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>

int main() {
  int choice;
  double value, result;

  do {
    printf("\nUnit Conversion Menu:\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. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    if (choice >= 1 && choice <= 4) {
      printf("Enter the value to convert: ");
      scanf("%lf", &value);
    }

    switch (choice) {
      case 1:
        result = value * 1.60934;
        printf("%.2f miles = %.2f kilometers\n", value, result);
```

```
            break;
          case 2:
            result = value * 3.78541;
            printf("%.2f gallons = %.2f liters\n", value, result);
            break;
          case 3:
            result = value * 0.453592;
            printf("%.2f pounds = %.2f kilograms\n", value, result);
            break;
          case 4:
            result = value * 2.54;
            printf("%.2f inches = %.2f centimeters\n", value, result);
            break;
          case 5:
            printf("Exiting program.\n");
            break;
          default:
            printf("Invalid choice. Try again.\n");
        }
    } while (choice != 5);

    return 0;
}
```

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 code design

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

```
}



int isLeapYear(int year) {
  return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
}


int countDaysInYear(int day, int month, int year) {
  int daysInMonths[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
  int totalDays = day;

  for (int i = 0; i < month - 1; i++) {
    totalDays += daysInMonths[i];
  }

  if (month > 2 && isLeapYear(year)) {
    totalDays++;
  }

  return totalDays;
}

int totalDaysSinceYearZero(int day, int month, int year) {
  int totalDays = 0;

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

  totalDays += countDaysInYear(day, month, year);
```

```
    return totalDays;
}

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

    printf("Enter the first date (DD/MM/YY): ");
    scanf("%d/%d/%d", &day1, &month1, &year1);

    printf("Enter the second date (DD/MM/YY): ");
    scanf("%d/%d/%d", &day2, &month2, &year2);

    year1 += 1900; // Convert 2-digit year to 4-digit year
    year2 += 1900;

    int totalDays1 = totalDaysSinceYearZero(day1, month1, year1);
    int totalDays2 = totalDaysSinceYearZero(day2, month2, year2);

    int difference = totalDays2 - totalDays1;

    printf("The total number of days between the two dates is: %d\n", difference);

    return 0;
}
```

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.)

```c
#include <stdio.h>

int main() {
    double fileSize, transmissionRate = 960;
    double timeInSeconds, timeInMinutes, timeInHours, timeInDays;

    printf("Enter the file size in bytes: ");
    scanf("%lf", &fileSize);

    timeInSeconds = fileSize / transmissionRate;
    timeInMinutes = timeInSeconds / 60;
    timeInHours = timeInMinutes / 60;
    timeInDays = timeInHours / 24;

    printf("Transmission time:\n");
    printf("%.2f seconds\n", timeInSeconds);
    printf("%.2f minutes\n", timeInMinutes);
    printf("%.2f hours\n", timeInHours);
    printf("%.2f days\n", timeInDays);

    return 0;
}
```

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

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

int main() {
    double amount, total;

    printf("Enter the amount: ");

    scanf("%lf", &amount);

    total = amount * 1.08;

    total = round(total * 100) / 100;

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

    return 0;

}
```

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

```
#include <stdio.h>


int isPrime(int num) {

    if (num <= 1) return 0;

    for (int i = 2; i * i <= num; i++) {

        if (num % i == 0) return 0;

    }

    return 1;

}


int main() {

    int num;

    printf("Enter a number: ");

    scanf("%d", &num);

    if (isPrime(num)) {

        printf("%d is a prime number.\n", num);
    } else {
        printf("%d is not a prime number.\n", num);
    }
    return 0;
}
```

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 num, positiveCount = 0, negativeCount = 0, n;

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

    printf("Enter the numbers:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &num);
        if (num > 0) {
            positiveCount++;
        } else if (num < 0) {
            negativeCount++;
        }
    }

    printf("Positive numbers: %d\n", positiveCount);
    printf("Negative numbers: %d\n", negativeCount);


    return 0;
}
```

7. program to find hcf of a given numbers using recursion

```
#include <stdio.h>


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


int main() {
    int num1, num2;
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);
    printf("HCF of %d and %d is: %d\n", num1, num2, hcf(num1, num2));
    return 0;
}
```

8.. program to find lcm of a given numbers using recursion

```
#include <stdio.h>


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

int lcm(int a, int b) {
   return (a * b) / hcf(a, b);
}

int main() {
   int num1, num2;

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

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

   return 0;
}
```

9.program to find gcd of a given numbers using recursion

```
#include <stdio.h>

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

int main() {
   int num1, num2;

