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.

```
/*specification: a program to convert English units to metric (i.e., miles to
kilometers, gallons to liters, etc.).
*/
/*Code design
Start the program
Declare integer as choice;
Declare float as n and res;
Print selection conversion type
Print 1.Miles to kilometers
Print 2.Gallons to liters
Read choice
Print Number to convert
Read n;
if choice==1 then
   res=n*1.60934;
   print res;
else if(choice==2)
   res=n*3.78541;
   print res;
else
   print invalid choice
end if
end

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

```
    int choice;
    float n,res;
    printf("Select conversion type:\n");
    printf("1.Miles to kilometers\n");
    printf("2.Gallons to liters\n");
    printf("Enter your choice:\n");
    scanf("%d",&choice);
    printf("Enter number to convert:");
    scanf("%f",&n);
    if(choice==1)
    {
       res=n*1.60934;
       printf("%.2fmiles=%.2f kilometers\n",n,res);
    }
    else if(choice==2)
    {
       res=n*3.78541;
       printf("%.2f gallons=%.2f litres\n",n,res);
    }
    else
    {
       printf("Invalid choice");
    }
    return 0;
}
```

Output:

Select conversion type:

1.Miles to kilometers

2.Gallons to liters

Enter your choice:

1

Enter number to convert:12

12.00miles=19.31 kilometers

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.

/* specification: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.

*/

/*Code design

Start

Declare integer m1,d1,y1,m2,d2,y2

Declare integer days 1,days2,total_Days

Print "Enter date1"

input m1,d1,y1

print "Enter date2"

input m2,d2,y2

if y1==y2 then

set days1=0

set days2=0

for (i=1;i<m1;i++) ,do

if(i==1||i==3||i==5||i==7||i==8||i==10||i==12) then

days1 += 31;

else if (i == 4 || i == 6 || i == 9 || i == 11) {

days1 += 30;

else if (i == 2)

days1 += 28;

end if

days1 += d1;

end for

```
   Add d2 to days2
   SET total_days = days2 - days1
   PRINT "The number of days between ", m1, "/", d1, "/", y1, " and ", m2, "/", d2, "/", y2, " is: ",
total_days, " days"
   ELSE
      PRINT "This program only works for dates in the same year."
   END IF

END

*/
#include<stdio.h>
int main()
{
   int m1, d1, y1, m2, d2, y2;
   int days1 = 0, days2 = 0;
   int total_days = 0;

   printf("Enter date1 (MM/DD/YYYY): ");
   scanf("%d/%d/%d", &m1, &d1, &y1);

   printf("Enter date2 (MM/DD/YYYY): ");
   scanf("%d/%d/%d", &m2, &d2, &y2);

   if (y1 == y2)
   {
      for (int i = 1; i < m1; i++)
      {
         if (i == 1 || i == 3 || i == 5 || i == 7 || i == 8 || i == 10 || i == 12)
         {
            days1 += 31;
```

```
        }
        else if (i == 4 || i == 6 || i == 9 || i == 11)
        {
            days1 += 30;
        }
        else if (i == 2)
        {
            days1 += 28;
        }
    }
    days1 += d1;

    for (int i = 1; i < m2; i++) {
        if (i == 1 || i == 3 || i == 5 || i == 7 || i == 8 || i == 10 || i == 12) {
            days2 += 31;
        } else if (i == 4 || i == 6 || i == 9 || i == 11) {
            days2 += 30;
        } else if (i == 2) {
            days2 += 28;
        }
    }
    days2 += d2;

    total_days = days2 - days1;
    printf("The number of days between %d/%d/%d and %d/%d/%d is: %d days\n", m1, d1, y1, m2,
d2, y2, total_days);
  }
  else
  {
    printf("Invalid\n");
  }
```

```
    return 0;
}
```

Output:

Enter date1 (MM/DD/YYYY): 05/09/2020

Enter date2 (MM/DD/YYYY): 10/12/2020

The number of days between 5/9/2020 and 10/12/2020 is: 156 days

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>
int main()
{
    long long file_size,time_in_sec;
    long long transmission_rate=960;
    long long days,hours,minutes,seconds;
    printf("Enter the file size in bytes:");
    scanf("%lld",&file_size);
    time_in_sec=file_size/transmission_rate;
    days=time_in_sec/(24*3600);
    hours=(time_in_sec%(24*3600))/3600;
    minutes=(time_in_sec%3600)/60;
    seconds=time_in_sec%60;
    printf("Time required to send the file is %lld days,%lld hours,%lld minutes,%lld seconds\n",days,hours,minutes,seconds);
    return 0;
}
```

