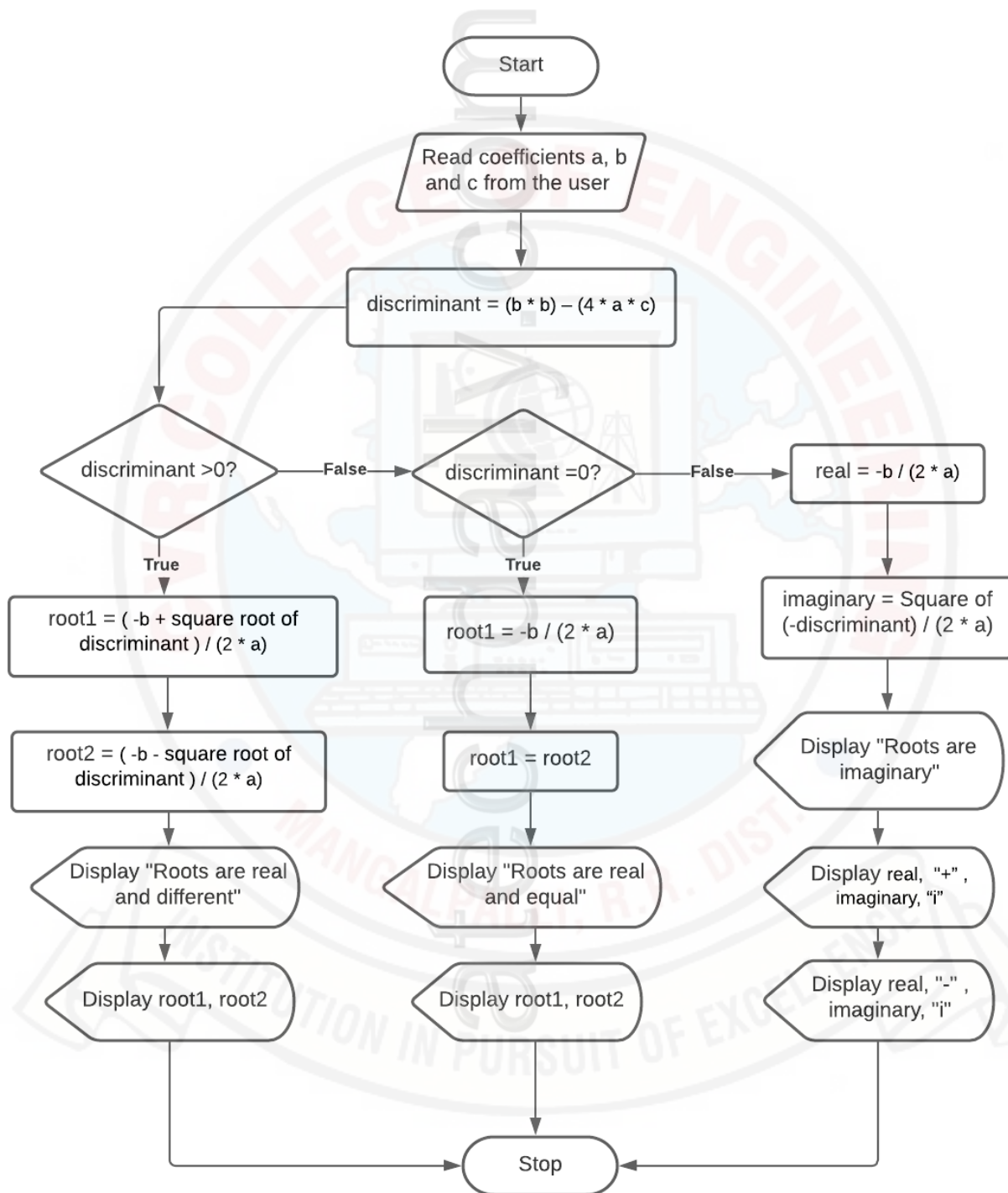
6.1: Calculate real = $-b / (2 * a)$

6.2: Calculate imaginary = $\text{sqrt}(-\text{discriminant}) / (2 * a)$

6.3: Display “Roots are imaginary”

6.4: Display real, “±”, imaginary, “i”

Step 7. Stop

Flowchart to find all the roots of a quadratic equation:

2. Write the algorithm and draw the flow chart to find the sum of digits of a given n digit number**Pseudo Code:**

- Input a Number
- Initialize Sum to zero
- While Number is not zero
 - Get Remainder by Number Mod 10
 - Add Remainder to Sum
 - Divide Number by 10
- Print sum

Detailed Algorithm:

Step 1: Input N

Step 2: Sum = 0

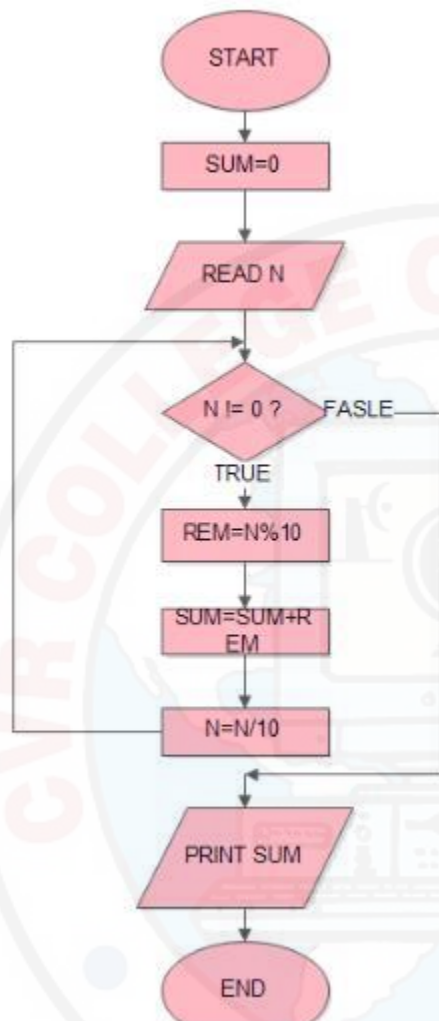
Step 3: While (N != 0)

 Rem = N % 10;

 Sum = Sum + Rem;

 N = N / 10;

Step 4: Print Sum

Flowchart:-

3. Write a C program to explore decimal, octal, hexadecimal, unsigned, unsigned long long formats of integers with printf and scanf functions

Program

```
#include <stdio.h>
main()
{
    char ch = 'B';
    printf("%c\n", ch); //printing character data
    //print decimal or integer data with d and i
    int x = 45, y = 90;
    printf("%d\n", x);
    printf("%i\n", y);

    float f = 12.67;
    printf("%f\n", f); //print float value
    printf("%e\n", f); //print in scientific notation

    // int a = 45;
    // printf("%u\n", a); //print in unsigned format
    // printf("%llu\n", a); //print in unsigned long long form

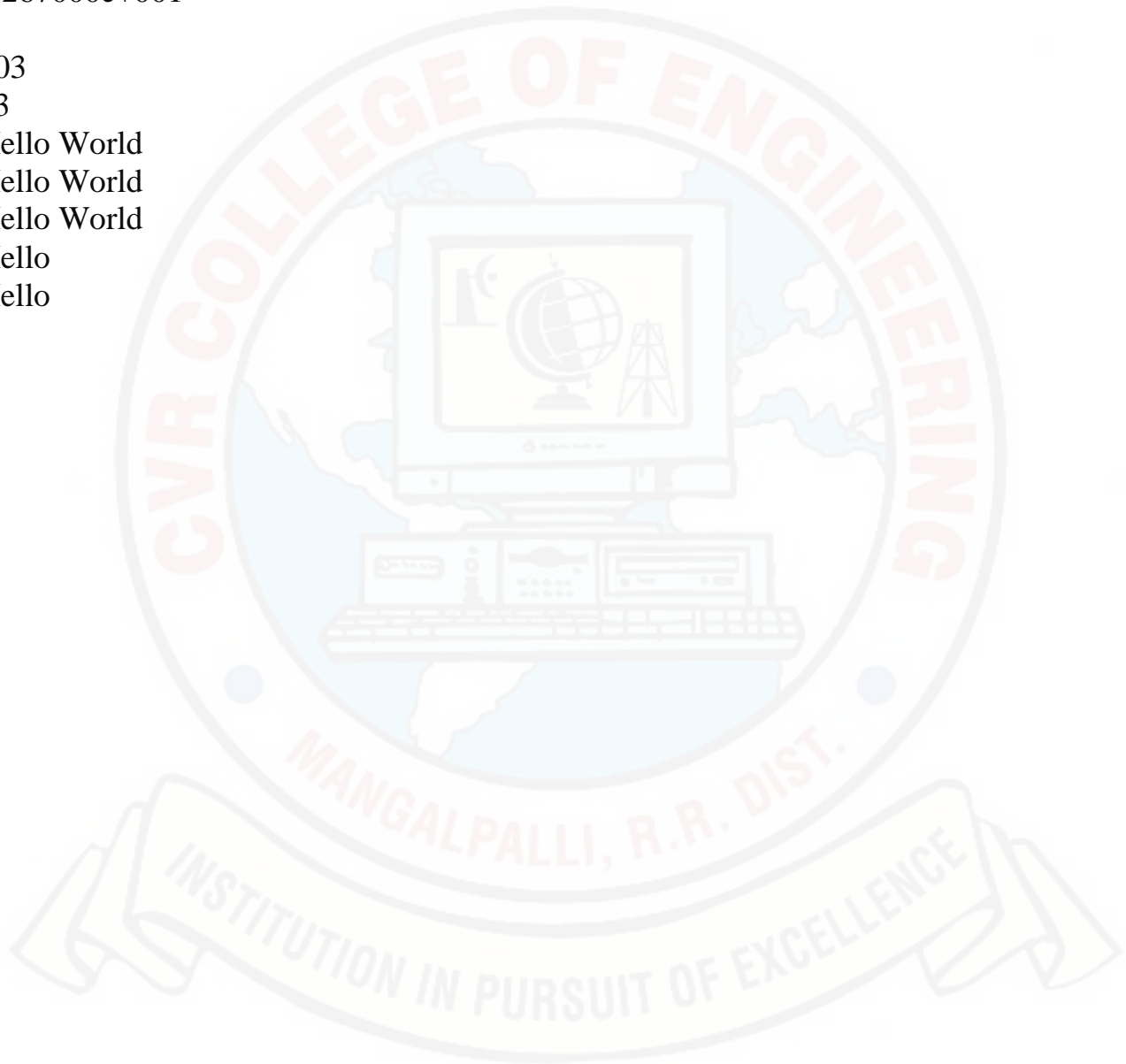
    int b=67;
    printf("%o\n", b); //print in octal format
    printf("%x\n", b); //print in hex format

    char str[] = "Hello World";
    printf("%s\n", str);
    printf("%20s\n", str); //shift to the right 20 characters including the string
    printf("%-20s\n", str); //left align
    printf("%20.5s\n", str); //shift to the right 20 characters including the string, and
                                print string up to 5 character
    printf("%-20.5s\n", str); //left align and print string up to 5 character
}
```