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

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

   return 0;
}
```

10.program to convert decimal to binary

```
#include <stdio.h>

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

int main() {
    int num;

    printf("Enter a decimal number: ");
    scanf("%d", &num);
    printf("Binary representation of %d is: ", num);
    if (num == 0) {
        printf("0");
    } else {
        decimalToBinary(num);
    }
    printf("\n");

    return 0;
}
```

11.binary to gray

```
#include <stdio.h>

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

void printGrayCode(int n) {
    int gray = binaryToGray(n);
    for (int i = (1 << (sizeof(n) * 8 - 1)); i > 0; i >>= 1) {
        printf("%d", (gray & i) ? 1 : 0);
    }
}

int main() {
    int num;

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

    printf("Gray code of %d is: ", num);
    if (num == 0) {
        printf("0");
    } else {
        printGrayCode(num);
    }
    printf("\n");

    return 0;
}
```

12.binary to gray using recursion

```
#include <stdio.h>

void binaryToGray(int n) {
    if (n == 0) {
        return;
    }
    binaryToGray(n / 2);
    printf("%d", n ^ (n >> 1) % 2);
}

int main() {
    int num;

    printf("Enter a decimal number: ");
    scanf("%d", &num);
    printf("Gray code of %d is: ", num);
    if (num == 0) {
        printf("0");
    } else {
        binaryToGray(num);
    }
    printf("\n");

    return 0;
}
```

13. 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 + 3/31 + 4/4! + ... N/N!

```
#include <stdio.h>

int main() {
    int N, i, j;
    double sum = 0.0, fact;

    printf("Enter value of N: ");
    scanf("%d", &N);

    for (i = 1; i <= N; i++) {
        fact = 1.0;
        for (j = 1; j <= i; j++) {
            fact *= j;
        }
        sum += i / fact;
    }

    printf("Sum of the series: %.2lf\n", sum);
    return 0;
}
```

14. C program to find sum of following series:

1+3^2/3^3+5^2/5^3+7^2/7^3 + ... till N terms 10. C program to replace all EVEN elements by 0 and Odd by 1 in One Dimensional Array

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

int main() {
    int N, i;
    double sum = 0.0;

    printf("Enter number of terms (N): ");
    scanf("%d", &N);

    for (i = 1; i <= N; i++) {
        int term = 2 * i - 1; // Odd numbers: 1, 3, 5, ...
        sum += pow(term, 2) / pow(term, 3);
    }

    printf("Sum of the series: %.2lf\n", sum);
    return 0;
}
```

15. C Program to Read a Matrix and Print Diagonals

```
#include <stdio.h>

int main() {
    int m, n;

    printf("Enter the number of rows and columns of the matrix: ");
    scanf("%d %d", &m, &n);

    if (m != n) {
        printf("The matrix is not square. Diagonals can only be printed for square
matrices.\n");
        return 1;
    }

    int matrix[m][n];

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

    printf("Main diagonal:\n");
    for (int i = 0; i < m; i++) {
        printf("%d ", matrix[i][i]);
    }
    printf("\n");

    printf("Secondary diagonal:\n");
    for (int i = 0; i < m; i++) {
        printf("%d ", matrix[i][m - i - 1]);
    }
    printf("\n");

    return 0;
}
```

16. C program to print the upper triangular portion of a matrix

```
#include <stdio.h>

int main() {
    int matrix[3][3], i, j;

    printf("Enter 3x3 matrix elements:\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }

    printf("Upper triangular portion:\n");
    for (i = 0; i < 3; i++) {
```

```
    for (j = 0; j < 3; j++) {
      if (j >= i)
        printf("%d ", matrix[i][j]);
      else
        printf("  "); // Empty space for formatting
    }
    printf("\n");
  }

  return 0;
}
```

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

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

int main() {
  char *text;
  int n;

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

  text = (char *)malloc((n + 1) * sizeof(char)); // Allocate memory
  if (text == NULL) {
    printf("Memory allocation failed.\n");
    return 1;
  }
```

```
    printf("Enter text: ");
    scanf(" ");
    fgets(text, n + 1, stdin);

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

    free(text); // Free memory
    return 0;
}
```

18. 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 n, i, sum = 0;
    int *arr;

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

    arr = (int *)malloc(n * sizeof(int)); // Allocate memory
    if (arr == NULL) {
        printf("Memory allocation failed.\n");
        return 1;
    }
```

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

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

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

  free(arr); // Free memory
  return 0;
}
```