Maharaja Surajmal Institute

Assignment of “C Programming”

Name: Atul Sharma

Course: BCA

Batch: 2023-2026

Submitted to: Neetu Narwal

Program 1: WAP to calculate the real root of the equation $ax^2+bx+c=0$ using the quadratic formula

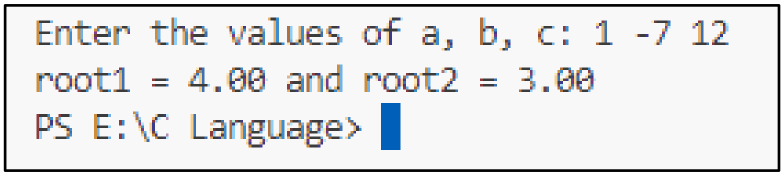
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Test with a=1, b=-7 , c=12, roots are 3 and 4

Program

```
#include <math.h>
#include <stdio.h>
int main() {
    double a, b, c, discriminant, root1, root2, realPart,
imagPart;
    printf("Enter the values of a, b, c: ");
    scanf("%lf %lf %lf", &a, &b, &c);

    discriminant = b * b - 4 * a * c;

    // condition for real and different roots
    if (discriminant > 0) {
        root1 = (-b + sqrt(discriminant)) / (2 * a);
        root2 = (-b - sqrt(discriminant)) / (2 * a);
        printf("root1 = %.2lf and root2 = %.2lf", root1,
root2);
    }

    // condition for real and equal roots
    else if (discriminant == 0) {
        root1 = root2 = -b / (2 * a);
        printf("root1 = root2 = %.2lf;", root1);
    }

    // if roots are not real
    else {
        realPart = -b / (2 * a);
        imagPart = sqrt(-discriminant) / (2 * a);
        printf("root1 = %.2lf+%.2lfi and root2 = %.2f-%.2fi",
realPart, imagPart, realPart, imagPart);
    }

    return 0;
}
```

Output



```
Enter the values of a, b, c: 1 -7 12
root1 = 4.00 and root2 = 3.00
PS E:\C Language> 
```

Program 2: WAP to calculate Compound interest using formula
Consider n as 1 in this example

$$A = P(1 + \frac{r}{n})^{nt}$$

A = final amount

P = initial principal balance

r = interest rate

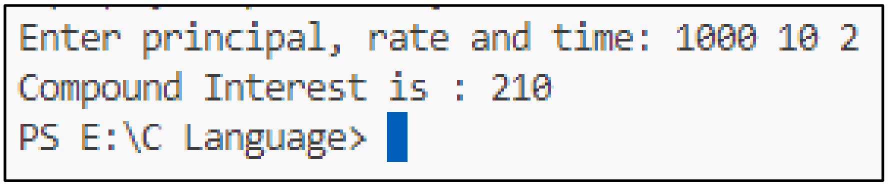
n = number of times interest applied per time period

t = number of time periods elapsed

Program

```
// C program to calculate Compound Interest
#include <stdio.h>
#include<math.h>

int main()
{
    // Declaring variables
    float principal,rate, time;

    // taking input from user
    printf("Enter principal, rate and time: ");
    scanf("%f %f %f", &principal, &rate, &time);

    // Calculating compound Interest
    float amount = principal *
                    ((pow((1 + rate / 100),
                        time)));
    int compint = amount - principal;

    printf("Compound Interest is : %d", compint);
    return 0;
}
```

Output



```
Enter principal, rate and time: 1000 10 2
Compound Interest is : 210
PS E:\C Language>
```

Program 3: : WAP to calculate degrees Celsius given temperature in degrees Fahrenheit using the formula

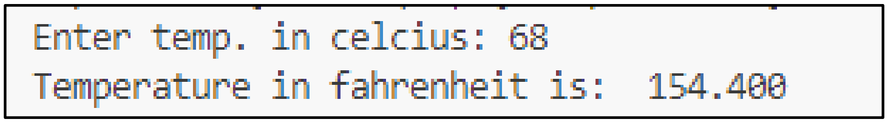
C= (5/9) * (F-32)

Test with following values 68,150,0,-22(degrees Fahrenheit 154.4, 302,32,-7.6).

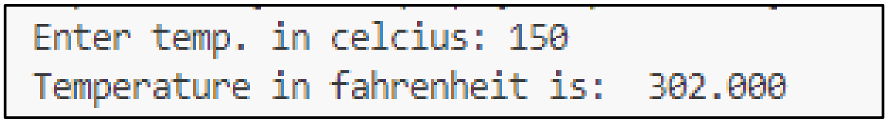
Program

```
#include<stdio.h>
int main()
{
    float fahrenheit, celsius;

    // taking input from user
    printf("Enter temp. in celcius: ");
    scanf("%f", &celsius);

    fahrenheit =( (celsius*9)/5)+32;
    printf("\n\n Temperature in fahrenheit
is:  %.5f",fahrenheit);
    return (0);
}
```