Output

B
45
90
12.670000
1.267000e+001

103
43
Hello World
Hello World
Hello World
Hello
Hello



4. Write a C program to convert the given temperature in Celsius into Fahrenheit**Program:**

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

void main()
{
    float celsius,fahrenheit;
    clrscr();

    printf("\n Enter the Temperature in Celsius : ");
    scanf("%f",&celsius);

    fahrenheit = (1.8 * celsius) + 32;
    printf("\n Temperature in Fahrenheit : %f ",fahrenheit);

    getch();
}
```

Output:

Enter the Temperature in Celsius : 31

Temperature in Fahrenheit : 87.800003

5. Write a C Program to generate the following pattern.

1
123
12345
1234567
123456789
1234567
12345
123
1

Program:

```
#include<stdio.h >
int main()
{
    int n, a;
    printf("Enter number of rows: ");
    scanf("%d",&n);

    // for first half portion
    // from top to bottom
    for(int i=1; i<=n; i++)
    {
        // In each iteration a will start from 1
        a = 1;

        // print space
        for(int j=i; j <= n; j++)
        {
            printf(" ");
        }

        // print digit
        for(int k=1; k <= 2*i-1; k++)
        {
            printf("%d",a++);
        }

        // new line
```

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

// for second portion
for(int i=n-1; i>=1; i--)
{
    // In each iteration a will start from 1
    a=1;

    // print space
    for(int j=n; j>=i; j--)
    {
        printf(" ");
    }

    // print digit
    for(int k=1; k<=2*i-1; k++)
    {
        printf("%d",a++);
    }

    // new line
    printf("\n");
}
return 0;
}
```

Output:

Enter N value: 5

```
1
123
12345
1234567
123456789
1234567
12345
123
1
```

6. Write a simple calculator program which reads operand1, operator and operand2 as input and displays the result.

Program 1:

```
#include <stdio.h>

int main()
{
    char op;
    float num1, num2, result=0.0f;

    /* Print welcome message */
    printf("WELCOME TO SIMPLE CALCULATOR\n");
    printf("-----\n");
    printf("Enter [number 1] [+ - * /] [number 2]\n");

    /* Input two number and operator from user */
    scanf("%f %c %f", &num1, &op, &num2);

    /* Switch the value and perform action based on operator*/
    switch(op)
    {
        case '+':
            result = num1 + num2;
            break;

        case '-':
            result = num1 - num2;
            break;

        case '*':
            result = num1 * num2;
            break;

        case '/':
            result = num1 / num2;
            break;

        default:
```

```
        printf("Invalid operator");
    }

    /* Prints the result */
    printf("%.2f %c %.2f = %.2f", num1, op, num2, result);

    return 0;
}
```

Output

WELCOME TO SIMPLE CALCULATOR

Enter [number 1] [+ - * /] [number 2] –

22 * 6

22.00 * 6.00 = 132.00

7. Write a C program to find the sum of individual digits of a positive integer**Program:**

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main ()
{
    int number = 0, digit = 0, sumOfDigits = 0;
    clrscr();
    printf("Enter any number\n ");
    scanf("%d", &number);
    while (number != 0)
    {
        digit = number % 10;
        sumOfDigits = sumOfDigits + digit;
        number = number / 10;
    }
    printf ("Sum of individual digits of a given number is %d", sumOfDigits);
    getch();
}
```

Output:

Enter any number

1234

Sum of individual digits of a given number is 10

8. Write the calculator program given in question no 6 to run the operations until user's choice is exit.

Program:

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

int main()
{
    // declaration of local variable op;
    int op, n1, n2;
    float res;
    char ch;
    do
    {
        // displays the multiple operations of the C Calculator
        printf (" Select an operation to perform the calculation in C Calculator: ");
        printf (" \n 1 Addition \t \t 2 Subtraction \n 3 Multiplication \t 4 Division \n 5 Square
                \t \t 6 Square Root \n 7 Exit \n \n Please, Make a choice ");
        scanf ("%d", &op); // accepts a numeric input to choose the operation

        // use switch statement to call an operation
        switch (op)
        {
            case 1:
                // Add two numbers
                printf (" You chose: Addition");
                printf ("\n Enter First Number: ");
                scanf (" %d", &n1);
                printf (" Enter Second Number: ");
                scanf (" %d", &n2);
                res = n1 + n2; // Add two numbers
                printf (" Addition of two numbers is: %.2f", res);
                break; // break the function

            case 2:
                // Subtract two numbers
                printf (" You chose: Subtraction");
                printf ("\n Enter First Number: ");
                scanf (" %d", &n1);
```



```
printf (" Enter Second Number: ");
scanf (" %d", &n2);
res = n1 - n2; // subtract two numbers
printf (" Subtraction of two numbers is: %.2f", res);
break; // break the function
```

case 3:

```
// Multiplication of the numbers
printf (" You chose: Multiplication");
printf ("\n Enter First Number: ");
scanf (" %d", &n1);
printf (" Enter Second Number: ");
scanf (" %d", &n2);
res = n1 * n2; // multiply two numbers
printf (" Multiplication of two numbers is: %.2f", res);
break; // break the function
```

case 4:

```
// Division of the numbers
printf (" You chose: Division");
printf ("\n Enter First Number: ");
scanf (" %d", &n1);
printf (" Enter Second Number: ");
scanf (" %d", &n2);
if (n2 == 0)
{
    printf (" \n Divisor cannot be zero. Please enter another value ");
    scanf ("%d", &n2);
}
res = n1 / n2; // divide two numbers
printf (" Division of two numbers is: %.2f", res);
break; // break the function
```

case 5:

```
// getting square of a number
printf (" You chose: Square");
printf ("\n Enter First Number: ");
scanf (" %d", &n1);

res = n1 * n1; // get square of a number
printf (" Square of %d number is: %.2f", n1, res);
break; // break the function
```

```
case 6:
    // getting the square root of the number
    printf (" You chose: Square Root");
    printf ("\n Enter First Number: ");
    scanf (" %d", &n1);

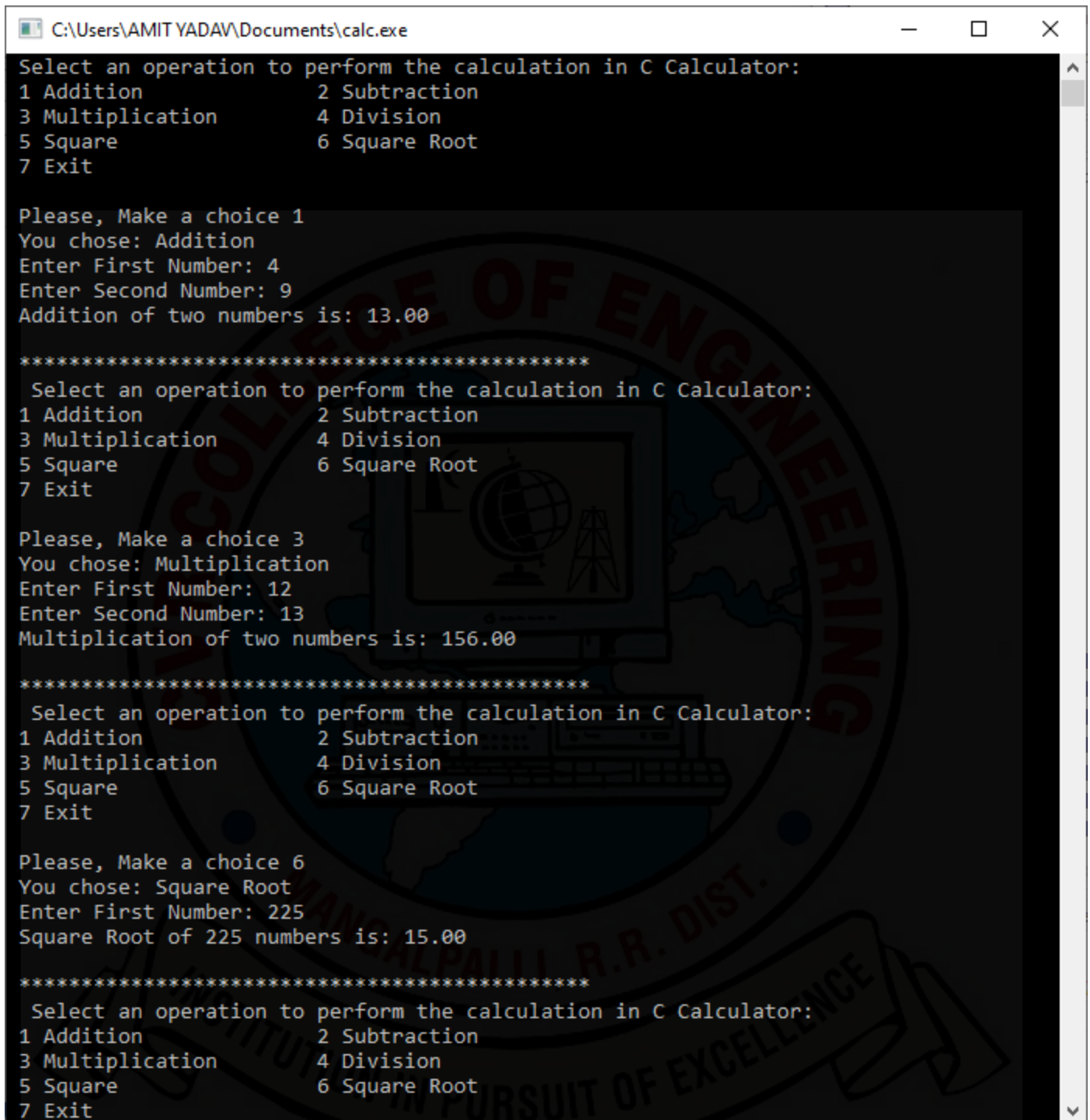
    res = sqrt(n1); // use sqrt() function to find the Square Root
    printf (" Square Root of %d numbers is: %.2f", n1, res);
    break; // break the function

case 7:
    printf (" You chose: Exit");
    exit(0);
    break; // break the function

default:
    printf(" Something is wrong!! ");
    break;
}
printf (" \n \n ***** \n ");
} while (op != 7);

return 0;
}
```

