Exercise 1: Write a program to convert English units to metric (i.e., miles to kilometers, gallons to liters, etc.). Include a specification and a code design.

```
#include <stdio.h>

#define MILES_TO_KM 1.60934
#define GALLONS_TO_LITERS 3.78541
#define POUNDS_TO_KG 0.453592
#define INCHES_TO_CM 2.54
#define FAHRENHEIT_TO_CELSIUS(f) ((f - 32) * 5 / 9)

void miles_to_km();
void gallons_to_liters();
void pounds_to_kg();
void inches_to_cm();
void fahrenheit_to_celsius();

int main()
{
   int choice;
   do
   {
      printf("English to Metric Converter:\n");
      printf("1. Miles to Kilometers\n");
      printf("2. Gallons to Liters\n");
      printf("3. Pounds to Kilograms\n");
      printf("4. Inches to Centimeters\n");
      printf("5. Fahrenheit to Celsius\n");
      printf("6. Exit\n");
      printf("Enter your choice: ");
      scanf(" %d", &choice);
```

```
        switch (choice)
        {
            case 1:
                miles_to_km();
                break;
            case 2:
                gallons_to_liters();
                break;
            case 3:
                pounds_to_kg();
                break;
            case 4:
                inches_to_cm();
                break;
            case 5:
                fahrenheit_to_celsius();
                break;
            case 6:
                printf("Exiting the program. Goodbye!\n");
                break;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    } while (choice != 6);

    return 0;
}



void miles_to_km()
{
```

```
    double miles;
    printf("Enter distance in miles: ");
    scanf("%lf", &miles);
    printf("%.2lf miles = %.2lf kilometers\n", miles, miles * MILES_TO_KM);
}

void gallons_to_liters()
{
    double gallons;
    printf("Enter volume in gallons: ");
    scanf("%lf", &gallons);
    printf("%.2lf gallons = %.2lf liters\n", gallons, gallons * GALLONS_TO_LITERS);
}

void pounds_to_kg()
{
    double pounds;
    printf("Enter weight in pounds: ");
    scanf("%lf", &pounds);
    printf("%.2lf pounds = %.2lf kilograms\n", pounds, pounds * POUNDS_TO_KG);
}

void inches_to_cm()
{
    double inche;
    printf("Enter length in inches: ");
    scanf("%lf", &inche);
    printf("%.2lf inches = %.2lf centimeters\n", inche, inche * INCHES_TO_CM);
}

void fahrenheit_to_celsius()
```

```
{
   double fahrenheit;
   printf("Enter temperature in fahrenheit: ");
   scanf("%lf", &fahrenheit);
   printf("%.2lf fahrenheit = %.2lf celsius\n", fahrenheit, FAHRENHEIT_TO_CELSIUS(fahrenheit));
}
```

Exercise 2: Write a program to perform date arithmetic such as how many days there are between 6/6/90 and 4/3/92. Include a specification and a code design.

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

int validate_date(int, int, int);
int days_in_month(int, int);
int isleap(int );
int days_since_start(int day, int month, int year);
int days_between_dates(int day1, int month1, int year1, int day2, int month2, int year2);

int main()
{
   int day1, month1, year1, day2, month2, year2;

   printf("Enter the first date (dd/mm/yyyy): ");
   scanf("%d/%d/%d", &day1, &month1, &year1);

   printf("Enter the second date (dd/mm/yyyy): ");
   scanf("%d/%d/%d", &day2, &month2, &year2);

   if(!validate_date(day1, month1, year1))
   {
      printf("The first date is invalid!!\n");
      return 1;
   }
```

```
    if(!validate_date(day2, month2, year2))
    {
        printf("The first date is invalid!!\n");
        return 1;
    }

    int difference = days_between_dates(day1, month1, year1, day2, month2, year2);
    printf("The number of days between %02d/%02d/%04d and %02d/%02d/%04d is %d days.\n",
           day1, month1, year1, day2, month2, year2, difference);

    return 0;
}

int validate_date(int day, int month, int year)
{
    if(day<1|| month<1 || month>12 || year<0)
    {
        return 0;
    }

    if(day <= days_in_month(month, year))
    {
        return 1;
    }
}

int days_in_month(int month, int year)
{
    if(month ==1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month
== 12)
```

