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>
#include<string.h>
void convertToMetric(float,char[]);
int main()
{
  float value;
  char unit[10];
  printf("enter the value to be converted and unit:");
  scanf("%f %s",&value,unit);
  convertToMetric(value,unit);

  return 0;
}
void convertToMetric(float value,char unit[10]){
  float org=value;
  if(strcmp(unit,"miles")==0){
    value=value*1.60;
    printf("%.2f miles=%.2f km",org,value);
    return;
  }
  else if(strcmp(unit,"gallons")==0){
    value=value*3.78;
    printf("%.2f gallons=%.2f liter",org,value);
    return;
  }
  else if (strcmp(unit, "pounds") == 0) {
    value=value* 0.453592;
    printf("%.2f pounds = %.2f kilograms\n",org,value);
  }
  else if (strcmp(unit, "inches") == 0) {
    value=value* 2.54;
    printf("%.2f inches = %.2f centimeters\n",org,value);
  }
  else {
    printf("Invalid unit entered. Supported units: miles, gallons, pounds, inches.\n");
  }


}
```

-_

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>

typedef struct Date {
  int day;
  int month;
  int year;
} DATE;

int isLeapYear(int);
int daysInMonth(int, int);
```

```
int calculateDays(DATE);

int main() {
   DATE date1 = {6, 6, 1990};
   DATE date2 = {4, 3, 1992};

   int totalDays1 = calculateDays(date1);
   int totalDays2 = calculateDays(date2);

   int diffDays = totalDays2 - totalDays1;

   printf("The number of days between %.2d/%.2d/%d and %.2d/%.2d/%d is: %d days\n",
        date1.day, date1.month, date1.year, date2.day, date2.month, date2.year, diffDays);

   return 0;
}

int isLeapYear(int year) {
   if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
      return 1;
   }
   return 0;
}

int daysInMonth(int month, int year) {
   switch (month) {
      case 1: case 3: case 5: case 7: case 8: case 10: case 12:
         return 31;
      case 4: case 6: case 9: case 11:
         return 30;
      case 2:
         return isLeapYear(year) ? 29 : 28;
      default:
         return 0;
   }
}

int calculateDays(DATE date) {
   int totalDays = 0;


   for (int y = 1900; y < date.year; y++) {
      totalDays = totalDays + (isLeapYear(y) ? 366 : 365);
   }


   for (int m = 1; m < date.month; m++) {
      totalDays = totalDays + daysInMonth(m, date.year);
   }


   totalDays = totalDays + date.day;

   return totalDays;
}
```

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>
#include <math.h>
#define Mb 1024
#define kb 1024
#define rate 960
int main(){
  int fileSize;
  printf("enter the file size:");
  scanf("%d",&fileSize);
  float noOfDays;
  float SizeinBytes=(fileSize * Mb)* kb;
  float BytesinSec=round(SizeinBytes/960);
  printf("total bytes in sec=%f\n",BytesinSec);
  noOfDays=BytesinSec/(24*60*60);
  printf("a %dMb file takes %.2f days",fileSize,noOfDays);

}
```

-_

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>

int main() {
  float amount, totalAmount, salesTax;
  printf("Enter the amount: $");
  scanf("%f", &amount);
  salesTax = amount * 0.08;
  totalAmount = amount + salesTax;
  printf("before rounding: $%.2f\n", totalAmount);

  totalAmount = round(totalAmount);
  printf("Total amount after 8%% sales tax: $%.2f\n", totalAmount);

  return 0;
}
```

-_

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

```
#include <stdio.h>

int isPrime(int num);

int main() {
  int number;
  printf("Enter a number to check if it's prime: ");
  scanf("%d", &number);

  if (isPrime(number)) {
    printf("%d is a prime number.\n", number);
  } else {
    printf("%d is not a prime number.\n", number);
  }

  return 0;
}
```

```
int isPrime(int num) {
   if (num <= 1) {
      return 0;
   }


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

   return 1;
}
```
-_

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

   int arr[n];

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

   printf("Number of positive values: %d\n", positiveCount);
   printf("Number of negative values: %d\n", negativeCount);

   return 0;
}
```
-_

```
 //1.HCF using recursion
#include<stdio.h>

int HCF(int, int, int);

int main(){
   int num1, num2;
   printf("Enter the two numbers: ");
   scanf("%d %d", &num1, &num2);
```

```
    int hcf = HCF(num1, num2, (num1 < num2 ? num1 : num2));
    printf("HCF = %d\n", hcf);
    return 0;
}

int HCF(int a, int b, int i) {

    if (i == 0) {
        return i;
    }
    if (a % i == 0 && b % i == 0) {
        return i;
    }

    // Recursive case: try with the next smaller number
    return HCF(a, b, i - 1);
}-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
//2.LCM using recursion
#include<stdio.h>
unsigned int LCM(unsigned int,unsigned int,unsigned int);
int main(){
    printf("enter the two numbers:");
    unsigned int num1,num2;
    scanf("%u %u",&num1,&num2);
    unsigned int largest =num1>num2?num1:num2;
    unsigned int lcm=LCM(num1,num2,largest);
    printf("LCM = %u",lcm);
    return 0;
}
unsigned int LCM(unsigned int n1,unsigned int n2,unsigned int l){

    if(l%n1==0 && l%n2==0){
        return l;
    }
    l=l+(n1>n2?n1:n2);
    return LCM(n1,n2,l);
}
-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
//3.Decimal to binary
#include<stdio.h>
int decTobin(int);
int main(){
    printf("enter the number in decimal format:");
    int num1;
    scanf("%d",&num1);
    int inBin=decTobin(num1);
    printf("binary of %d is %d",num1,inBin);
    return 0;
}
int decTobin(int n1){
    int bin=0;
    int pos=1;
    while(n1!=0){
        bin=bin+(n1%2)*pos;
        n1=n1/2;
        pos=10*pos;
```