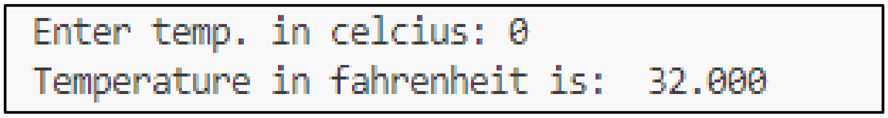
Output



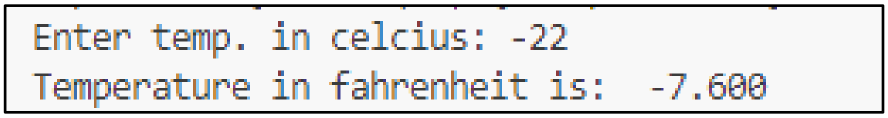
```
Enter temp. in celcius: 68
Temperature in fahrenheit is:  154.400
```



```
Enter temp. in celcius: 150
Temperature in fahrenheit is:  302.000
```



```
Enter temp. in celcius: 0
Temperature in fahrenheit is:  32.000
```



```
Enter temp. in celcius: -22
Temperature in fahrenheit is:  -7.600
```

Program 4: WAP to Calculate the volume and area of a sphere using the formulas accept radius from user

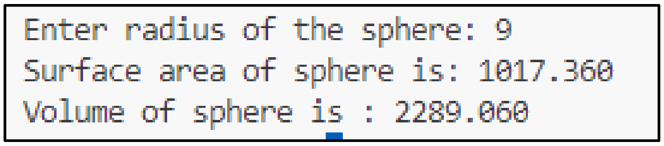
$$A = 4\pi r^2 \qquad V = \frac{4}{3}\pi r^3$$

Test with radius=9cm, Volume = 3053.628 cm3 , Area= 1017.87 cm

Program

```
// C program to find area and volume of sphere
#include <stdio.h>
#include <math.h>
 #define pie 3.14
int main()
{
    // declaring variables
    float radius;
    float surface_area, volume;

    printf("Enter radius of the sphere: ");
    scanf("%f", &radius);

    surface_area =  4 * pie * radius * radius;
    volume = (4/3) * pie * (radius * radius * radius);
    printf("Surface area of sphere is: %.3f", surface_area);
    printf("\nVolume of sphere is : %.3f", volume);
    return 0;
}
```

Output



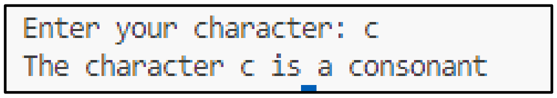
```
Enter radius of the sphere: 9
Surface area of sphere is: 1017.360
Volume of sphere is : 2289.060
```

Program 5: WAP to accept a character from user and check whether an alphabet is vowel or consonant using conditional operator

Program

```
#include <stdio.h>
int main()
{
    // declaring variables
    char ch;

    // taking input from user
    printf("Enter your character: ");
    scanf("%c", &ch);

    if (ch == 'a' || ch == 'A' || ch == 'e' || ch == 'E'
        || ch == 'i' || ch == 'I' || ch == 'o' || ch == 'O'
        || ch == 'u' || ch == 'U')
        {
            printf("The character %c is a vowel", ch);}

    else {
        printf("The character %c is a consonant", ch);}

    return 0;
}
```

Output



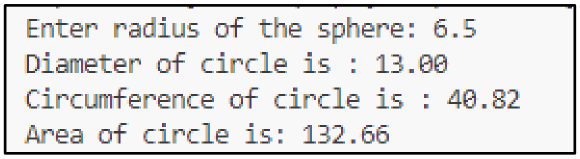
```
Enter your character: c
The character c is a consonant
```

Program 6: Write a Program to calculate diameter, circumference and area of circle accept radius from user

Program

```
// C program to find area and volume of sphere
#include <stdio.h>
#include <math.h>
 #define pie 3.14
int main()
{
    // declaring variables
    float radius, diameter, circumference, area;

    printf("Enter radius of the sphere: ");
    scanf("%f", &radius);

    diameter =  2*radius;
    circumference = 2*pie*radius;
    area = pie*radius*radius;

    printf("Diameter of circle is : %.2f", diameter);
    printf("\nCircumference of circle is : %.2f",
                                  circumference);
    printf("\nArea of circle is: %.2f", area);

    return 0;
}
```

Output



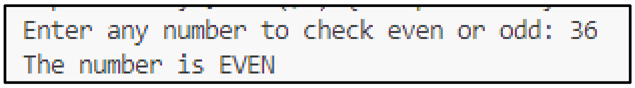
```
Enter radius of the sphere: 6.5
Diameter of circle is : 13.00
Circumference of circle is : 40.82
Area of circle is: 132.66
```

Program 7: WAP to accept number from user and display whether it is even or odd. Hint : use conditional and modulus operator

Program

```
// C program to check even or odd number using conditional
operator

#include <stdio.h>

int main()
{
    int num;

    /* Input a number from user */
    printf("Enter any number to check even or odd: ");
    scanf("%d", &num);

    (num%2 == 0)
        ? printf("The number is EVEN")
        : printf("The number is ODD");

    return 0;
}
```

Output



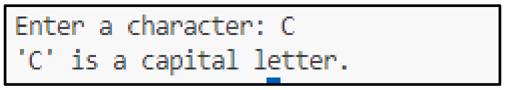
```
Enter any number to check even or odd: 36
The number is EVEN
```

Program 8: WAP to accept a character and display whether it is Capital Letter, Small Letter or Integer value