```
    {
        return 31;
    }
    else if(month == 4 || month == 6 || month == 9 || month==11)
    {
        return 30;
    }
    else
    { //feb
        if(isleap(year))
        {
            return 29;
        }
        else
        {
            return 28;
        }
    }
}
int isleap(int year)
{
    if(year%4 == 0)
    {
        if(year%100 != 0)
        {
            return 1;
        }
    }
    else if(year%400 == 0)
    {
        return 1;
```

```
    }
    else
    {
        return 0;
    }
}

int days_since_start(int day, int month, int year)
{
    int days = 0;

    for (int i = 0; i < year; i++)
    {
        days += isleap(i) ? 366 : 365;
    }


    for (int i = 1; i < month; i++)
    {
        days += days_in_month(i, year);
    }

    days += day;

    return days;
}

int days_between_dates(int day1, int month1, int year1, int day2, int month2, int year2) {
    int days1 = days_since_start(day1, month1, year1);
    int days2 = days_since_start(day2, month2, year2);
    return abs(days2 - days1);
```

```
}
```

Exercise 3: A serial transmission line can transmit 960 characters each second. Write a program that will calculate the time required to send a file, given the file's size. Try the prog ram on a 400MB (419,430,400 -byte) file. Use appropriate units. (A 400MB file takes days.)

```
#include <stdio.h>

void calculate_time(long long file_size)
{
    const int TRANSMISSION_RATE = 960; // characters per second
    long long total_seconds = file_size / TRANSMISSION_RATE;

    int days = total_seconds / 86400;
    total_seconds %= 86400;

    int hours = total_seconds / 3600;
    total_seconds %= 3600;

    int minutes = total_seconds / 60;
    int seconds = total_seconds % 60;

    printf("Transmission time:\n");
    printf("%d days, %d hours, %d minutes, %d seconds\n", days, hours, minutes, seconds);
}

int main()
{
    long long file_size;

    // Example for 400MB file size
    printf("Enter the file size in bytes: ");
```

```
    scanf("%lld", &file_size);

    calculate_time(file_size);

    return 0;
}
```

Exercise 4: Write a program to add an 8% sales tax to a given amount and round the result to the nearest penny.

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

double calculate_total_with_tax(double amount, double tax_rate)
{
    double tax = amount * tax_rate;
    double total = amount + tax;

    // Round to the nearest penny

    total = (int)(total * 100 + 0.5) / 100.0;
    return total;
}

int main() {
    double amount;
    const double TAX_RATE = 0.08; // 8% tax


    printf("Enter the amount in dollars: ");
    scanf("%lf", &amount);
```

```
    if (amount < 0)
    {
        printf("Invalid amount. Please enter a positive value.\n");
        return 1;
    }


    double total = calculate_total_with_tax(amount, TAX_RATE);

    printf("The total amount including 8%% sales tax is: $%.2f\n", total);

    return 0;
}
```

Exercise 5: Write a program to tell if a number is prime.

```
 #include <stdio.h>

int isprime(int);

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

    int res = isprime(num);
    res == 0 ? printf("Prime number\n") : printf("Not a prime number\n");
}
```

```
int isprime(int num)
{
   for(int i=2; i<num; i++)
   {
      if(num % i == 0)
      return 1;
   }
   return 0;
}
```

Exercise 6: Write a program that takes a series of numbers and counts the number of positive and negative values.

```
#include <stdio.h>

int main()
{
   int number;
   int positive_count = 0, negative_count = 0;

   printf("Enter a series of integers (enter 0 to stop):\n");

   while (1)
   {
      scanf("%d", &number);

      if (number == 0)
      {
         break;
      }

      if (number > 0)
```

```
        {
            positive_count++;
        }
        else if (number < 0)
        {
            negative_count++;
        }
    }
    printf("Number of positive numbers: %d\n", positive_count);
    printf("Number of negative numbers: %d\n", negative_count);

    return 0;
}
```