```
  }
  return bin;
}
```

-_-

```
//5.binary to Gray code
#include <stdio.h>

int binTogray(int);
int decTobin(int);

int main() {
  printf("Enter the number in decimal format: ");
  int num1;
  scanf("%d", &num1);

  int inBin = decTobin(num1);
  printf("Binary representation of %d is %d\n", num1, inBin);

  int ingray = binTogray(num1);


  return 0;
}

// Function to convert binary to Gray code
int binTogray(int n1) {
  int gray = n1 ^ (n1 >> 1);
  int inBin = decTobin(gray);
  printf("Gray code of %d in binary is %d\n", n1, inBin);
  return inBin;
}

// Function to convert decimal to binary
int decTobin(int n1) {
  int bin = 0;
  int pos = 1;

  while (n1 != 0) {
    bin = bin + (n1 % 2) * pos;
    n1 = n1 / 2;
    pos = 10 * pos;
  }
  return bin;
}
```

-_-

```
//pyramid.
**********
****  ****
***      ***
**          **
*              *
#include <stdio.h>

int main() {
  int n = 5;
```

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

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


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


        for (int j = i; j < n; 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

```
#include <stdio.h>
unsigned long long factorial(int num) {
    unsigned long long fact = 1;
    for (int i = 1; i <= num; i++) {
        fact *= i;
    }
    return fact;
}

int main() {
    int N;

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

    if (N <= 0) {
        printf("Please enter a positive integer.\n");
        return 1;
    }

    unsigned long long sum_of_natural_numbers = 0;
    unsigned long long sum_of_factorials = 0;

    for (int i = 1; i <= N; i++) {
        sum_of_natural_numbers += i;
        sum_of_factorials += factorial(i);
    }

    printf("Sum of natural numbers from 1 to %d: %llu\n", N, sum_of_natural_numbers);
    printf("Sum of factorials of natural numbers from 1 to %d: %llu\n", N, sum_of_factorials);

    return 0;
}
```

```
-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
//9. C program to find sum of following series:
1 + 3^2/303 + 5^2/543 + 7^2/7^3 + ... till N terms
#include <stdio.h>
#include <math.h>

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

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

  if (N <= 0) {
    printf("Please enter a positive integer.\n");
    return 1; // Exit if invalid input
  }

  for (int i = 1; i <= N; i++) {
    int term = 2 * i - 1; // Generate odd numbers: 1, 3, 5, ...
    double numerator = pow(term, 2); // term^2
    double denominator = pow(term, 3); // term^3
    sum += numerator / denominator; // Add term to the sum
  }

  printf("Sum of the series up to %d terms: %.6f\n", N, sum);

  return 0;
}
-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
//10. C program to replace all EVEN elements by 0 and Odd by 1 in One Dimensional Array
#include <stdio.h>

int main() {
  int n;


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

  if (n <= 0) {
    printf("Invalid array size.\n");
    return 1;
  int arr[n];


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


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

```
    }
  }


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

  return 0;
}
```

---

11. C Program to Read a Matrix and Print Diagonals

```
#include <stdio.h>

int main() {
  int n;

  // Input the size of the square matrix
  printf("Enter the size of the matrix (n x n): ");
  scanf("%d", &n);

  if (n <= 0) {
    printf("Invalid matrix size.\n");
    return 1; // Exit if invalid size
  }

  int matrix[n][n];

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

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

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

  return 0;
}
```

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

```c
#include <stdio.h>

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

  // Input the matrix elements
  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]);
    }
  }

  // Print the upper triangular portion
  printf("Upper triangular portion:\n");
  for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
      if (j >= i) {
        printf("%d ", matrix[i][j]);
      } else {
        printf("  "); // Leave space for alignment
      }
    }
    printf("\n");
  }

  return 0;
}
```

-_-

```c
//13. C program to input and print text using Dynamic Memory Allocation.
#include <stdio.h>
#include <stdlib.h>

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


  printf("Enter the number of characters in the text (including spaces): ");
  scanf("%d", &size);


  text = (char *)malloc((size + 1) * sizeof(char));
  if (text == NULL) {
    printf("Memory allocation failed!\n");
    return 1;
  }


  printf("Enter the text: ");
  getchar();
  fgets(text, size + 1, stdin);


  printf("You entered: %s", 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 Dynamid Memory Allocation.
#include <stdio.h>
#include <stdlib.h>

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


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


    array = (int *)malloc(n * sizeof(int));
    if (array == NULL) {
        printf("Memory allocation failed!\n");
        return 1;
    }


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


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


    printf("Array elements: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", array[i]);
    }
    printf("\nSum of all elements: %d\n", sum);


    free(array);

    return 0;
}
```