Output:



```
C:\Users\AMIT YADAV\Documents\calc.exe
Select an operation to perform the calculation in C Calculator:
1 Addition          2 Subtraction
3 Multiplication    4 Division
5 Square           6 Square Root
7 Exit

Please, Make a choice 1
You chose: Addition
Enter First Number: 4
Enter Second Number: 9
Addition of two numbers is: 13.00

*****
Select an operation to perform the calculation in C Calculator:
1 Addition          2 Subtraction
3 Multiplication    4 Division
5 Square           6 Square Root
7 Exit

Please, Make a choice 3
You chose: Multiplication
Enter First Number: 12
Enter Second Number: 13
Multiplication of two numbers is: 156.00

*****
Select an operation to perform the calculation in C Calculator:
1 Addition          2 Subtraction
3 Multiplication    4 Division
5 Square           6 Square Root
7 Exit

Please, Make a choice 6
You chose: Square Root
Enter First Number: 225
Square Root of 225 numbers is: 15.00

*****
Select an operation to perform the calculation in C Calculator:
1 Addition          2 Subtraction
3 Multiplication    4 Division
5 Square           6 Square Root
7 Exit
```

9. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

Program 1:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n, i, j, count;
    clrscr();
    printf("Prime no.series\n");
    printf("Enter any number\n");
    scanf("%d", &n);
    printf("The prime numbers between 1 to %d\n",n);
    for(i = 1; i <= n; i++)
    {
        count = 0;
        for(j = 1; j <=i; j++)
            if(i % j == 0)
            {
                count++;
            }
        if(count == 2)
        {
            printf("%d\t", i);
        }
    }
    getch();
}
```

Output:

```
Prime no. series
Enter any number
10
The prime numbers between 1 to 10
2 3 5 7
```

10. A number is said to be Armstrong if the number is equivalent to the sum of cubes of its digits. Write a C program to check whether a given number is Armstrong or not.

Program 1:

```
#include <stdio.h>
int main()
{
    int num, originalNum, remainder, result = 0;
    printf("Enter a three-digit integer: ");
    scanf("%d", &num);
    originalNum = num;

    while (originalNum != 0)
    {
        // remainder contains the last digit
        remainder = originalNum % 10;

        result += remainder * remainder * remainder;

        // removing last digit from the original number
        originalNum /= 10;
    }

    if (result == num)
        printf("%d is an Armstrong number.", num);
    else
        printf("%d is not an Armstrong number.", num);

    return 0;
}
```

Output:

Enter a three-digit integer: 371
371 is an Armstrong number.

Program:

```
#include<stdio.h>
int main()
{
    int n,r,sum=0,temp;
    printf("enter the number=");
    scanf("%d",&n);
    temp=n;
    while(n>0)
    {
        r=n%10;
        sum=sum+(r*r*r);
        n=n/10;
    }
    if(temp==sum)
        printf("armstrong number ");
    else
        printf("not armstrong number");
    return 0;
}
```

Output:

enter the number=153
armstrong number

enter the number=5
not armstrong number

11. Write a C program to define the macros SUM (a, b), SQUARE (a) and SQUARE (SUM (a, b)) and print the results.

Program:

```
/* C program to find square and cube of a number using macro */

#include <stdio.h>

// Define macro to find SUM (a,b) and SQUARE(a)
#define SUM(a,b) (a + b)
#define SQUARE(a) (a * a)

int main()
{
    int num1, num2, num3;

    // Input two numbers from user
    printf("Enter any two numbers to find sum: ");
    scanf("%d%d", &num1, &num2);

    // Calculate and print sum using macro
    printf("SUM(%d, %d) = %d\n", num1, num2, SUM(num1, num2));

    // Input a number from user
    printf("Enter any number to find square: ");
    scanf("%d", &num3);

    // Calculate and print square
    printf("SQUARE(%d) = %d\n", num3, SQUARE(num3));

    // Calculate and print square(sum(a,b))
    printf("SQUARE(SUM(%d, %d) = %d\n", num1, num2, SQUARE(SUM(num1, num2)));

    return 0;
}
```


Output:

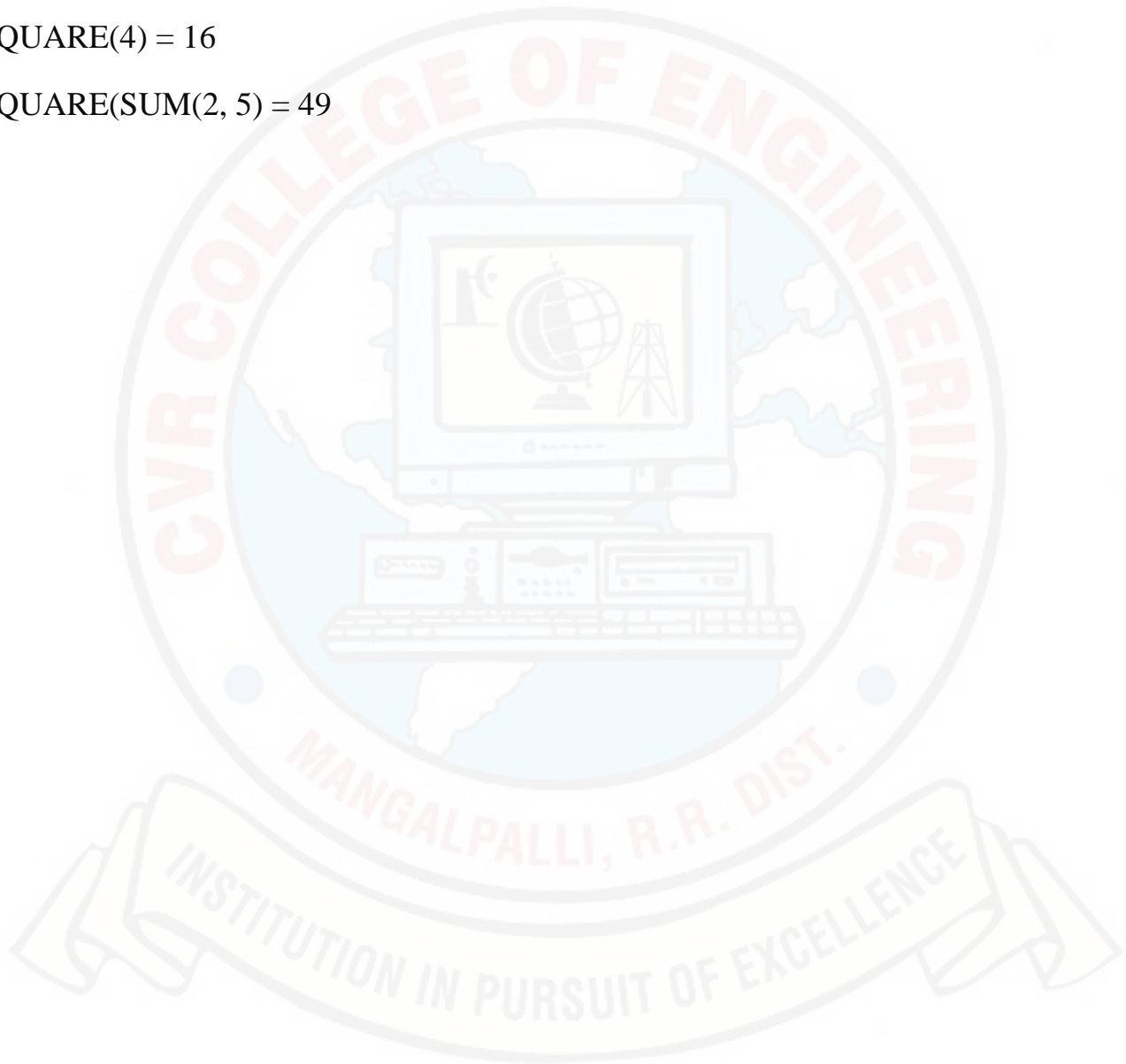
Enter any two numbers to find sum: 2 5

SUM(2, 5) = 7

Enter any number to find square: 4

SQUARE(4) = 16

SQUARE(SUM(2, 5)) = 49



12. Write a C program to illustrate functions without parameters and without return type, without parameters and with return type, with parameters and without return type and with parameters and with return type.

Program:

1. C Program using Function with No argument and No Return value Example

```
#include<stdio.h>

// Declaration
void Addition();

void main()
{
    printf("\n ..... \n");

    Addition();
}

void Addition()
{
    int Sum, a = 10, b = 20;
    Sum = a + b;

    printf("\n Sum of a = %d and b = %d is = %d", a, b, Sum);
}
```

Output :

.....

Sum of a = 10 and b = 20 is = 30

2. C Program using Function with no argument and with Return value

```
#include<stdio.h>

int Multiplication();

int main()
{
```

```
int Multi;

Multi = Multiplication();
printf("\n Multiplication of a and b is = %d \n", Multi );

return 0;
}

int Multiplication()
{
    int Multi, a = 20, b = 40;

    Multi = a * b;

    return Multi;
}
```

Output :

Multiplication of a and b is = 800

3. C Program using Function with argument and No Return value

```
#include<stdio.h>

void Addition(int, int);

void main()
{
    int a, b;

    printf("\n Please Enter two integer values \n");
    scanf("%d %d",&a, &b);

