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

```
#include<stdio.h>

void milesToKilometres(float miles);
void gallonsToLitres(float gallons);
void poundsToKilograms(float pounds);

int main(){
   int op;
   float miles,gallons,pounds;
   printf("Choose one option: \n");
   printf("1.Miles to kilometres\n");
   printf("2.Gallons to Litres\n");
   printf("3.Pounds to kilograms\n");
   scanf(" %d",&op);
   switch(op){
      case 1:
         printf("Enter distance in miles: ");
         scanf("%f",&miles);
         milesToKilometres(miles);
         break;
      case 2:
         printf("Enter volume in gallons: ");
         scanf("%f",&gallons);
         gallonsToLitres(gallons);
         break;
      case 3:
         printf("Enter weight in pounds: ");
         scanf("%f",&pounds);
         poundsToKilograms(pounds);
```

```
            break;
        default:
            printf("Incorrect option!");
            break;
    }

    return 0;
}
void milesToKilometres(float miles){
    float km = miles * 1.60934;
    printf("%.2f miles = %.2f kilometres\n",miles,km);
}
void gallonsToLitres(float gallons){
    float l = gallons * 3.78541;
    printf("%.2f gallons = %.2f litres\n",gallons,l);
}
void poundsToKilograms(float pounds){
    float kg = pounds * 0.453592;
    printf("%.2f pounds = %.2f kilograms\n",pounds,kg);
}
```

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 leapyear(int year);
int monthdays(int month,int year);
int totaldays(int startday,int startmonth,int startyear,int endday,int endmonth,int endyear);

int main(){
    int startday,startmonth,startyear;
    int endday,endmonth,endyear;
    printf("Enter the start date: ");
```

```
    scanf("%d %d %d",&startday,&startmonth,&startyear);
    printf("Enter the end date: ");
    scanf("%d %d %d",&endday,&endmonth,&endyear);
    int no_of_days = totaldays(startday,startmonth,startyear,endday,endmonth,endyear);
    printf("Total number of days between %d/%d/%d and %d/%d/%d is
%d",startday,startmonth,startyear,endday,endmonth,endyear,no_of_days);
    return 0;
}
int leapyear(int year){
    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
        return 1;
    return 0;
}
int monthdays(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(leapyear(year))? 29: 28;
        default: return 0;
    }
}
int totaldays(int startday,int startmonth,int startyear,int endday,int endmonth,int endyear){
    int count=0;

if(startyear>endyear||(startyear==endyear&&startmonth>endmonth)||(startyear==endyear&&start
month==endmonth&&startday>endday)){
        int tempday=startday,tempmonth=startmonth,tempyear=startyear;
        startday = endday;
        startmonth = endmonth;
        startyear = endyear;
        endday = tempday;
        endmonth = tempmonth;
```

```
        endyear = tempyear;
    }
    while (startyear < endyear || (startyear == endyear && startmonth < endmonth) || (startyear == endyear && startmonth == endmonth && startday < endday)){
        count++;
        startday++;
        if(startday>monthdays(startmonth,startyear)){
            startday=1;
            startmonth++;
        }
        if(startmonth>12){
            startmonth=1;
            startyear++;
        }
    }
    return count;
}
```

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 transmissiontime(long long filesize,int transmissionrate);

int main(){
    int transmissionrate = 960;
    long long filesize = 4194304000;
    transmissiontime(filesize,transmissionrate);
    return 0;
}
void transmissiontime(long long filesize,int transmissionrate){
```

```
    long long timeinseconds = filesize/transmissionrate;
    printf("Time in seconds required to send the file is %lld\n",timeinseconds);
    long long hours = timeinseconds/3600;
    long long minutes = (timeinseconds%3600)/60;
    long long seconds = timeinseconds%60;
    printf("Total time required to send the file: %lld hours,%lld minutes,%lld
seconds",hours,minutes,seconds);
}
```

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>
void salestax(float amount,float taxrate);

int main(){
    float amount,taxrate = 8.0;
    printf("Enter the amount: ");
    scanf("%f",&amount);
    salestax(amount,taxrate);
    return 0;
}
void salestax(float amount,float taxrate){
    float tax = amount * (taxrate/100);
    float totalamount = amount + tax;
    round(totalamount*100)/100;
    printf("Total amount after tax = %.2f",totalamount);
}
```

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

```
#include<stdio.h>

int isprime(int num);
```

```
int main(){
   int num;
   printf("Enter a number: ");
   scanf("%d",&num);
   if(isprime(num)){
      printf("%d is prime",num);
   }
   else{
      printf("%d is not prime",num);
   }
}

int isprime(int num){
   if(num<=1){
      return 0;
   }
   if(num==2||num==3){
      return 1;
   }
   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 num;
```

```
    int positive=0,negative=0;
    while(1){
        printf("Enter a number: ");
        scanf("%d",&num);
        if(num==0){
            break;
        }
        if(num>0){
            positive++;
        }else if(num<0){
            negative++;
        }
    }

    printf("Number of positive numbers: %d\n",positive);
    printf("Number of negative numbers: %d",negative);

    return 0;
}
```

HCF using recursion

```
#include<stdio.h>

int hcf(int a,int b);
int main(){
    int a,b;
    printf("Enter two numbers: ");
    scanf("%d %d",&a,&b);
    int h = hcf(a,b);
    printf("HCF of %d and %d is %d",a,b,h);
```

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

LCM using recursion