Output:

Enter the file size in bytes:456783964

Time required to send the file is 5 days,12 hours,10 minutes,16 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>
int main()
{
   float amount,total;
   printf("Enter the amount:");
   scanf("%f",&amount);
   total=amount*1.08;
   total=(int)(total*100+0.5)/100;
   printf("Total amount after adding 8 percent sales tax is:%.2f\n",total);
   return 0;
}
```

Output:

```
Enter the amount:100
Total amount after adding 8 percent sales tax is:108.00
```

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

```
#include<stdio.h>
int main()
{
   int n,i,is_prime=1;
   printf("Enter a number:");
   scanf("%d",&n);
   if(n<=1)
   {
      is_prime=0;
   }
   else{
      for(i=2;i*i<=n;i++)
```

```
        {
            if(n%i==0)
            {
                is_prime=0;
                break;
            }
        }
    }
    if(is_prime)
    {
        printf("%d is a prime number \n",n);
    }
    else{
        printf("%d is not a prime number \n",n);
    }
    return 0;
}
```

Output:

Enter a number:12

12 is not a prime number

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,num,positive=0,negative=0;
    printf("Enter the numbers:");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
```

```
        printf("Enter number  %d:",i+1);
        scanf("%d",&num);
        if(num>0)
        {
           positive++;
        }
        else if(num<0)
        {
           negative++;
        }
    }
    printf("Total count of positive numbers is %d\n",positive);
    printf("Total count of negative numbers is %d\n",negative);
    return 0;
}
```

Output:

```
Enter the numbers:5
Enter number  1:-1
Enter number  2:4
Enter number  3:-6
Enter number  4:9
Enter number  5:-3
Total count of positive numbers is 2
Total count of negative numbers is 3
```

```
//1.C program to find the HCFof given numbers using recursion
#include<stdio.h>
int hcf(int ,int);
int main()
{
```

```
    int num1,num2;
    printf("Enter number 1:");
    scanf("%d",&num1);
    printf("Enter number 2:");
    scanf("%d",&num2);
    printf("HCF(Highest Common Factor) is %d",hcf(num1,num2));
    return 0;
}
int hcf(int a,int b)
{
    if(b==0)
    {
        return a;
    }
    else
    {
        return hcf(b,a%b);
    }
}
```

Output:

Enter number 1:40

Enter number 2:50

HCF(Highest Common Factor) is 10

```
//2.C program to find the lcm of given numbers using recursion
#include<stdio.h>
int hcf(int,int);
int lcm(int,int);
int main()
{
```

```
    int num1,num2;
    printf("Enter number 1:");
    scanf("%d",&num1);
    printf("Enter number 2:");
    scanf("%d",&num2);
    printf("LCM(Least common multiple) is %d",lcm(num1,num2));
}
int hcf(int a,int b)
{
    if(b==0)
    {
        return a;
    }
    else
    {
        return hcf(b,b%a);
    }
}
int lcm(int a,int b)
{
    return (a*b)/hcf(a,b);
}
```

Output:

```
Enter number 1:20
Enter number 2:15
LCM(Least common multiple) is 20
```

```
//3.C program to find GCD of given numbers using recursion
#include<stdio.h>
int gcd(int,int);
int main()
{
   int num1,num2;
   printf("Enter number1:");
   scanf("%d",&num1);
   printf("Enter number2:");
   scanf("%d",&num2);
   printf("GCD(Greatest Common Divisor) is %d",gcd(num1,num2));
}
int gcd(int a,int b)
{
   if(b==0)
   {
      return a;
   }
   else
   {
      return gcd(b,b%a);
   }
}
```

Output:

```
Enter number1:4
Enter number2:5
GCD(Greatest Common Divisor) is 1
```

```
//4.C program to convert decimal number to binary using recursion
#include<stdio.h>
void printBinary(int );
int main()
{
   int num;
   printf("Enter the number:");
   scanf("%d",&num);
   if(num==0)
   {
      printf("Binary: 0\n");
   }
   else
   {
      printf("Binary:");
      printBinary(num);
   }
   return 0;
}
void printBinary(int n)
{
   if(n==0)
   {
      return;
   }
   printBinary(n/2);
   printf("%d",n%2);
}
```

Output:

```
Enter the number:5
Binary:101
```

```
//C program to convert a binary number to gray code
#include<stdio.h>
int binaryToGray(int);
int main()
{
   int binary,gray;
   printf("Enter a binary number:");
   scanf("%d",&binary);
   gray=binaryToGray(binary);
   printf("Gray code is %d \n",gray);
   return 0;
}
int binaryToGray(int binary)
{
   return binary^(binary>>1);
}
```