Program

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

bool is_capital_letter(char c) {
    return 'A' <= c && c <= 'Z';
}

bool is_small_letter(char c) {
    return 'a' <= c && c <= 'z';
}

bool is_integer(char c) {
    return '0' <= c && c <= '9';
}

int main() {
    char c;
    printf("Enter a character: ");
    scanf(" %c", &c);

    if (is_capital_letter(c)) {
        printf("'%c' is a capital letter.\n", c);
    } else if (is_small_letter(c)) {
        printf("'%c' is a small letter.\n", c);
    } else if (is_integer(c)) {
        printf("'%c' is an integer.\n", c);
    } else {
        printf("'%c' is an unknown character.\n", c);
    }

    return 0;
}
```

Output



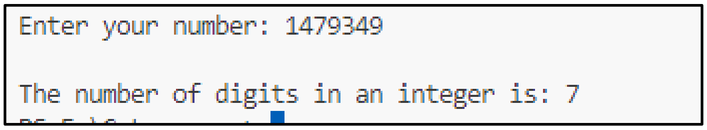
```
Enter a character: C
'C' is a capital letter.
```

Program 9: Write a program to count number of digits in any number

Program

```
#include <stdio.h>
int main()
{
    // variable declaration
    int n, count=0;

    printf("Enter your number: ");
    scanf("%d",&n);

    // using while loop
    while(n!=0)
    {
       n=n/10;
       count++;
    }

    printf("\nThe number of digits in an integer is:
%d",count);
    return 0;
}
```

Output



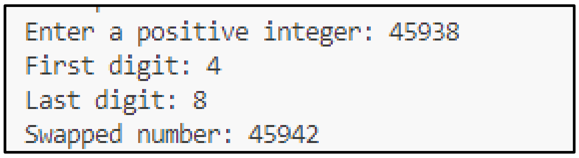
```
Enter your number: 1479349

The number of digits in an integer is: 7
```

Program 10: Write a program to find first and last digit of any number display them, then swap the digits and create a new number and display it

Program

```
#include <stdio.h>
int main()
{
    int number, firstDigit, lastDigit, temp, numOfDigits;

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

    // Step 2: Calculate the number of digits in the given
number
    numOfDigits = snprintf(NULL, 0, "%d", number);

    // Step 3: Extract the first and last digits
    lastDigit = number % 10;
    firstDigit = number / pow(10, numOfDigits - 1);

    printf("First digit: %d\n", firstDigit);
    printf("Last digit: %d\n", lastDigit);

    // Step 6: Swap the first and last digits
    temp = firstDigit;
    firstDigit = lastDigit;
    lastDigit = temp;

    // Step 7: Construct the final swapped number
    int swappedNumber = lastDigit * pow(10, numOfDigits - 1)
+ number % ((int)pow(10, numOfDigits - 1));

    printf("Swapped number: %d\n", swappedNumber);
    return 0;
}
```

Output



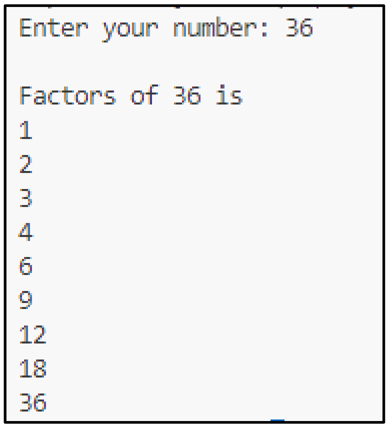
```
Enter a positive integer: 45938
First digit: 4
Last digit: 8
Swapped number: 45942
```

Program 11: Write a program to enter any number and print all factors of the number

Program

```
#include <stdio.h>
int main ()
{
    // declaring variables

    int num, i;

    printf("Enter your number: ");
    scanf("%d", &num);

    printf("\nFactors of %d is", num);
    for (i=1; i<=num;i++)
    {
        if(num%i==0)
        {
            printf("\n%d", i);
        }
    }
    return 0;
}
```

Output



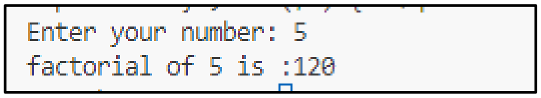
```
Enter your number: 36

Factors of 36 is
1
2
3
4
6
9
12
18
36
```

Program 12: Write a Program to enter any number and calculate its factorial

Program

```
#include <stdio.h>
int main ()
{
    int num;
    int factorial=1;
    printf("Enter your number: ");
    scanf("%d", &num);
    if(num<0)
    {
        printf("factorial is not defined for negative numbers
\n");
    }
    else
    {
        for(int i=1; i<=num;++i)
        {
            factorial=factorial*i;
        }
    printf("factorial of %d is :%d\n", num, factorial);
    }
    return 0;
}
```

Output



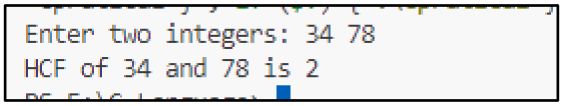
```
Enter your number: 5
factorial of 5 is :120
```

Program 13: Write a program to find HCF (GCD) of two numbers

Program

```
#include <stdio.h>
int main ()
{
    int n1,n2,i,hcf;

    printf("Enter two integers:");
    scanf("%d %d", &n1, &n2);