1.C program to find the HCF (Highest Common Factor) of given numbers using recursion

```
#include <stdio.h>

int findHCF(int a, int b);

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

    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    hcf = findHCF(num1, num2);

    printf("The HCF of %d and %d is: %d\n", num1, num2, hcf);

    return 0;
```

```
}

int findHCF(int a, int b)
{
   if (b == 0)
   {
      return a;
   }
   return findHCF(b, a % b);
}
```

2. C program to find the LCM (Lowest Common Multiple) of given numbers using recursion

```
#include <stdio.h>

int findHCF(int a, int b);
int findLCM(int a, int b);

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

   printf("Enter two numbers: ");
   scanf("%d %d", &num1, &num2);

   lcm = findLCM(num1, num2);

   printf("The LCM of %d and %d is: %d\n", num1, num2, lcm);

   return 0;
}

int findHCF(int a, int b)
```

```
{
    if (b == 0)
    {
        return a;
    }
    return findHCF(b, a % b);
}


int findLCM(int a, int b)
{
    return (a * b) / findHCF(a, b);
}
```

3. C program to find the GCD (Greatest Common Divisor) of given numbers using recursion

```
include <stdio.h>

int findGCD(int, int);

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

    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    gcd = findGCD(num1, num2);

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

    return 0;
}
```

```
int findGCD(int a, int b)
{
   if (b == 0)
   {
      return a;
   }
   return findGCD(b, a % b);
}
```

4. C program to convert a Decimal number to Binary using Recursion.

```
#include <stdio.h>

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

   DecimaltoBinary(num);

   return 0;
}
void DecimaltoBinary(int num)
{
   if(num > 1)
   DecimaltoBinary(num>>1);

   printf("%d", num&1);
}
```

5. C program to convert a Binary number to Gray Code

```
#include <stdio.h>

void printBinary(int n) ;
int binaryToGray(int n) ;

int main()
{
   int num;

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

   int grayCode = binaryToGray(num);

   printf("Binary representation: ");
   printBinary(num);
   printf("\n");

   printf("Gray code representation: ");
   printBinary(grayCode);
   printf("\n");

   return 0;
}


int binaryToGray(int n)
{
   return n ^ (n >> 1);
}
```

```
void printBinary(int n)
{
   for (int i = 31; i >= 0; i--)
   {
      printf("%d", (n >> i) & 1);
   }
}
```

6. C program to convert a Binary number to Gray Code using Recursion

```
#include <stdio.h>

int bintogray(int);

int main ()
{
   int bin, gray;

   printf("Enter a binary number: ");
   scanf("%d", &bin);
   gray = bintogray(bin);
   printf("The gray code of %d is %d\n", bin, gray);
   return 0;
}

int bintogray(int bin)
{
   int a, b, result = 0, i = 0;

   if (!bin)
   {
```

```
        return 0;
    }
    else
    {
        a = bin % 10;
        bin = bin / 10;
        b = bin % 10;
        if ((a && !b) || (!a && b))
        {
            return (1 + 10 * bintogray(bin));
        }
        else
        {
            return (10 * bintogray(bin));
        }
    }
}
```

7. C program to print the following pyramid.

```
* * * * * * * * * *
* * * *   * * * *
* * *     * * *
* *       * *
*         *
```

```
#include <stdio.h>

int main()
{
    int n;
    printf("Enter the number of rows: ");
    scanf("%d", &n);
```

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

        for(int j = 0; j < n - i; j++)
        {
            printf("* ");
        }



        for(int j = 0; j < 2 * i ; j++)
        {
            printf("  ");
        }



        for(int j = 0; j < n - i; j++)
        {
                printf("* ");
        }



        printf("\n");
    }

    return 0;
}
```

8.C program to find the sum of Natural Number/Factorial of Number of all natural numbers from 1 to N.

Series: 1/1! + 2/21 + 3/3! + 4/4! + N/N!

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

