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>

//Conversion factors
#define MILES_TO_KM 1.60934
#define GALLONS_TO_LITERS 3.78541
#define POUNDS_TO_KG 0.453592
#define INCHES_TO_CM 2.78541

int main(){
    int op;
    float value;
    do{


    printf("Select the conversion type:\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("Enter your choice (1-4): ");
    scanf("%d",&op);

    switch(op){
        case 1:{
            float result;
            printf("Enter value in miles: ");
            scanf("%f", &value);
            result = value * MILES_TO_KM;
            printf("%.2f miles is equal to %.2f kilometers.\n", value, result);
            break;
        }
        case 2:{
            float result;
            printf("Enter value in Gallons: ");
            scanf("%f", &value);
            result = value * GALLONS_TO_LITERS;
            printf("%.2f gallons is equal to %.2f Liters.\n", value, result);
            break;
        }
        case 3:{
            float result;
            printf("Enter value in Pounds: ");
            scanf("%f", &value);
            result = value * POUNDS_TO_KG;
            printf("%.2f Pounds is equal to %.2f Kg.\n", value, result);
            break;
        }
        case 4:{
            float result;
            printf("Enter value in in: ");
            scanf("%f", &value);
            result = value * INCHES_TO_CM;
            printf("%.2f Inches is equal to %.2f cm.\n", value, result);
            break;
        }
        case 0:
            printf("Exiting...\n");
            break;
        default:
            printf("Invalid option!");
    }

    }while(op != 0);
}
```

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>


int isLeapYear(int year) {
    if (year % 4 == 0) {
        if (year % 100 == 0) {
            if (year % 400 == 0)
                return 1;
            else
                return 0;
        } else
            return 1;
    } else
        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; // Invalid month
    }
}


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


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


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


    totalDays += day;

    return totalDays;
}

int main() {
    int day1, month1, year1;
    int 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);

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

    int daysBetween = totalDays2 - totalDays1;

    printf("The number of days between %d/%d/%d and %d/%d/%d is %d days.\n",
           day1, month1, year1, day2, month2, year2, daysBetween);

    return 0;
}


/*Exercise 3: A serial transmission line can transmit 960 characters each second.*/

#include <stdio.h>


#define TRANSMISSION_RATE 960 // 960 characters per second

void calculateTransmissionTime(long long fileSize) {
    long long totalSeconds = fileSize / TRANSMISSION_RATE;

    int days = totalSeconds / (24 * 3600);
    totalSeconds %= (24 * 3600);
    int hours = totalSeconds / 3600;
    totalSeconds %= 3600;
    int minutes = totalSeconds / 60;
    int seconds = totalSeconds % 60;

    printf("Transmission time:\n");
    if (days > 0) printf("%d days ", days);
    if (hours > 0 || days > 0) printf("%d hours ", hours);
    if (minutes > 0 || hours > 0 || days > 0) printf("%d minutes ", minutes);
    printf("%d seconds\n", seconds);
}

int main() {
    long long fileSize;
    printf("Enter the file size in bytes: ");
    scanf("%lld", &fileSize);

    calculateTransmissionTime(fileSize);

    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>

#define SALES_TAX_RATE 0.08 // 8% sales tax

int main() {
    float amount, total;

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


    total = amount + (amount * SALES_TAX_RATE);


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

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

    return 0;
}
```

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

```
#include <stdio.h>

int isPrime(int number) {
    if (number <= 1) {
        return 0;
    }
    for (int i = 2; i * i <= number; i++) {
        if (number % i == 0) {
            return 0;
        }
    }
    return 1;
}

int main() {
    int number;

    printf("Enter a number to check if it is 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;
}
```

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 count, positiveCount = 0, negativeCount = 0;

    printf("Enter the number of elements (up to 100): ");
    scanf("%d", &count);


    if (count <= 0 || count > 100) {
        printf("Invalid number of elements. Please enter a number between 1 and 100.");
        return 1;
    }

    int numbers[count];
    printf("Enter the %d numbers :",count);

    for (int i = 0; i < count; i++) {
        printf("\nEnter number %d: ", i + 1);
        scanf("%d", &numbers[i]);
    }


    for (int i = 0; i < count; i++) {
        if (numbers[i] > 0) {
            positiveCount++;
        } else if (numbers[i] < 0) {
            negativeCount++;
        }
    }