    //Calling with dynamic values
    Addition(a, b);
}

void Addition(int a, int b)
{
    int Sum;

    Sum = a + b;

    printf("\n Addition of %d and %d is = %d \n", a, b, Sum);
}
```

```
}
```

Output :

Please Enter two integer values

40

90

Addition of 40 and 90 is = 130

4. C Program using Function with argument and Return value

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

```
int Multiplication(int, int);
```

```
int main()
```

```
{
```

```
    int a, b, Multi;
```

```
    printf("\n Please Enter two integer values \n");
```

```
    scanf("%d %d",&a, &b);
```

```
    //Calling the with dynamic values
```

```
    Multi = Multiplication(a, b);
```

```
    printf("\n Multiplication of %d and %d is = %d \n", a, b, Multi);
```

```
    return 0;
```

```
}
```

```
int Multiplication(int a, int b)
```

```
{
```

```
    int Multi;
```

```
    Multi = a * b;
```

```
    return Multi;
```

```
}
```

Output :

Please Enter two integer values

30

60

Multiplication of 30 and 60 is = 1800

13. Write a C function to calculate the sine series sum $1 - x^3/3! + x^5/5! - \dots$ and call the function.

Program:

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

double factorial(int);
void calc(float, float*);

int main()
{
    int x;
    float radian, result = 0;

    printf("Enter value of x in degrees\n");
    scanf("%d", &x);

    radian = x * (3.14159 / 180.0); // Convert Degree To Radian

    calc(radian, &result);

    printf("Sin(%d) = %f\n", x, result);

    return 0;
}

void calc(float num, float *res)
{
    int count, n = 1, sign = 1;

    for(count = 1; (n <= 10); count += 2)
    {
        *res += sign * ( pow(num, count) / factorial(count) );
        n += 1;
        sign *= -1;
    }
}
```

```
double factorial(int num)
{
    int count;
    double sum = 1;

    for(count = 1; count <= num; count++)
    {
        sum *= count;
    }
    return(sum);
}
```

Output 1:

Enter value of x is degrees

0

Sin(0) = 0.000000

Output 2:

Enter value of x is degrees

30

Sin(30) = 0.500000

Output 3:

Enter value of x is degrees

45

Sin(45) = 0.707106

Output 4:

Enter value of x is degrees

60

Sin(60) = 0.866025

Output 5:

Enter value of x is degrees

90

Sin(90) = 1.000000

14. Write a C program in which a recursive and non-recursive functions are called to compute factorial values based on user's choice

Program:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n, a, b;
    clrscr();
    printf("Enter any number\n");
    scanf("%d", &n);
    printf("1. Factorial using Recursion\n");
    printf("2. Factorial without using Recursion\n");
    printf("Enter your Choice:\n");
    scanf("%d", &ch);
    if(ch==1)
    {
        a = recfactorial(n);
        printf("The factorial of a given number using recursion is %d \n", a);
    }
    else if(ch==2)
    {
        b = nonrecfactorial(n);
        printf("The factorial of a given number using non-recursion is %d ", b);
    }
    else
        printf("Invalid Choice\n");

    getch();
}

int recfactorial(int x)
{
    int f;
    if(x == 0)
    {
        return(1);
    }
}
```

```
else
{
    f = x * recfactorial(x - 1);
    return(f);
}
}
```

```
int nonrecfactorial(int x)
{
    int i, f = 1;
    for(i = 1; i <= x; i++)
    {
        f = f * i;
    }
    return(f);
}
```

Output 1:

Enter any number

4

1. Factorial using Recursion
2. Factorial without using Recursion

Enter your Choice:

1

The factorial of a given number using recursion is 24

Output 2:

Enter any number

5

1. Factorial using Recursion
2. Factorial without using Recursion

Enter your Choice:

2

The factorial of a given number using non-recursion is 120

15. Write a C program in which a recursive and non-recursive functions are called to generate Fibonacci series based on user's choice**Program:**

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>

void fib(int n)
{
    int a = 0, b = 1, c, count = 3;
    if(n == 1)
        printf("0");
    else if(n == 2)
        printf("0 1");
    else
    {
        printf("0 1 ");
        while(count <= n)
        {
            c = a + b;
            printf("%d ", c);
            a = b;
            b = c;
            count++;
        }
    }
}

int rfib(int n)
{
    if(n == 1)
        return 0;
    else if(n == 2)
        return 1;
    else
    {
        return rfib(n - 1) + rfib(n - 2);
    }
}
```

```
    }  
}  
  
int main(int argc, char **argv)  
{  
    int n, count = 3;  
    printf("Enter a number: ");  
    scanf("%d", &n);  
    printf("\nNon-recursive fibonacci sequence upto %d terms: \n", n);  
    fib(n);  
    printf("\nRecursive fibonacci sequence upto %d terms: \n", n);  
    if(n == 1)  
        printf("0");  
    else if(n == 2)  
        printf("0 1");  
    else  
    {  
        printf("0 1 ");  
        while(count <= n)  
        {  
            printf("%d ", rfib(count));  
            count++;  
        }  
    }  
    getch();  
    return 0;  
}
```

Output:

Enter a number: 10

Non-recursive fibonacci sequence upto 10 terms:

0 1 1 2 3 5 8 13 21 34

Recursive fibonacci sequence upto 10 terms:

0 1 1 2 3 5 8 13 21 34

16. Write a C program to illustrate Command-Line Arguments.

Program:

```
#include<stdio.h>

int main(int argc, char *argv[])
{
    int counter;
    printf("Program Name Is: %s",argv[0]);
    if(argc==1)
        printf("\nNo Extra Command Line Argument Passed Other Than Program Name");
    if(argc>=2)
    {
        printf("\nNumber Of Arguments Passed: %d",argc);
        printf("\n----Following Are The Command Line Arguments Passed----");
        for(counter=0;counter<argc;counter++)
            printf("\nargv[%d]: %s",counter,argv[counter]);
    }
    return 0;
}
```

Output:

Output in different scenarios:

1. **Without argument:** When the above code is compiled and executed without passing any argument, it produces following output.

```
$ ./a.out
Program Name Is: ./a.out
No Extra Command Line Argument Passed Other Than Program Name
```

2. **Three arguments :** When the above code is compiled and executed with a three arguments, it produces the following output.

```
$ ./a.out First Second Third
Program Name Is: ./a.out
Number Of Arguments Passed: 4
----Following Are The Command Line Arguments Passed----
argv[0]: ./a.out
```

```
argv[1]: First  
argv[2]: Second  
argv[3]: Third
```

- 3. Single Argument :** When the above code is compiled and executed with a single argument separated by space but inside double quotes, it produces the following output.

```
$ ./a.out "First Second Third"  
Program Name Is: ./a.out  
Number Of Arguments Passed: 2  
---Following Are The Command Line Arguments Passed---  
argv[0]: ./a.out  
argv[1]: First Second Third
```

- 4. Single argument in quotes separated by space :** When the above code is compiled and executed with a single argument separated by space but inside single quotes, it produces the following output.

```
$ ./a.out 'First Second Third'  
Program Name Is: ./a.out  
Number Of Arguments Passed: 2  
---Following Are The Command Line Arguments Passed---  
argv[0]: ./a.out  
argv[1]: First Second Third
```


17. Write a C program to find the Sum of the Elements of a given List (Array).**Program:**

```
#include<stdio.h>

int main()
{
    //let's assume the maximum array size as 100.
    //initialize sum as 0. Otherwise, it will take some garbage value.
    int arr[100], size, i, sum = 0;

    //Get size input from user
    printf("Enter array size\n");
    scanf("%d",&size);

    //Get all elements using for loop and store it in array
    printf("Enter array elements\n");
    for(i = 0; i < size; i++)
        scanf("%d",&arr[i]);

    //add all elements to the variable sum.
    for(i = 0; i < size; i++)
        sum = sum + arr[i]; // same as sum += arr[i];

    //print the result
    printf("Sum of the array = %d\n",sum);

    return 0;
}
```

Output:

```
~/O9Mk8UatKEF$ gcc main.c
```

```
~/O9Mk8UatKEF$ ./a.out
```

```
Enter array size
```

```
5
```