    // using for loop
    for(i=1;i<=n1 && i<=n2; i++)
    {
        if(n1%i==0 && n2%i==0)
        hcf=i;
    }
    printf("HCF of %d and %d is %d", n1, n2, hcf);
    return 0;
}
```

Output



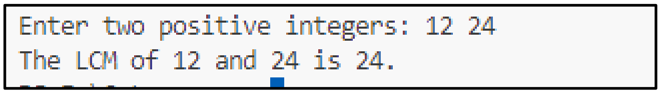
```
Enter two integers: 34 78
HCF of 34 and 78 is 2
```

Program 14: Write a Program to find LCM of two numbers

Program

```
#include <stdio.h>

int main() {

    int n1, n2, max;

    printf("Enter two positive integers: ");
    scanf("%d %d", &n1, &n2);

    // maximum number between n1 and n2 is stored in max
    max = (n1 > n2) ? n1 : n2;

    while (1) {
        if ((max % n1 == 0) && (max % n2 == 0)) {
            printf("The LCM of %d and %d is %d.", n1, n2,
max);
            break;
        }
        ++max;
    }
    return 0;
}
```

Output



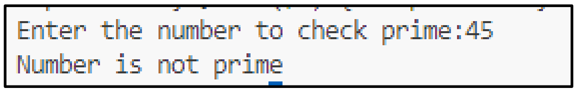
```
Enter two positive integers: 12 24
The LCM of 12 and 24 is 24.
```

Program 15: Write a program to check whether a number is Prime number or not

Program

```
#include<stdio.h>
int main()
{
    int n,i,m=0,flag=0;
    printf("Enter the number to check prime:");
    scanf("%d",&n);
    m=n/2;
    for(i=2;i<=m;i++)
    {
        if(n%i==0)
        {
            printf("Number is not prime");
            flag=1;
            break;
        }
    }
    if(flag==0)
    {
        printf("Number is prime");
    }
    return 0;
}
```

Output



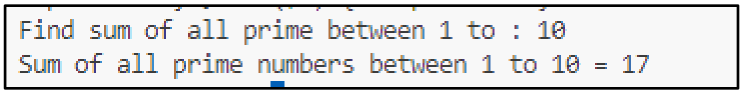
```
Enter the number to check prime:45
Number is not prime
```

Program 16: Write a program to find sum of all prime numbers between 1 to n

Program

```
#include <stdio.h>
int main()
{
    int i, j, end, isPrime, sum=0;

    /* Input upper limit from user */
    printf("Find sum of all prime between 1 to : ");
    scanf("%d", &end);

    /* Find all prime numbers between 1 to end */
    for(i=2; i<=end; i++)
    {
        // Check if the current number i is Prime or not
        isPrime = 1;
        for(j=2; j<=i/2 ;j++)
        {
            if(i%j==0)
            {
                /* 'i' is not prime */
                isPrime = 0;
                break;
            }
        }

        // if 'i' is Prime then add to sum
        if(isPrime==1)
        {
            sum += i;
        }
    }
    printf("Sum of all prime numbers between 1 to %d = %d",
end, sum);

    return 0;
}
```

Output



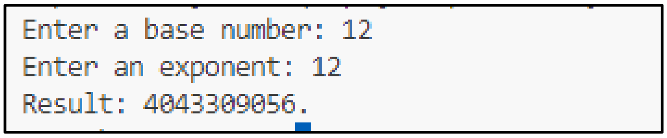
```
Find sum of all prime between 1 to : 10
Sum of all prime numbers between 1 to 10 = 17
```

Program 17: Write a function “power (a,b)”, to calculate the value of a raised to the power of b

Program

```
#include <stdio.h>

unsigned long power(int a, int b) {
    if (b == 0) {
        return 1;
    } else {
        return a * power(a, b - 1);
    }
}

int main() {
    int base, exponent;
    printf("Enter a base number: ");
    scanf("%d", &base);
    printf("Enter an exponent: ");
    scanf("%d", &exponent);

    if (exponent < 0) {
        printf("Negative exponent is not allowed.\n");
    } else {
        printf("Result: %lu.\n", power(base, exponent));
    }

    return 0;
}
```

Output



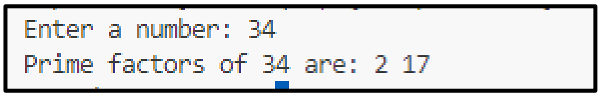
```
Enter a base number: 12
Enter an exponent: 12
Result: 4043309056.
```

Program 18: Write a program to enter any number and print all factors of the number

Program