```
int sumofnbyf(int num);
int fact(int num);

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

     printf("The sum of n/fact(n) for all numbers up to %d is %d\n", num, sumofnbyf(num));
}



int sumofnbyf(int num)
{
   int sum = 0;
   for(int i=1; i<=num; i++)
   {
      sum+= i/fact(i);
   }
   return sum;
}

int fact(int num)
{
   int fact=1;
   while(num>=1)
   {
      fact = fact * num;
      --num;
   }
```

```
    return fact;
}
```

9. C program to find sum of following series:

1+3^2/3^3 + 5^2/5^3+7^2/7^3 + ... till N terms

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

double sumofseries(int limit);

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

    printf("The sum of the series of %d terms is %.5lf", limit, sumofseries(limit));

    return 0;
}

double sumofseries(int limit)
{
    double sum = 0;
    int num = 1;
    for(int i=1; i<=limit; i++)
    {
        if(num%2)
        {
            sum+=pow(num,2)/pow(num,3);
        }
        else
```

```
        {
            i--;
        }
        num++;
    }
    return sum;
}
```

10. C program to replace all EVEN elements by 0 and Odd by 1 in One Dimensional Array

```
#include <stdio.h>

int main()
{
    int s;
    printf("Enter the size: ");
    scanf("%d", &s);

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

    for(int i=0; i<s; i++)
    {
        if(arr[i] % 2)
            arr[i] = 1;
        else
            arr[i] = 0;
    }
```

```
    printf("After modification\n");
    for(int i=0; i<s; i++)
    {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

11. C Program to Read a Matrix and Print Diagonals

```
#include <stdio.h>

int main()
{
    int n;
    printf("Enter the value of n for (n*n) matrix: ");
    scanf("%d", &n);

    int arr[n][n];

    printf("Enter the elements of the matrix: \n");
    for(int i=0; i<n; i++)
    {
        for(int j=0; j<n; j++)
        {
            scanf("%d", &arr[i][j]);
        }
    }

    printf("The diagonal elements are:\n");
    for(int i=0; i<n; i++)
    {
```

```
        for(int j=0; j<n; j++)
        {
            if(i == j || j == n-1-i)
            printf("%d ",arr[i][j]);
            else
            printf("  ");
        }
        printf("\n");
    }

    return 0;
}
```

12. C program to print the upper triangular portion of a 3x3matrix

```
#include <stdio.h>

int main()
{
    int matrix[3][3];
    printf("Enter the elements of the 3x3 matrix:\n");


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


    printf("Upper triangular portion of the matrix is:\n");
```

```
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {

            if (i <= j)
            {
                printf("%d ", matrix[i][j]);
            }
            else
            {
                printf("  ");
            }
        }
        printf("\n");
    }

    return 0;
}
```

13. C program to input and print text using Dynamic Memory Allocation.

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

int main()
{
    char *text;
    int length = 10;



    text = (char *)malloc(length * sizeof(char));
```

```
    if (text == NULL)
    {
        printf("Memory allocation failed.\n");
        return 1;
    }

    printf("Enter the text: ");
    scanf("%[^\n]", text);

    printf("\nYou entered: %s\n", text);
    free(text);

    return 0;
}
```

14. C program to read a one dimensional array, print sum of all elements along with inputted array elements using Dynamic Memory Allocation.

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

int main()
{
    int *arr;
    int n, sum = 0;

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

    arr = (int *)malloc(n * sizeof(int));
```

```
if (arr == NULL)
{
   printf("Memory allocation failed!\n");
   return 1;
}


printf("Enter %d elements:\n", n);
for (int i = 0; i < n; i++)
{
   scanf("%d", &arr[i]);
}


for (int i = 0; i < n; i++)
{
   sum += arr[i];
}


printf("\nArray elements are: ");
for (int i = 0; i < n; i++)
{
   printf("%d ", arr[i]);
}

printf("\nSum of all elements: %d\n", sum);


free(arr);
```

```
    return 0;
}
```