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

    return 0;
}
```

```
*C programme to find HCF of given number*/

#include <stdio.h>

int findHCF(int a,int b);
 int main(){
    int num1,num2;
    printf("Enter two number:");
    scanf("%d %d",&num1,&num2);

    int HCF = findHCF(num1,num2);
    printf("HCF of %d and %d is %d",num1,num2,HCF);
 }

 int findHCF(int a,int b){

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

```
C programme to find LCM of numbers using recursion*/

#include <stdio.h>

int findLCM(int a,int b,int lcm);
int main(){
    int num1,num2;
    printf("Enter two number:");
    scanf("%d %d",&num1,&num2);

    int LCM = findLCM(num1,num2,(num1 > num2) ? num1 : num2);
    printf("LCM of %d and %d is %d",num1,num2,LCM);
 }


 int findLCM(int a,int b,int lcm){


    if(lcm%a == 0 && lcm%b == 0){
       return lcm;
    }
    return findLCM(a,b,lcm+1);

 }
```

```
C programme to find GCD of two numbers using recursion*/

#include <stdio.h>

int findGCD(int a,int b);
 int main(){
    int num1,num2;
    printf("Enter two number:");
    scanf("%d %d",&num1,&num2);

    int GCD = findGCD(num1,num2);
    printf("GCD of %d and %d is %d",num1,num2,GCD);
 }

 int findGCD(int a,int b){

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

C programme to convert decimal to binary using recursion*/

```
#include <stdio.h>

int deci_to_bin(int n);
int main(){
   int num;
   printf("Enter number to be converted:");
   scanf("%d",&num);

   printf("Binary representation of %d is: ", num);
   if(num == 0){
      printf("0");
   }
   else{
      deci_to_bin(num);
   }
}

int deci_to_bin(int n){

   if(n > 1){
      deci_to_bin(n/2);
   }
   printf("%d ",n%2);
}
```

C programme to convert binary to gray code*/

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


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

int main() {
   int num;


   printf("Enter a binary number (as an integer): ");
```

```
    scanf("%d", &num);


    int gray = binaryToGray(num);
    printf("Gray code of %d is: %d\n", num, gray);

    return 0;
}


/*PATTERN*/

#include <stdio.h>
int main(){

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

    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n-i;j++)
        {
            printf("*");
        }
        for(int sp=1;sp<=i;sp++)
        {
            printf(" ");
        }
        for(int sp=1;sp<=i;sp++)
        {
            printf(" ");
        }
        for(int j=0;j<n-i;j++)
        {
            printf("*");
        }
        printf("\n");
    }
}


*C program to calculate sum of n natural numbers /factorial*/

#include <stdio.h>

int findSum(int );
int findFact(int);
int findFact(int);
int main(){
    int n;
    printf("Enter number:");
    scanf("%d",&n);
    int op;
    do{

        printf("choose option:\n1->finf sum of n natural numbers\n2->to finfd factorial of n0->To exit\n");
        scanf("%d",&op);
        switch(op){

            case 1:
            int sum = findSum(n);
            printf("Sum of %d numbers is %d",n,sum);
            break;

            case 2:
            int fact = findFact(n);
            printf("Factorial of %d is %d",n,fact);
            break;

            case 0:
            printf("Exiting...\n");
            break;


            default:
            printf("Invalid option\n");
        }
    }while(op != 0);




}

int findSum(int n){
    int sum =0;
    for(int i=1;i<=n;i++)
    {
        sum += i;
    }
    return sum;
}

int findFact(int n){
    if(n == 0 || n == 1){
        return 1;
    }
    else{
        return n*findFact(n-1);

        }

}
```

To find sum of a series?

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

int main(){
    int n;
    printf("Enter the limit:");
    scanf("%d",&n);
    float sum = 1;
    printf("The series is 1 + ");
    for(int i=2;i<=n;i++){
        if(i%2 != 0){
            float term = pow(i, 2) / pow(i, 3);
            sum += term ;
            printf("%d^2 / %d^3 +",i,i);

        }
    }

    printf("The sum of series is %.2f",sum);
}
```

Cprogramme to replace all even elements as 0 and odd elements as 1 in an 1d array*/

```
#include <stdio.h>

int main(){
    int n;
    printf("Enter size of an array:");
    scanf("%d",&n);

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

    }


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

        printf("%d ",arr[i]);
    }

    printf("Array after modification\n");
    for(int i=0;i<n;i++){
        if(arr[i]%2 == 0){
            arr[i] = 0;
        }
        else{
            arr[i] = 1;
        }
        printf("%d ",arr[i]);
    }

}
```

c program to read a matrix and print diagonals*/

```
#include <stdio.h>

int main() {
    int n;
    printf("Enter the size of the matrix (n x n): ");
    scanf("%d", &n);

    int matrix[n][n];


    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]);
        }
    }

    printf("The diagonals are: ");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if(i == j){
                printf("%d ",matrix[i][j]);
            }
        }
    }
}
```

```
/*C program to print the upper triangular portion of a 3*3 matrix*/

#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:\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;
}
```

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

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


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


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


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


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


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


   free(array);

   return 0;
}
```

Cprogramme to convert binary 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));
       }
   }
}
```