```
#include<stdio.h>

int lcm(int a,int b,int multiple);

int main(){
    int a,b;
    printf("Enter two numbers: ");
    scanf("%d %d",&a,&b);
    int l = lcm(a,b,a);
    printf("LCM of %d and %d is %d",a,b,l);
    return 0;
}
int lcm(int a,int b,int multiple){
    if(multiple%b==0){
        return multiple;
    }
    return lcm(a,b,multiple+a);
}
```

GCD using recursion

```
#include <stdio.h>
int gcd(int a, int b);
```

```
int main() {
    int num1, num2;
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);
    int result = gcd(num1, num2);
    printf("GCD of %d and %d is: %d\n", num1, num2, result);

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

Decimal to binary using recursion

```
#include<stdio.h>
void decimaltobinary(int num);

int main(){
    int num;
    printf("Enter the decimal number: ");
    scanf("%d",&num);
    if(num==0){
        printf("0");
    }
    else{
        printf("Binary representation:");
        decimaltobinary(num);
    }
    return 0;
}
```

```
void decimaltobinary(int num){
   if(num==0){
      return;
   }
   decimaltobinary(num/2);
   printf("%d",num%2);
}
```

Binary to gray code

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

int binarytogray(int num);
void printBinary(int num);
int main(){
   int num;
   printf("Enter the number: ");
   scanf("%d",&num);
   int graycode = binarytogray(num);
   printf("Binary number: ");
   printBinary(num);
   printf("Gray code: ");
   printBinary(graycode);

   return 0;
}
int binarytogray(int num){
   return num^(num>>1);
}
void printBinary(int num) {
   int bits = sizeof(num) * 8;
   for (int i = bits - 1; i >= 0; i--) {
```

```
        printf("%d", (num >> i) & 1);
    }
    printf("\n");
}
```

Using recursion

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

void binaryToGray(int n);
int main() {
    int num;
    printf("Enter a binary number: ");
    scanf("%d", &num);
    printf("Gray code: ");
    binaryToGray(num);
    printf("\n");

    return 0;
}
void binaryToGray(int n) {
    if (n == 0) {
        return;
    }
    int bit = n & 1;
    binaryToGray(n >> 1);

    if (n > 1) {
        printf("%d", bit ^ ((n >> 1) & 1));
    } else {
        printf("%d", bit);
    }}
```

Pyramid

```
#include<stdio.h>
int main(){
   int n=5;
   for(int i=0;i<n;i++){
      for(int j=0;j<n-i;j++){
         printf("*");
      }

      for(int j=0;j<2*i-1;j++){
         printf(" ");
      }

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

Sum of Natural number/Factorial of number

```
#include<stdio.h>

int factorial(int num);
int main(){
   int n;
   double sum=0.0;
   printf("Enter the number upto which the sum is to be calculated: ");
   scanf("%d",&n);
   for(int i=1;i<=n;i++){
```

```
        sum+=(double)i/(factorial(i));
    }
    printf("Sum = %.2f",sum);

}
int factorial(int num){
    int fact=1;
    for(int i=1;i<=num;i++){
        fact=fact*i;
    }
    return fact;
}
```

1 + 3^2/3^3+5^2/5^3+7^2/7^3+….

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

int main(){
    int n;
    double sum=0.0;
    printf("Enter the number of terms: ");
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        int odd = 2*i-1;
        sum+=(pow(odd,2)/pow(odd,3));
    }
    printf("Sum of the series is %.3f",sum);
    return 0;
}
```

Even by 0 and odd by 1

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

```
int main(){
   int n;
   printf("Enter the size of the array: ");
   scanf("%d",&n);
   int arr[n];
   printf("Enter the array elements: ");
   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;
      }
      else{
         arr[i]=1;
      }
   }
   printf("Modified array: ");
   for(int i=0;i<n;i++){
      printf("%d ",arr[i]);
   }
}
```

Diagonals of matrix

```
#include<stdio.h>
int main(){
   int n;
   printf("Enter the size of matrix: ");
   scanf("%d",&n);
   int matrix[n][n];
   printf("Enter the matrix elements: ");
```

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

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

Upper triangle of matrix

```
#include<stdio.h>
int main(){
    int matrix[3][3];
    printf("Enter the matrix elements: ");
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            scanf("%d",&matrix[i][j]);
        }
    }
    printf("Matrix: \n");
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            printf("%d ",matrix[i][j]);
        }
        printf("\n");
```

```
    }
    printf("The upper triangle elements are : ");
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            if(i<=j){
                printf("%d ",matrix[i][j]);
            }
        }
    }
    return 0;
}
```

Input and print text using dynamic memory allocation

```
#include<stdio.h>
#include<stdlib.h>
int main(){
    char *str;
    int length;
    printf("Enter the length of the text: ");
    scanf("%d",&length);
    str = (char*)malloc((length+1)*sizeof(char));
    if(str==NULL){
        printf("Memory allocation failed\n");
        return 1;
    }
    getchar();
    printf("Enter the text: ");
    fgets(str,length+1,stdin);
    printf("Your entered string: %s\n",str);
    free(str);
    return 0;
}
```

Sum of array using dynamic memory allocation

```
#include<stdio.h>
#include<stdlib.h>
int main(){
   int *arr = NULL;
   int n,sum=0;
   printf("Enter the number of elements: ");
   scanf("%d",&n);
   arr = (int *)malloc(n*sizeof(int));
   printf("Enter the array elements: ");
   for(int i=0;i<n;i++){
      scanf("%d",&arr[i]);
   }
   printf("Array elements: \n");
   for(int i=0;i<n;i++){
      printf("arr[%d] = %d\n",i,arr[i]);
   }
   for(int i=0;i<n;i++){
      sum+=arr[i];
   }
   printf("\nSum of elements in array = %d",sum);

}
```