```
#include <stdio.h>

void primefactor(int a) {
    int i = 2;

    printf("Prime factors of %d are: ", a);

    while (i <= a) {
        if (a % i == 0) {
            printf("%d ", i);
            a = a / i;
        } else {
            i++;
        }
    }
    printf("\n");
}

int main() {
    int number;
    printf("Enter a number: ");
    scanf("%d", &number);

    if (number < 0) {
        printf("Prime factors of negative numbers are not
allowed.\n");
    } else {
        primefactor(number);
    }

    return 0;
}
```

Output



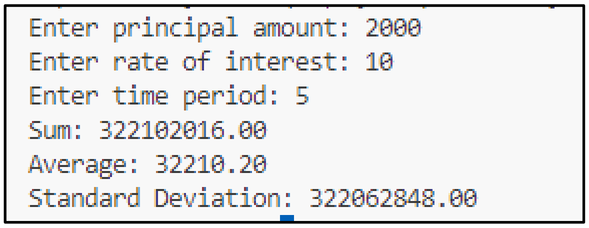
```
Enter a number: 34
Prime factors of 34 are: 2 17
```

Program 19: Write a C program with function name CompoundInterest(P,r,t) it receives 3 arguments principal, rate, time and displays the sum, average and the standard deviation of these numbers

Program

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

// Function to calculate compound interest
void CompoundInterest(float P, float r, float t)
{
    float sum, average, standard_deviation;
    int n = 10000; // number of random values to calculate
standard deviation

    // calculate sum
    sum = P * pow(1 + r, t);

    // calculate average
    average = sum / n;

    // calculate standard deviation
    float deviations[n];
    for (int i = 0; i < n; i++) {
        deviations[i] = P * pow(1 + r, t) - average;
    }
    float variance = 0;
    for (int i = 0; i < n; i++) {
        variance += pow(deviations[i], 2);
    }
    variance /= n;
    standard_deviation = sqrt(variance);

    printf("Sum: %.2f\n", sum);
    printf("Average: %.2f\n", average);
    printf("Standard Deviation: %.2f\n", standard_deviation);
}

int main() {
    float P, r, t;

    printf("Enter principal amount: ");
    scanf("%f", &P);

    printf("Enter rate of interest: ");
    scanf("%f", &r);
    printf("Enter time period: ");
    scanf("%f", &t);

    CompoundInterest(P, r, t);

    return 0;
}
```

Output



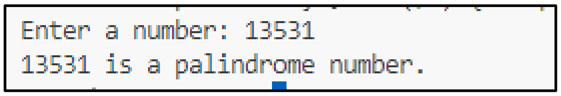
```
Enter principal amount: 2000
Enter rate of interest: 10
Enter time period: 5
Sum: 322102016.00
Average: 32210.20
Standard Deviation: 322062848.00
```

Program 20: Write a function Palindrome(a), that accepts an integer and return 1 if the number is a palindrome number and return 0 if it is not

Program

```
#include <stdio.h>

// Function to check if a number is a palindrome
int Palindrome(int a) {
    int reversed = 0;
    int original = a;

    // Reverse the digits of the number
    while (a != 0) {
        reversed = reversed * 10 + a % 10;
        a = a / 10;
    }

    // If the reversed number is equal to the original number, it is a
palindrome
    if (original == reversed) {
        return 1;
    }
    else {
        return 0;
    }
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);

    if (Palindrome(num)) {
        printf("%d is a palindrome number.\n", num);
    }
    else {
        printf("%d is not a palindrome number.\n", num);
    }

    return 0;
}
```

Output



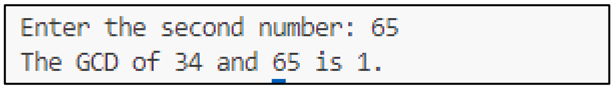
```
Enter a number: 13531
13531 is a palindrome number.
```

Program 21: Write a recursive function GCDRec(a,b), that accepts two integer values and returns the GCD of two numbers.

Program

```
#include <stdio.h>

// Function to calculate the GCD of two numbers using
recursion
int GCDRec(int a, int b) {
    // If the second number is 0, the GCD is the first number
    if (b == 0) {
        return a;
    }

    // Otherwise, the GCD is the GCD of the second number and
the remainder of the first number divided by the second
number
    else {
        return GCDRec(b, a % b);
    }
}

int main() {
    int num1, num2;
    printf("Enter the first number: ");
    scanf("%d", &num1);
    printf("Enter the second number: ");
    scanf("%d", &num2);

    printf("The GCD of %d and %d is %d.\n", num1, num2,
GCDRec(num1, num2));

    return 0;
}
```

Output



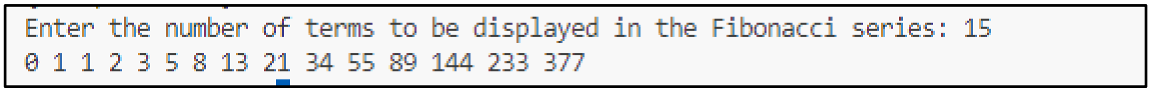
```
Enter the second number: 65
The GCD of 34 and 65 is 1.
```

Program 22: : Write a recursive function FiboRec(a), that accepts number of terms to be displayed in fibonacci series starting with 1 ,1

Program

```
#include <stdio.h>

// Function to calculate the Fibonacci series using recursion
int FiboRec(int a) {
    // Base case: If the input number is 0, the function
returns 0
    if (a == 0) {
        return 0;
    }

    // Base case: If the input number is 1, the function
returns 1
    else if (a == 1) {
        return 1;
    }

    // Recursive case: The function returns the sum of the
Fibonacci series calculated for the two preceding numbers
    else {
        return FiboRec(a - 1) + FiboRec(a - 2);
    }
}

int main() {
    int num;
    printf("Enter the number of terms to be displayed in the
Fibonacci series: ");
    scanf("%d", &num);

    // Iterate over the number of terms specified by the user
    for (int i = 0; i < num; i++) {
        printf("%d ", FiboRec(i));
    }

    printf("\n");

    return 0;
}
```