Enter array elements

12

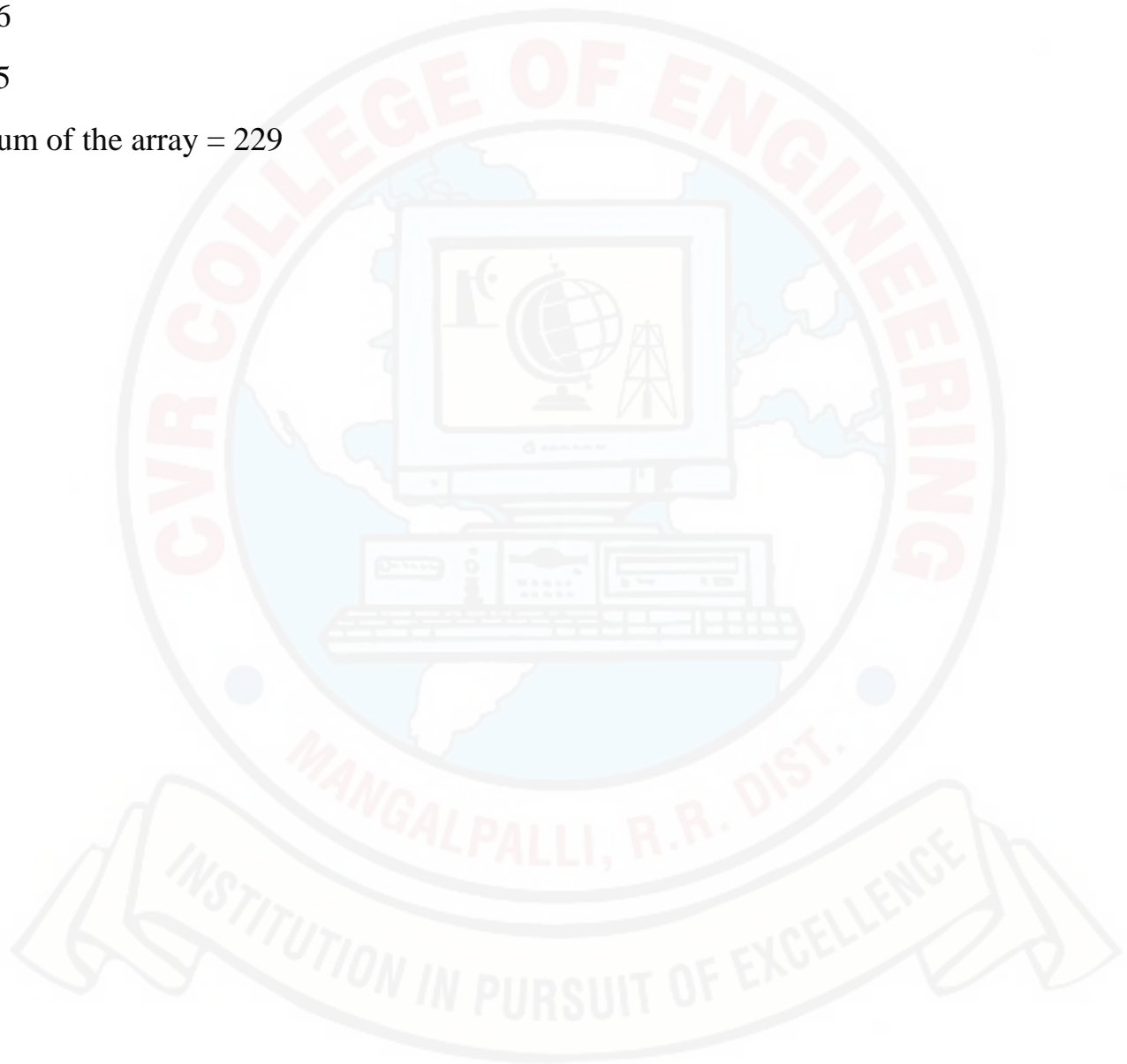
45

31

76

65

Sum of the array = 229



18. Write a C program to implement two separate functions which return the minimum and maximum values of a given array-list and call these functions.

Program:

```
/* C program to find the maximum and minimum element in an array */
```

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

```
int main()
```

```
{  
    int n;  
    printf("Enter number of elements in array: ");  
    scanf("%d",&n);  
    int numbers[n];  
    int i;  
    int min, max ;  
    printf("Enter %d numbers : ", n);  
    for (i = 0; i < n; i++)  
    {  
        scanf("%d", &numbers[i]);  
    }  
    min = minimum(numbers, n);  
    max= maximum(numbers, n);  
    printf("\nMinimum number in the array is : %d\n", min);  
    printf("\nMaximum number is the array is %d\n", max);  
    return 0;  
}
```

```
int minimum(int numbers[], int n)
```

```
{  
    int min = numbers[0];  
    int i;  
    for (i = 1; i <= n; i++)  
    {  
        if (min > numbers[i])  
            min = numbers[i];  
    }  
    return min;  
}
```

```
int maximum(int numbers[], int n)
{
    int max = numbers[0];
    int i;
    for (i = 1; i <= n; i++)
    {
        if (max < numbers[i])
            max = numbers[i];
    }
    return max;
}
```

Output:

Enter number of elements in array: 6

Enter 6 numbers :

4
6
1
2
5
3

Minimum number in the array is : 1

Maximum number in the array is : 6

19. Write a C program to find the transpose of a given input matrix (read the dimensions of matrix too as input).

Program:

```
#include <stdio.h>
int main()
{
    int a[10][10], transpose[10][10], r, c;
    printf("Enter rows and columns: ");
    scanf("%d %d", &r, &c);

    // assigning elements to the matrix
    printf("\nEnter matrix elements:\n");
    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j)
        {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }

    // printing the matrix a[][]
    printf("\nEnter matrix: \n");
    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j)
        {
            printf("%d ", a[i][j]);
            if (j == c - 1)
                printf("\n");
        }

    // computing the transpose
    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j)
        {
            transpose[j][i] = a[i][j];
        }

    // printing the transpose
    printf("\nTranspose of the matrix:\n");
```

```
for (int i = 0; i < c; ++i)
for (int j = 0; j < r; ++j)
{
    printf("%d ", transpose[i][j]);
    if (j == r - 1)
        printf("\n");
}
return 0;
}
```

Output :

Enter rows and columns: 2 3

Enter matrix elements:

Enter element a11: 1

Enter element a12: 4

Enter element a13: 0

Enter element a21: -5

Enter element a22: 2

Enter element a23: 7

Entered matrix:

1 4 0

-5 2 7

Transpose of the matrix:

1 -5

4 2

0 7

20. Write a C program to implement two separate functions for finding the sum and product of matrices and call these functions.

Program:

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

// main function
int main()
{
    // matrix
    int a[][3] = { {5,6,7}, {8,9,10}, {3,1,2} };
    int b[][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
    int c[3][3];

    // print both matrix
    printf("First Matrix:\n");
    display(a);
    printf("Second Matrix:\n");
    display(b);

    //Sum of Two Matrices
    add(a, b, c);
    printf("Sum of matrix: \n");
    display(c);

    // Product of Two Matrices
    multiply(a, b, c);
    printf("Multiplication of matrix: \n");
    display(c);

    return 0;
}

// function to add two 3x3 matrix
void add(int m[3][3], int n[3][3], int sum[3][3])
{
    for(int i=0;i<3;i++)
        for(int j=0;j<3;j++)
```

```
        sum[i][j] = m[i][j] + n[i][j];
    }

// function to multiply two 3x3 matrix
void multiply(int m[3][3], int n[3][3], int result[3][3])
{
    for(int i=0; i < 3; i++)
    {
        for(int j=0; j < 3; j++)
        {
            result[i][j] = 0; // assign 0
            // find product
            for (int k = 0; k < 3; k++)
            {
                result[i][j] += m[i][k] * n[k][j];
            }
        }
    }
}

// function to display 3x3 matrix
void display(int matrix[3][3])
{
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            printf("%d\t",matrix[i][j]);
        }
        printf("\n"); // new line
    }
}
```

Output :

First Matrix:

5	6	7
8	9	10
3	1	2

Second Matrix:

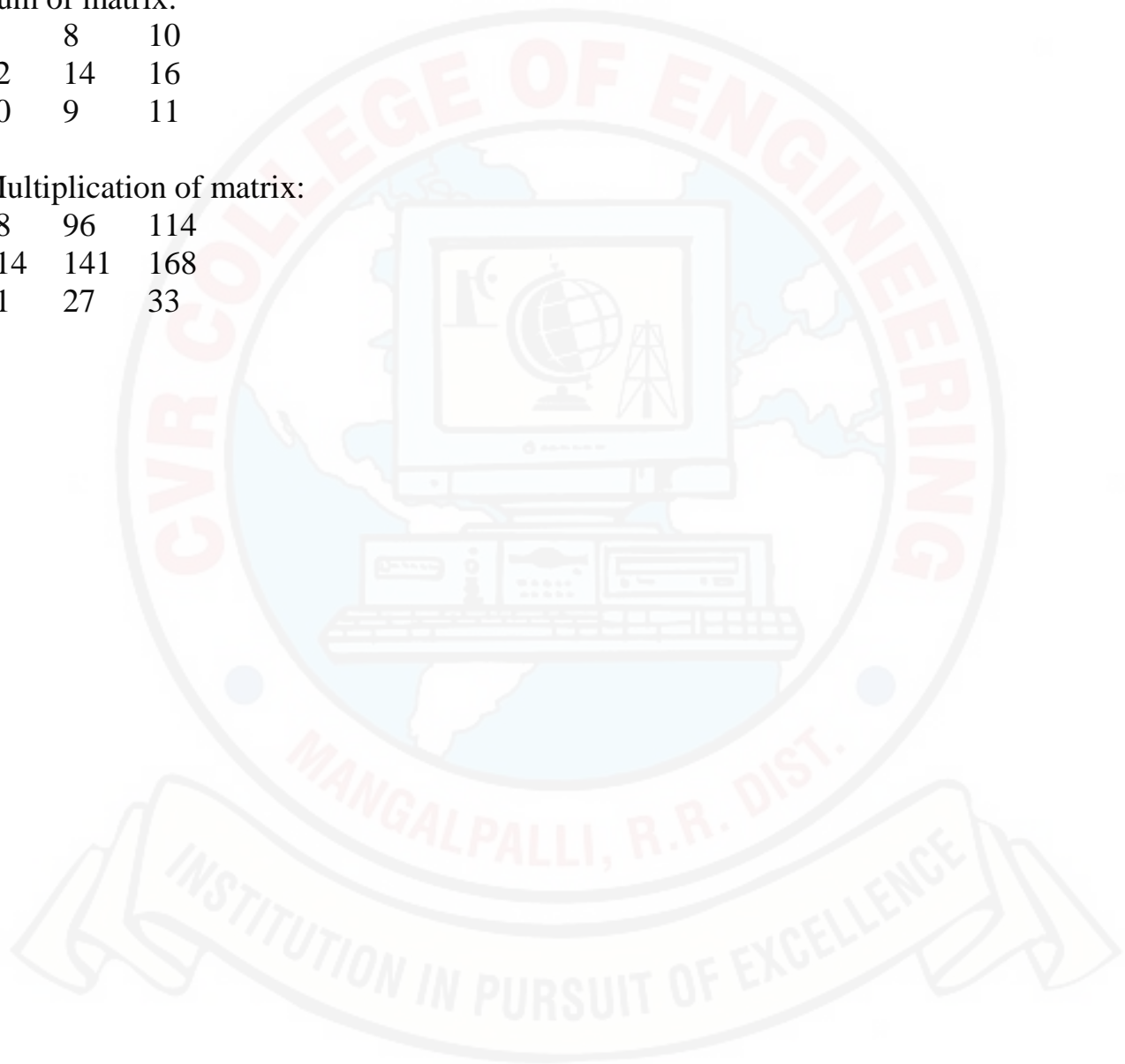
1	2	3
4	5	6
7	8	9

Sum of matrix:

6	8	10
12	14	16
10	9	11

Multiplication of matrix:

78	96	114
114	141	168
21	27	33



21. Write a C function to exchange the values of given two variables and call the function (using pointers).

Program:

```
#include<stdio.h>

void swap(int*, int*);

int main()
{
    int a, b;

    printf("Enter values for a and b\n");
    scanf("%d%d", &a, &b);

    printf("\n\nBefore swapping: a = %d and b = %d\n", a, b);

    swap(&a, &b);

    printf("\n\nAfter swapping: a = %d and b = %d\n", a, b);

    return 0;
}

void swap(int *x, int *y)
{
    int temp;

    temp = *x;
    *x = *y;
    *y = temp;
}
```

Output 1 :

```
Enter values for a and b
100
200
```

Before swapping: $a = 100$ and $b = 200$

After swapping: $a = 200$ and $b = 100$

Output 2 :

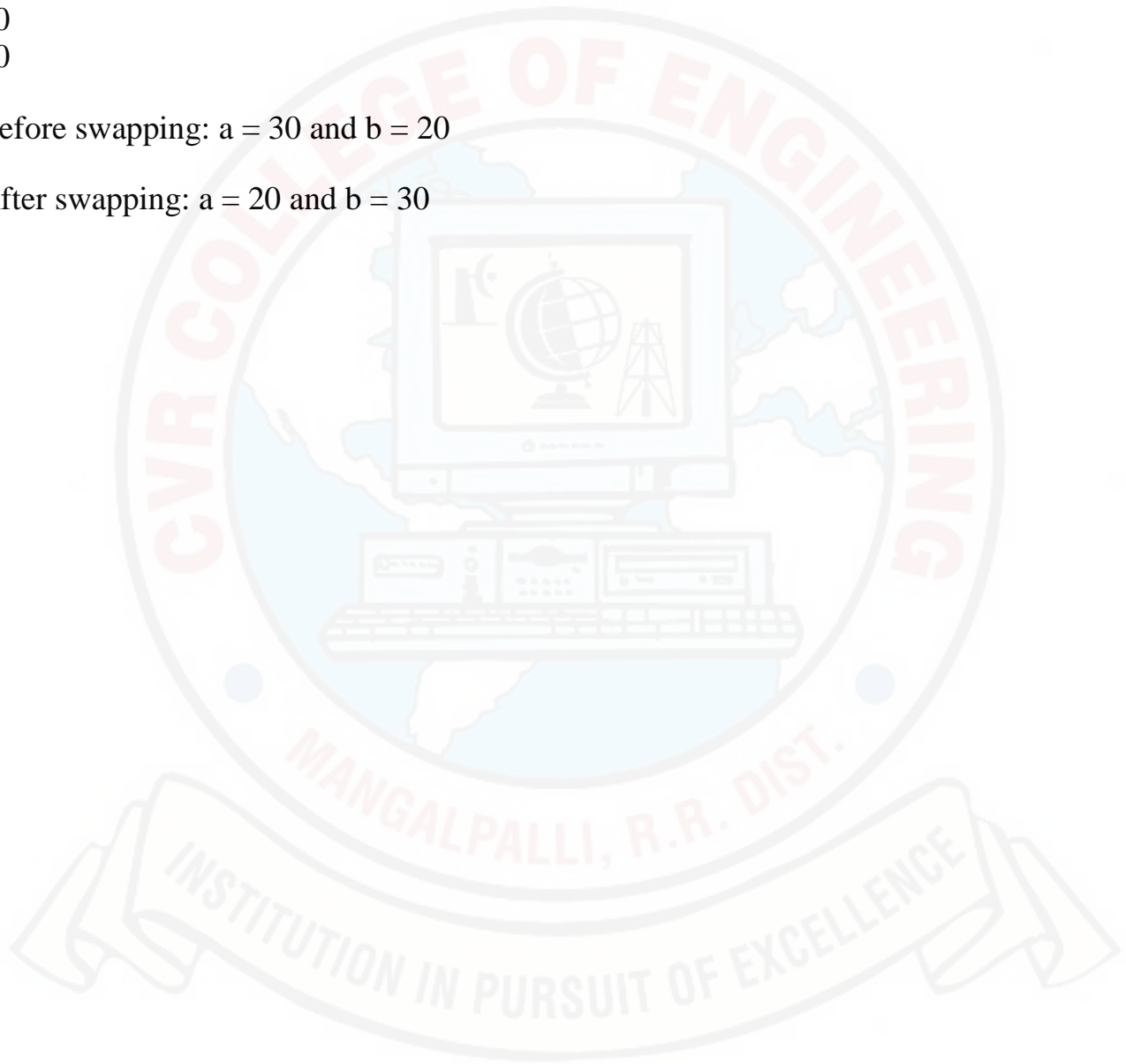
Enter values for a and b

30

20

Before swapping: $a = 30$ and $b = 20$

After swapping: $a = 20$ and $b = 30$



22. Write a C program to Implement two separate C functions to perform insertion of an element and deletion of an element operations on an array at a specified position (pass the array and its size as pointers).

Program:

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

void insert(int a[], int size);
void delete(int a[], int size);

int a[100];
int element, i, loc, size, n, j, choice;

int main()
{
    printf("Enter the size of an array\n");
    scanf("%d",&size);
    printf("Enter %d array elements\n",size);
    for(i=0;i<size;i++)
    {
        scanf("%d", &a[i]);
    }

    printf("Program to Insert and Delete an Element in an Array using
    switchcase\n");
    printf("1. Inserting an Element in an Array\n");
    printf("2. Deleting an Element in an Array\n");
    printf("Select your choice : ");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1:
            insert (a, size);
            break;
        case 2:
            delete(a,size);
            break;
        default:
```



```
        printf("Wrong choice, Please try again later");
    }
    return 0;
}
```

```
void insert(int a[], int size)
{
    printf("List before Insertion: ");
    for(i=0;i<size;i++)
    {
        printf("%d ",a[i]);
    }
    printf("\nEnter an element to insert\n");
    scanf("%d",&element);
    printf("Enter a position to insert an element %d\n",element);
    scanf("%d",&loc);
    loc--;
    for(i=size-1;i>=loc;i--)
    {
        a[i+1]=a[i];
    }
    a[loc]=element;
    printf("List after Insertion: ");
    for(i=0;i<size+1;i++)
    {
        printf("%d ",a[i]);
    }
}
```

```
void delete(int a[], int size)
{
    printf("List before deletion\n");
    for(i=0;i<size;i++)
    {
        printf("%d ",a[i]);
    }
    printf("\nEnter an element to delete\n");
    scanf("%d",&n);
    for(i=0;i<size;i++)
    {
```

```

        if(a[i]==n)
        {
            for(j=i;j<(size-1);j++)
            {
                a[j]=a[j+1];
            }
            break;
        }
    }
    printf("List after deletion\n");
    for(i=0;i<(size-1);i++)
    {
        printf("%d ",a[i]);
    }
}

```

Output 1 :

Enter the size of an array

4

Enter 4 array elements

1

2

3

4

Program to Insert and Delete an Element in an Array using switch case

1. Inserting an Element in an Array

2. Deleting an Element in an Array

Select your choice: 1

List before Insertion: 1 2 3 4

Enter an element to insert

5

Enter a position to insert an element 5

3

List after Insertion: 1 2 5 3 4

Output 2 :

Enter the size of an array

5

Enter 3 array elements

1
2
3
4
5

Program to Insert and Delete an Element in an Array using switch case

1. Inserting an Element in an Array

2. Deleting an Element in an Array

Select your choice: 2

List before deletion

1 2 3 4 5

Enter an element to delete

3

List after deletion

1 2 4 5

23. Write a C program to create a dynamic list of real numbers where the size of the list is accepted as input, extend its size and release it (use dynamic memory allocation functions).

Program:

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

int main()
{
    int i, max, newSize;
    int *ptr;

    // Input maximum elements of array
    printf("Enter total number of elements: ");
    scanf("%d", &max);

    // Allocate memory for 'max' integer elements using malloc
    ptr = (int *) malloc(max * sizeof(int));

    // If memory not allocated
    if(ptr == NULL)
    {
        printf("Memory is not created!!!");
        exit(0); // Exit from the program
    }

    // Input elements from user
    printf("Enter %d elements: \n", max);
    for (i = 0; i < max; i++)
        scanf("%d", (ptr + i));

    // Reallocate memory
    printf("\nEnter new size of the array: ");
    scanf("%d", &newSize);
    ptr = (int *) realloc(ptr, (newSize * sizeof(int)));

    // Input elements in newly allocated memory
```

```
printf("\nEnter %d elements: \n", (newSize - max));
for (i = max; i < newSize; i++)
    scanf("%d", (ptr + i));

// Print all elements
printf("\nArray elements are:\n");
for (i = 0; i < newSize; i++)
    printf("%d ", *(ptr + i));

// Release allocated memory
free(ptr);

return 0;
}
```

Output :

Enter total number of elements: 5

Enter 5 elements:
10 20 30 40 50

Enter new size of the array: 7

Enter 2 elements: 60 70

Array elements are:
10 20 30 40 50 60 70

24. Write a C program to accept string as input and find its length using a user-defined string length function, reverse the string and check whether the string is palindrome or not.

Program:

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

int main()
{
    char str[100], tmp, rev[100];
    int begin, end;

    //input
    printf("Enter string: ");
    scanf("%s", str);

    //copy str to rev
    strcpy(rev, str); //rev will be used to check palindrome

    //reverse
    begin = 0;
    end = strlen(str) - 1; // -1 because last character is NULL \0
    while(begin < end)
    {
        tmp = str[begin];
        str[begin] = str[end];
        str[end] = tmp;
        begin++;
        end--;
    }

    //output
    printf("Reverse string: %s\n", str);

    //checking palindrome
    if(strcmp(rev, str) == 0)
        printf("%s is palindrome!\n", rev);
    else
```



```
        printf("%s is not palindrome!\n", rev);  
  
    return 0;  
}
```

Output 1:

Enter String: hello

Reverse String: olleh

hello is not Palindrome!

Output 2:

Enter String: malayalam

Reverse String: malayalam

malayalam is Palindrome!

25. Write a C function to read a multi-word string and copy the input string to other string (the destination string must be a dynamically allocated string).