Output:

```
Enter a binary number:10
Gray code is 15
```

```
//C program to convert a binary number to gray code using recursion
#include<stdio.h>
int binaryToGray(int);
int main()
{
   int binary,gray;
   printf("Enter a binary number:");
   scanf("%d",&binary);
   gray=binaryToGray(binary);
```

```
    printf("Gray code is %d \n",gray);
    return 0;
}
int binaryToGray(int binary)
{
    if(binary==0)
    {
        return 0;
    }
    return binary^(binary>>1);
}
```

Output:

Enter a binary number:45

Gray code is 59

```
//c program to print 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;
}
```

Output:

```
Enter the number of rows:5
**********
****    ****
***       ***
**          **
*             *
```

```
/*C program to find the sum of natural numbers/factorial of number of all natural numbers from 1 to numbers
series:1/1!+2/2!+3/3!+4/4!+....N/N!
*/
```

```
#include<stdio.h>
int main()
{
   int n,i=1;
   float sum=0;
   int fact;
   printf("Enter the value N:");
   scanf("%d",&n);
   while(i<=n)
   {
      fact=1;
      int j=1;
      while(j<=i)
      {
         fact=fact*j;
         j++;
      }
      sum+=(float)i/fact;
      i++;
   }
   printf("Factorial is %.2f\n",sum);
   return 0;
}
```

Output:

```
Enter the value N:5
Factorial is 2.71
```

```
/*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>
int main()
{
   int n,num;
   float sum=0;
   int square,cube;
   printf("Enter the number of terms:");
   scanf("%d",&n);
   for(int i=1;i<=n;i++)
   {
      num=2*i-1;
      square=num*num;
      cube=num*num*num;
      sum+=(float)square/cube;
   }
   printf("Sum of the series is: %.2f\n",sum);
   return 0;
}
```

Output:

```
Enter the number of terms:10
Sum of the series is: 2.13
```

```
//C program to replace all even numbers by 0 and odd numbers by 1 in one dimensional array
#include<stdio.h>
int main()
{
   int n;
   printf("Enter the size of array:");
   scanf("%d",&n);
   int arr[n];
```

```
    printf("Enter the 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("Array is:");
    for(int i=0;i<n;i++)
    {
        printf("%d",arr[i]);
    }
    return 0;
}
```

Output:

Enter the size of array:5

Enter the elements:1 2 3 4 5

Array is:10101

```c
//C program to read a matrix and print diagonals
#include<stdio.h>
int main()
{
   int m,n;
   printf("Enter the number of rows:");
   scanf("%d",&m);
   printf("Enter the number of columns:");
   scanf("%d",&n);
   int matrix[m][n];
   printf("Enter the elements:");
   for(int i=0;i<m;i++)
   {
      for(int j=0;j<n;j++)
      {
         scanf("%d",&matrix[i][j]);
      }
   }
   printf("Entered matrix:\n");
   for(int i=0;i<m;i++)
   {
      for(int j=0;j<n;j++)
      {
         printf("%d",matrix[i][j]);
      }
      printf("\n");
   }
   printf("First diagonal:");
   for(int i=0;i<m;i++)
   {
      printf("%d",matrix[i][i]);
```

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

Output:

```
Enter the number of rows:3
Enter the number of columns:3
Enter the elements:
1 2 3
4 5 6
7 8 9
Entered matrix:
123
456
789
First diagonal:159Second diagonal:357
```

```
//C program to print the upper triangular portion of a 3x3 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("\nUpper Triangular 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;
}



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

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

    free(arr);
    return 0;
}
```

Output:

Enter the elements of the 3x3 matrix:

1 2 3

4 5 6

7 8 9

Upper Triangular Matrix:

1 2 3

 5 6

  9

```
//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 maximum number of characters: ");
   scanf("%d", &size);

   text = (char *)malloc(size * sizeof(char));

   if (text == NULL) {
      printf("Memory allocation failed!\n");
      return 1;
   }
   getchar();
   printf("Enter text: ");
```

```
    scanf("[^\n]", text);

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

    free(text);
    return 0;
}
```

Output:

Enter the maximum number of characters: 20

Enter text: hello goodmorning

You entered: hello goodmorning

//C program to read 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: ");
    scanf("%d", &n);

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

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

    printf("Enter the elements: \n");
```

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

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

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

    free(arr);
    return 0;
}
```

Output:

```
Enter the number of elements: 6
Enter the elements:
1 2 3 4 5 6
Array elements are:
1 2 3 4 5 6
Sum of all elements: 21
```