Output



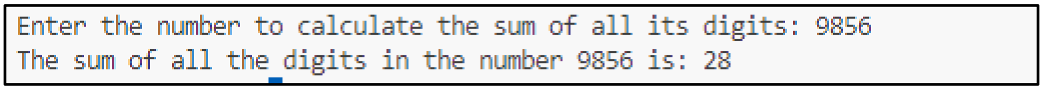
```
Enter the number of terms to be displayed in the Fibonacci series: 15
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

Program 23: Write a recursive fuction SumTermRec(a), that accepts an integer value and return the sum of all the digits.

Program

```
#include <stdio.h>

// Function to calculate the sum of all the digits in an
integer value using recursion
int SumTermRec(int a) {
    // Base case: If the input number is 0, the function
returns 0
    if (a == 0) {
        return 0;
    }

    // Recursive case: The function returns the sum of the
last digit of the number and the sum of all the other digits
    else {
        return (a % 10) + SumTermRec(a / 10);
    }
}

int main() {
    int num;
    printf("Enter the number to calculate the sum of all its
digits: ");
    scanf("%d", &num);

    printf("The sum of all the digits in the number %d is:
%d\n", num, SumTermRec(num));

    return 0;
}
```

Output



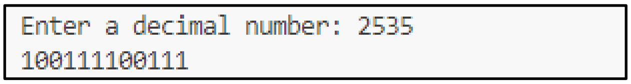
```
Enter the number to calculate the sum of all its digits: 9856
The sum of all the digits in the number 9856 is: 28
```

Program 24: Write a recursive function DectoBinaryRec(a), that accepts an integer and return Binary equivalent of the decimal number.

Program

```
#include <stdio.h>

void DectoBinaryRec(int a)
{
    if(a == 0)
        return;
    DectoBinaryRec(a / 2);
    printf("%d", a % 2);
}

int main()
{
    int num;
    printf("Enter a decimal number: ");
    scanf("%d", &num);
    DectoBinaryRec(num);
    return 0;
}
```

Output



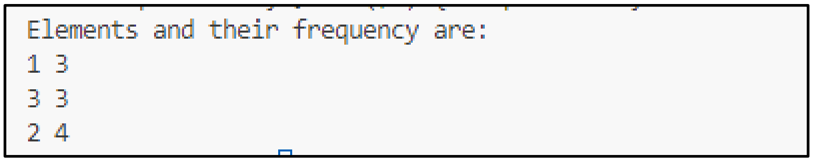
```
Enter a decimal number: 2535
100111100111
```

Program 25: Write a Program to count frequency of each element in an array

Program

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

void countFrequency(int arr[], int n)
{
   int i, j, count;
   printf("Elements and their frequency are: \n");
   for (i = 0; i < n; i++)
   {
       count = 1;
       for (j = i + 1; j < n; j++)
       {
           if (arr[i] == arr[j])
           {
               count++;
               arr[j] = -1;
           }
       }
       if (arr[i] != -1)
           printf("%d %d\n", arr[i], count);
   }
}

int main()
{
   int arr[] = {1, 3, 3, 2, 1, 2, 2, 2, 3, 1};
   int n = sizeof(arr) / sizeof(arr[0]);

   countFrequency(arr, n);

   return 0;
}
```

Output



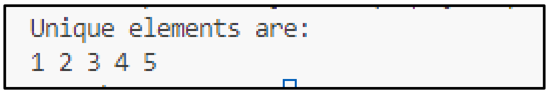
```
Elements and their frequency are:
1 3
3 3
2 4
```

Program 26: Write a Program to count frequency of each element in an array

Program

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

void printUnique(int arr[], int n) {
   if (n == 0)
       return;

   printUnique(arr, n - 1);

   int i;
   for (i = 0; i < n - 1; i++)
       if (arr[i] == arr[n - 1])
           break;

   if (i == n - 1)
       printf("%d ", arr[n - 1]);
}

int main() {
   int arr[] = {1, 1, 2, 2, 3, 3, 4, 4, 5, 5};
   int n = sizeof(arr) / sizeof(arr[0]);

   printUnique(arr, n);

   return 0;
}
```

Output



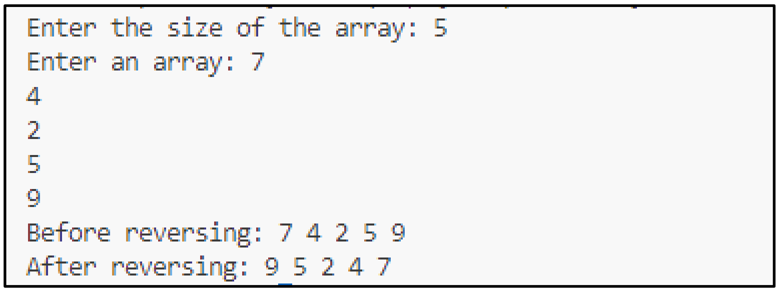
```
Unique elements are:
1 2 3 4 5
```

Program 27: Write a Program to find reverse of an array

Program