Program:

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

void copy(char *, char *);

void main()
{
    char str1[50], *str2;
    int n;

    printf("\n Enter String 1:");
    scanf("%s", str1);

    printf("\n String 1 is %s", str1);

    n = strlen(str1) + 1; //For Null character at the end
    str2 = (char *) calloc(n, sizeof(str1));

    printf("\n Copying String 1 to String 2:");
    copy(str1, str2);

    printf("\n String 1 is %s and String 2 is %s", str1, str2);

    free(str2);
}

void copy(char *str1, char *str2)
{
    int i = 0;
    for(i=0; str1[i]!='\0'; i++)
    {
        str2[i] = str1[i];
    }
}
```

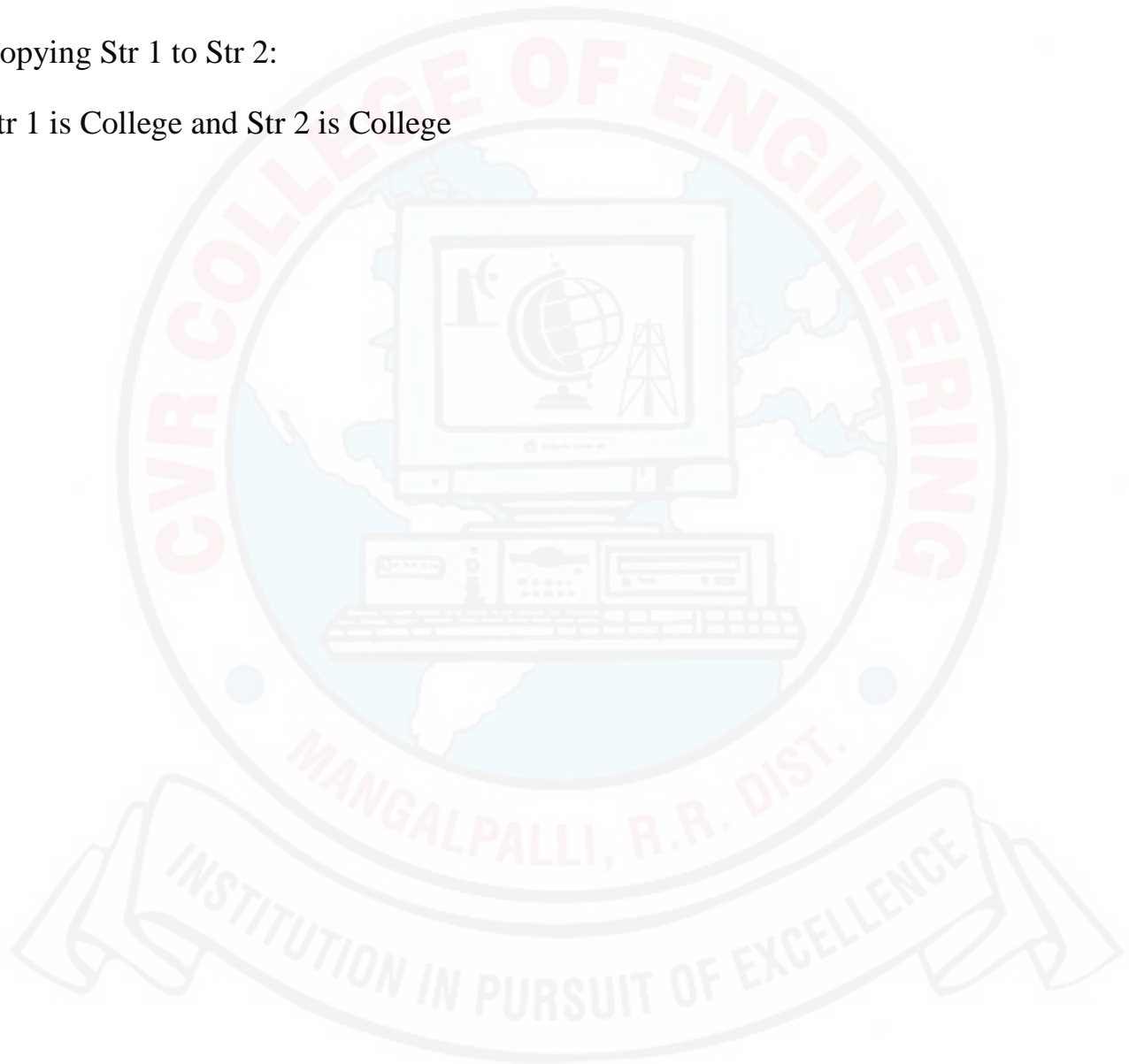
Output:

Enter String 1: College

String 1 is College

Copying Str 1 to Str 2:

Str 1 is College and Str 2 is College



26. Write a C program to create a user defined data-type Complex and implement addition, subtraction and multiplication operations on complex numbers.

Program:

```
#include <stdio.h>
#include <stdlib.h>
struct complex
{
    int real, img;
};
int main()
{
    int choice, x, y, z;
    struct complex a, b, c;

    printf("\nEnter a and b where a + ib is the first complex number.");
    printf("\na = ");
    scanf("%d", &a.real);
    printf("b = ");
    scanf("%d", &a.img);
    printf("\nEnter c and d where c + id is the second complex number.");
    printf("\nc = ");
    scanf("%d", &b.real);
    printf("d = ");
    scanf("%d", &b.img);

    while(1)
    {
        printf("\nPress 1 to add two complex numbers.\n");
        printf("Press 2 to subtract two complex numbers.\n");
        printf("Press 3 to multiply two complex numbers.\n");
        printf("Press 4 to exit.\n");
        printf("Enter your choice\n");
        scanf("%d", &choice);

        if (choice == 1)
        {
            c.real = a.real + b.real;
            c.img = a.img + b.img;
```

```
        if (c.img >= 0)
            printf("Sum of the complex numbers = %d + %di",
                    c.real, c.img);
        else
            printf("Sum of the complex numbers = %d %di",
                    c.real, c.img);-
    }
    else if (choice == 2)
    {
        c.real = a.real - b.real;
        c.img = a.img - b.img;
        if (c.img >= 0)
            printf("Difference of the complex numbers = %d +
                    %di", c.real, c.img);
        else
            printf("Difference of the complex numbers = %d
                    %di", c.real, c.img);
    }
    else if (choice == 3)
    {
        c.real = a.real*b.real - a.img*b.img;
        c.img = a.img*b.real + a.real*b.img;
        if (c.img >= 0)
            printf("Multiplication of the complex numbers = %d +
                    %di", c.real, c.img);
        else
            printf("Multiplication of the complex numbers = %d
                    %di", c.real, c.img);
    }
    else if (choice == 4)
        exit(0);
    else
        printf("Invalid choice.");
    printf("\nPress any key to enter choice again...\n");
}
}
```

Output:

Enter a and b where $a + ib$ is the first complex number.

$a = 2$

$b = 2$

Enter c and d where $c + id$ is the second complex number.

$c = 3$

$d = 3$

Press 1 to add two complex numbers.

Press 2 to subtract two complex numbers.

Press 3 to multiply two complex numbers.

Press 4 to exit.

Enter your choice

1

Sum of the complex numbers = $5 + 5i$

Press any key to enter choice again...

Press 1 to add two complex numbers.

Press 2 to subtract two complex numbers.

Press 3 to multiply two complex numbers.

Press 4 to exit.

Enter your choice

2

Difference of the complex numbers = $-1 - 1i$

Press any key to enter choice again...

Press 1 to add two complex numbers.

Press 2 to subtract two complex numbers.

Press 3 to multiply two complex numbers.

Press 4 to exit.

Enter your choice

3

Multiplication of the complex numbers = $0 + 12i$

Press any key to enter choice again...

Press 1 to add two complex numbers.

Press 2 to subtract two complex numbers.

Press 3 to multiply two complex numbers.

Press 4 to exit.

Enter your choice

4

27. Write a C program to create a user defined data-type Student containing the fields Roll No, name and date of birth (by creating a user defined type Date). Implement C functions to read the details of a student and create an array of students.

Program:

```
#include <stdio.h>

struct date
{
    int dd, mm, yyyy;
};

struct student
{
    int roll;
    char name[50];
    struct date dob;
};

int main()
{
    int i, n;
    printf("\nEnter the number of students:");
    scanf("%d", &n);

    struct student st[n];

    // storing information
    for (i = 0; i < n; i++)
    {
        printf("\n\nEnter the Information of student[%d]:", i+1);
        printf("\nEnter Roll No:");
        scanf("%d", &st[i].roll);
        printf("Enter Name: ");
        scanf("%s", st[i].name);
        printf("Enter Date of Birth (dd, mm, yyyy): ");
        scanf("%d%d%d", &st[i].dob.dd, &st[i].dob.mm, &st[i].dob.yyyy);
    }
}
```

```
printf("\n\nDisplaying Information:\n");

// displaying information
for (i = 0; i < n; i++)
{
    printf("\n\nInformation of Student[%d]:", i + 1);
    printf("\nName:%s", st[i].name);
    printf("\nRoll No:%d", st[i].roll);
    printf("\nDate of Birth: %d/%d/%d", st[i].dob.dd, st[i].dob.mm, st[i].dob.yyyy);
}
return 0;
}
```

Output:

Enter the number of students: 2
Enter the Information of student[1]:
Enter Roll No: 10
Enter Name: abishek
Enter Date of Birth (dd, mm, yyyy): 10 02 1991

Enter the Information of student[2]:
Enter Roll No: 20
Enter Name: balaji
Enter Date of Birth (dd, mm, yyyy): 12 05 2000

Displaying Information:

Information of Student[1]:
Name: abishek
Roll No: 10
Date of Birth: 10/2/1991

Information of Student[2]:
Name: balaji
Roll No: 20
Date of Birth: 12/5/2000

28. Write a C program to illustrate the user-defined datatype union.**Program:**

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

union student
{
    char name[20];
    char subject[20];
    float percentage;
};

int main()
{
    union student record1;
    union student record2;

    // assigning values to record1 union variable
    strcpy(record1.name, "Raju");
    strcpy(record1.subject, "Maths");
    record1.percentage = 86.50;

    printf("Union record1 values example\n");
    printf(" Name      : %s \n", record1.name);
    printf(" Subject   : %s \n", record1.subject);
    printf(" Percentage : %f \n\n", record1.percentage);

    // assigning values to record2 union variable
    printf("Union record2 values example\n");
    strcpy(record2.name, "Mani");
    printf(" Name      : %s \n", record2.name);

    strcpy(record2.subject, "Physics");
    printf(" Subject   : %s \n", record2.subject);

    record2.percentage = 99.50;
    printf(" Percentage : %f \n", record2.percentage);
    return 0;
}
```