```
#include <stdio.h>
#define N 1000

int main()
{
    int arr[N];

    int n;
    // Inputting the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    // Inputting the array
    printf("Enter an array: ");
    for (int i = 0; i< n; i++){
        scanf("%d", &arr[i]);
    }

    printf("Before reversing: ");
    for (int i = 0; i< n; i++){
        printf("%d ", arr[i]);
    }

    printf("\n");

    // Printing the reverse of the array
    printf("After reversing: ");
    for (int i = n-1; i>=0; i--){
        printf("%d ", arr[i]);
    }

    return 0;
}
```

Output



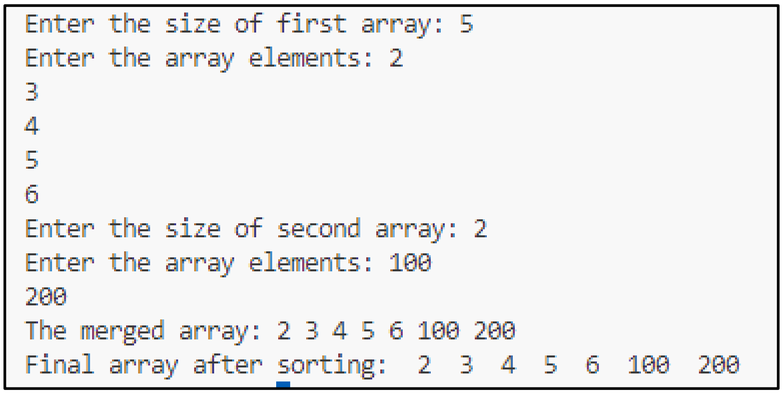
```
Enter the size of the array: 5
Enter an array: 7
4
2
5
9
Before reversing: 7 4 2 5 9
After reversing: 9 5 2 4 7
```

Program 28: Write a Program to merge two arrays to third array

Program

```
#include <stdio.h>
int main()
{
    int n1,n2,n3;            //Array Size Declaration
    int a[10000], b[10000], c[20000];
    printf("Enter the size of first array: ");
    scanf("%d",&n1);
    printf("Enter the array elements: ");
    for(int i = 0; i < n1; i++)
       scanf("%d", &a[i]);
    printf("Enter the size of second array: ");
        scanf("%d",&n2);
    printf("Enter the array elements: ");
    for(int i = 0; i < n2; i++)
       scanf("%d", &b[i]);
    n3 = n1 + n2;
    for(int i = 0; i < n1; i++)
       c[i] = a[i];
    for(int i = 0; i < n2; i++)
        c[i + n1] = b[i];

    printf("The merged array: ");
    for(int i = 0; i < n3; i++)
        printf("%d ", c[i]);        //Print the merged array

    printf("\nFinal array after sorting: ");
    for(int i = 0; i < n3; i++){
        int temp;
        for(int j = i + 1; j < n3; j++) {
            if(c[i] > c[j]) {
                temp = c[i];
                c[i] = c[j];
                c[j] = temp;
            }
        }
    }
    for(int i = 0; i < n3 ; i++)
Array
        printf(" %d ",c[i]);
    return 0;
}
```

Output



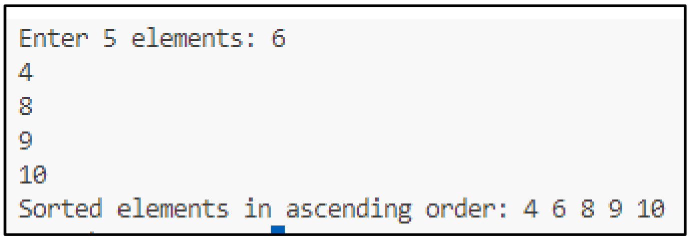
```
Enter the size of first array: 5
Enter the array elements: 2
3
4
5
6
Enter the size of second array: 2
Enter the array elements: 100
200
The merged array: 2 3 4 5 6 100 200
Final array after sorting:  2  3  4  5  6  100  200
```

Program 29: Write a program to accept 5 elements, performs sorting operation with two functions : SelectionSortAsc(), SelectionSortDesc()

Program

```
#include <stdio.h>

void swap(int *xp, int *yp) {
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void selectionSortAsc(int arr[], int n) {
    int i, j, min_idx;

    for (i = 0; i < n - 1; i++) {
        min_idx = i;

        for (j = i + 1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;

        swap(&arr[min_idx], &arr[i]);
    }
}

int main() {
    int arr[5], n = 5, i;

    printf("Enter 5 elements: ");

    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    selectionSortAsc(arr, n);

    printf("Sorted elements in ascending order: ");

    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);

    return 0;
}
```

Output

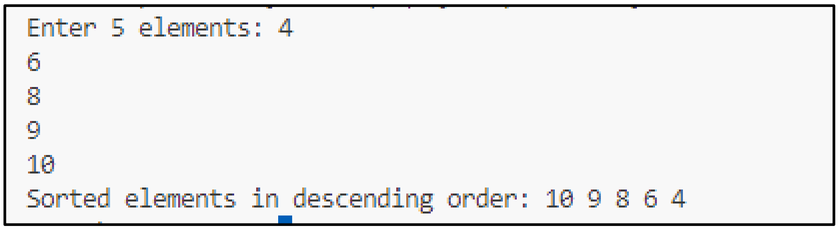


```
Enter 5 elements: 6
4
8
9
10
Sorted elements in ascending order: 4 6 8 9 10
```

Program

```
#include <stdio.h>

void swap(int *xp, int *yp) {
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void selectionSortDesc(int arr[], int n) {
    int i, j, max_idx;

    for (i = 0; i < n - 1; i++) {
        max_idx = i;

        for (j = i + 1; j < n; j++)
            if (arr[j] > arr[max_idx])
                max_idx = j;

        swap(&arr[max_idx], &arr[i]);
    }
}

int main() {
    int arr[5], n = 5, i;

    printf("Enter 5 elements: ");

    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    selectionSortDesc(arr, n);

    printf("Sorted elements in descending order: ");

    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);

    return 0;
}
```

Output



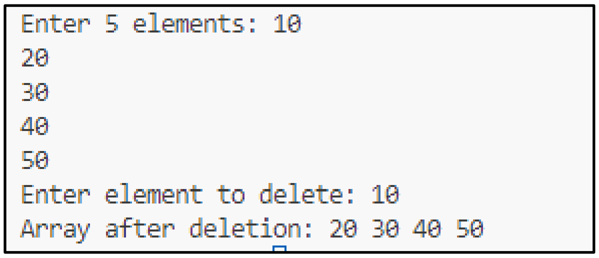
```
Enter 5 elements: 4
6
8
9
10
Sorted elements in descending order: 10 9 8 6 4
```

Program 30: Program 30: Write a Program to delete a specific element specified by the user and then shift all the elements towards left and last element will become zero.

Program

```c
#include <stdio.h>

void shiftAndDelete(int arr[], int *n, int x) {
    int i;

    for (i = 0; i < *n; i++) {
        if (arr[i] == x) {
            for (int j = i; j < *n - 1; j++)
                arr[j] = arr[j + 1];
            arr[*n - 1] = 0;
            (*n)--;
            break;
        }
    }
}

int main() {
    int arr[5], n = 5, i;

    printf("Enter 5 elements: ");

    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("Enter element to delete: ");
    int x;
    scanf("%d", &x);

    shiftAndDelete(arr, &n, x);

    printf("Array after deletion: ");

    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    arr[i] = 0;

    return 0;
}
```

Output



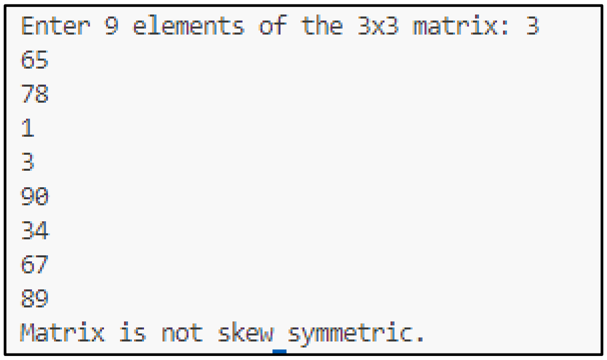
```
Enter 5 elements: 10
20
30
40
50
Enter element to delete: 10
Array after deletion: 20 30 40 50
```

Program 31: Write a Program to accept elements in 2D matrix of size 3x3. Find whether it is skew symmetric matrix or not.

Program

```c
#include <stdio.h>

void findSkewSymmetricMatrix(int arr[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (i != j && arr[i][j] != 0) {
                printf("Matrix is not skew symmetric.\n");
                return;
            }
        }
    }

    printf("Matrix is skew symmetric.\n");
}

int main() {
    int arr[3][3];

    printf("Enter 9 elements of the 3x3 matrix: ");

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            scanf("%d", &arr[i][j]);
        }
    }

    findSkewSymmetricMatrix(arr);

    return 0;
}
```

Output



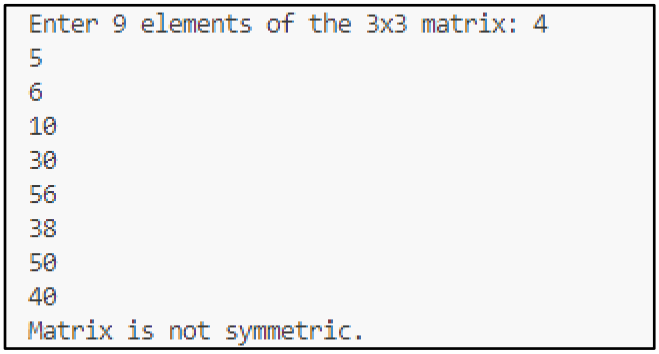
```
Enter 9 elements of the 3x3 matrix: 3
65
78
1
3
90
34
67
89
Matrix is not skew symmetric.
```

Program 32: Write a Program to accept elements in 2D matrix of size 3x3. Find whether it is symmetric matrix or not..

Program

```c
#include <stdio.h>

void findSymmetricMatrix(int arr[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (arr[i][j] != arr[j][i]) {
                printf("Matrix is not symmetric.\n");
                return;
            }
        }
    }

    printf("Matrix is symmetric.\n");
}

int main() {
    int arr[3][3];

    printf("Enter 9 elements of the 3x3 matrix: ");

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            scanf("%d", &arr[i][j]);
        }
    }

    findSymmetricMatrix(arr);

    return 0;
}
```

Output



```
Enter 9 elements of the 3x3 matrix: 4
5
6
10
30
56
38
50
40
Matrix is not symmetric.
```