Output:

Union record1 values example

Name :

Subject :

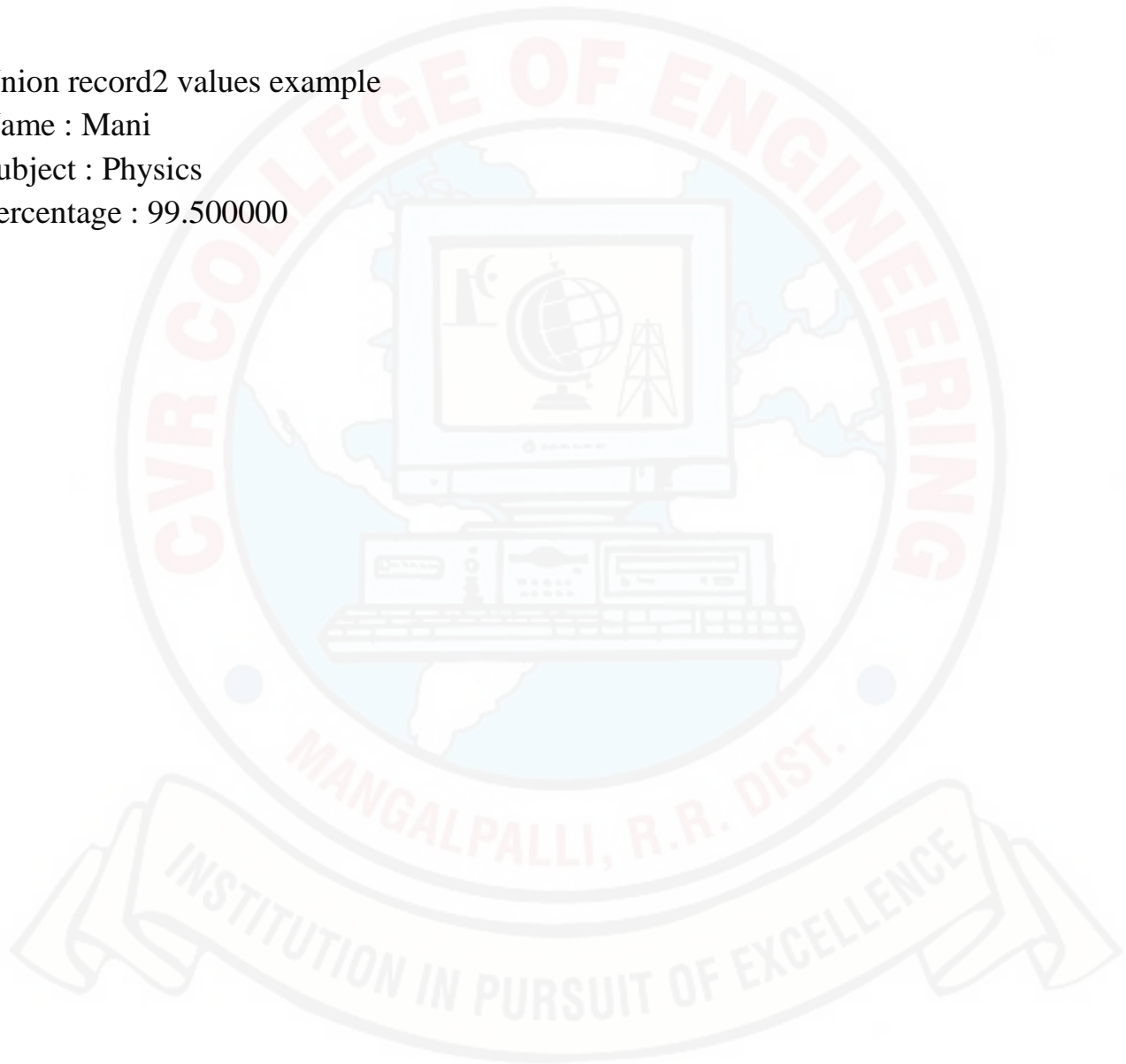
Percentage : 86.500000;

Union record2 values example

Name : Mani

Subject : Physics

Percentage : 99.500000



29. Write a C program to read the content of a given text file and count the number of characters, words and lines in it (Read the file name as command line argument).

Program:

```
// testfile.c
#include<stdio.h>
#include<stdlib.h>

int main(int argc, char *argv[])
{
    FILE *fp;
    char ch;
    int c=0, w=0, l=0;
    if(argc!=2)
    {
        printf("Enter two arguments only");
        exit(0);
    }
    fp=fopen(argv[1], "r");
    if(fp==NULL)
    {
        printf("Unable to open file");
        exit(0);
    }

    printf("\nThe contents of the input file is:");
    while((ch=fgetc(fp))!=EOF)
    {
        printf("%c", ch);
        c++;
        if(ch==' ' || ch=='\n')
            w++;
        if(ch=='\n')
            l++;
    }

    printf("\n No. of characters = %d \n No. of words = %d \n No. of lines = %d",
        c, w, l);
}
```

```
    fclose(fp);  
    return 0;  
}
```

Output:

```
$ gedit inputfile  
This is Manish  
I had worked in Wipro and Cisco
```

```
$ gcc testfile.c  
$ a.out inputfile
```

```
The contents of the input file is:  
This is Manish  
I had worked in Wipro and Cisco
```

```
No. of characters = 38  
No. of words = 10  
No. of lines = 2
```


30. Write a C program to read the content of a given text file, convert all lower case letters into upper case and display it on the screen.

Program:

```
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
int main()
{
    FILE *fp1, *fp2;
    char ch;

    fp1 = fopen("source.txt", "r");
    if (fp1 == NULL)
    {
        puts("File does not exist..");
        exit(1);
    }

    fp2 = fopen("target.txt", "w");
    if (fp2 == NULL)
    {
        puts("File does not exist..");
        fclose(fp1);
        exit(1);
    }

    while((ch=fgetc(fp1))!=EOF)
    {
        ch = toupper(ch);
        fputc(ch,fp2);
    }
    rewind(fp2);
    printf("\nContents of the Target File is:\n");
    while((ch=getc(fp2))!=EOF)
    {
        printf("%c",ch);
    }
    printf("\n");
}
```

```
fclose(fp1);  
fclose(fp2);  
return 0;  
  
}
```

Output:**Source.txt File**

Here is source code of the C Program to convert the content of file to Upper Case. The C program is successfully compiled and run on a Linux system.

Contents of the Target File is:

HERE IS SOURCE CODE OF THE C PROGRAM TO CONVERT THE CONTENT OF FILE TO UPPER CASE. THE C PROGRAM IS SUCCESSFULLY COMPILED AND RUN ON A LINUX SYSTEM.

31. Write a C program to copy the contents of one file into another.**Program:**

```
#include <stdio.h>
#include <stdlib.h> // For exit()

int main()
{
    FILE *fptr1, *fptr2;
    char file1[50], file2[50], c;

    printf("Enter the filename to open for reading \n");
    scanf("%s", file1);

    // Open one file for reading
    fptr1 = fopen(file1, "r");
    if (fptr1 == NULL)
    {
        printf("Cannot open file %s \n", file1);
        exit(0);
    }

    printf("Enter the filename to open for writing \n");
    scanf("%s", file2);

    // Open another file for writing
    fptr2 = fopen(file2, "w");
    if (fptr2 == NULL)
    {
        printf("Cannot open file %s \n", file2);
        exit(0);
    }

    // Read contents from file
    c = fgetc(fptr1);
    while (c != EOF)
    {
        fputc(c, fptr2);
        c = fgetc(fptr1);
    }
```

```
}

printf("\nContents copied to %s", file2);

fclose(fp1);
fclose(fp2);
return 0;
}
```

Output:

Enter the filename to open for reading

a.txt

Enter the filename to open for writing

b.txt

Contents copied to b.txt

32. Write a C program to write the record list of Student type into a binary file student.dat. Re-open the file, read the records from the file and display on the screen.

Program:

```
#include <stdio.h>
struct student
{
    char name[50];
    int rollno;
    float avg;
};

int main()
{
    struct student a[10], b[10];
    FILE *fptr;
    int i, n;
    fptr=fopen("student.dat","wb");
    printf("\nEnter the Number of Students:");
    scanf("%d", &n);

    for (i=0; i<n; i++)
    {
        fflush(stdin);
        printf("\nEnter Student %d Information", i+1);
        printf("\nEnter name: ");
        scanf("%s", a[i].name);
        printf("\nEnter Roll No: ");
        scanf("%d",&a[i].rollno);
        printf("\nEnter Average Mark: ");
        scanf("%f",&a[i].avg);
    }
    fwrite(a,sizeof(a),1,fptr);
    fclose(fptr);

    fptr=fopen("student.dat","rb");
    fread(b,sizeof(b),1,fptr);
    printf("\nThe Student Records present in the File are:");
```

```
for (i=0; i<n; i++)  
{  
    printf("\nName: %s\tRoll No: %d\tAverage Mark: %f\n", b[i].name,  
        b[i].rollno, b[i].avg);  
}  
fclose(fp);  
}
```

Output:

Enter the Number of Students: 3

Enter Student 1 Information

Enter name: Abishek

Enter Roll No: 11

Enter Average Mark: 84.5

Enter Student 2 Information

Enter name: Balaji

Enter Roll No: 12

Enter Average Mark: 78.6

Enter Student 3 Information

Enter name: Deepak

Enter Roll No: 13

Enter Average Mark: 92.4

The Student Records present in the File are:

Name: Abishek	Roll No: 11	Average Mark: 84.5
Name: Balaji	Roll No: 12	Average Mark: 78.6
Name: Deepak	Roll No: 13	Average Mark: 92.4