

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>

float convertMilesToKilometers(float a);
float convertGallonsToLiters(float a);
float convertPoundsToKilograms(float a);
float convertInchesToCentimeters(float a);
float convertFeetToMeters(float a);

int main() {
    int op;
    float a, result;

    while (1) {
        printf("\nSelect a 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("5. Feet to Meters\n");
        printf("6. Exit\n");
        printf("Enter your choice (1-6): ");
        scanf("%d", &op);

        switch (op) {
            case 1:
                printf("\nEnter value in Miles::");
                scanf("%f", &a);
                result = convertMilesToKilometers(a);
                printf("\n%.2f Kilometers\n", result);
                break;
            case 2:
                printf("\nEnter value in Gallons::");
                scanf("%f", &a);
                result = convertGallonsToLiters(a);
                printf("\n%.2f Liters\n", result);
                break;
            case 3:
                printf("\nEnter value in Pounds::");
                scanf("%f", &a);
                result = convertPoundsToKilograms(a);
                printf("\n%.2f Kilograms\n", result);
                break;
```

```

        case 4:
            printf("\nEnter value in Inches::");
            scanf("%f",a);
            result = convertInchesToCentimeters(a);
            printf("\n%.2f Centimeters\n", result);
            break;
        case 5:
            printf("\nEnter value in Feet::");
            scanf("%f",a);
            result = convertFeetToMeters(a);
            printf("\n%.2f Meters\n", result);
            break;
        case 6:
            printf("\nExiting...");
            return 0;
            break;
        default:
            printf("Invalid op\n");
    }
}

return 0;
}

float convertMilesToKilometers(float a) {
    return a * 1.60934;
}

float convertGallonsToLiters(float a) {
    return a * 3.78541;
}

float convertPoundsToKilograms(float a) {
    return a * 0.453592;
}

float convertInchesToCentimeters(float a) {
    return a * 2.54;
}

float convertFeetToMeters(float a) {
    return a * 0.3048;
}

```

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 calculateDays(int day, int month, int year);
int main() {
    int day, month, year;
    int fday, fmonth, fyear;
    int daycount;

    printf("\nEnter starting Date(DD MM YYYY)::");
    scanf("%d %d %d", &day, &month, &year);

    printf("\nEnter ending date(DD MM YYYY)::");
    scanf("%d %d %d", &fday, &fmonth, &fyear);

    int startDays = calculateDays(day, month, year);
    int endDays = calculateDays(fday, fmonth, fyear);
    daycount = endDays - startDays;

    printf("\nDays between :: %d\n", daycount);

    return 0;
}
int calculateDays(int day, int month, int year) {
    int daysInMonth[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30,
31};
    int totalDays = 0;
    for (int i = 1; i < year; i++) {
        totalDays += (i % 4 == 0 && (i % 100 != 0 || i % 400 == 0)) ?
366 : 365;
    }
    for (int i = 0; i < month - 1; i++) {
        totalDays += daysInMonth[i];
    }
    if (month > 2 && (year % 4 == 0 && (year % 100 != 0 || year % 400
== 0))) {
        totalDays += 1;
    }
}
```

```

    totalDays += 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 program on a 400MB (419,430,400 -byte) file. Use appropriate units. (A 400MB file takes days.)

```

#include <stdio.h>

void time(int size, int rate);

int main() {

    int size;
    printf("\nEnter File Size::");
    scanf("%d",&size);
    time(size, 960);

    return 0;
}

void time(int size, int rate) {

    int time_seconds = size / rate;
    int days = time_seconds / (24 * 3600);
    time_seconds %= (24 * 3600);
    int hours = time_seconds / 3600;
    time_seconds %= 3600;
    int minutes = time_seconds / 60;
    int seconds = time_seconds % 60;
    printf("Time required to send the file: %d days, %d hours, %d
minutes, %d seconds\n", days, 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>

float Tax(float, float );

int main() {

    float amount, total;

```

```

printf("Enter the amount: ");
scanf("%f", &amount);
total = Tax(amount, 8.0);
printf("\nAfter adding %.1f%% sales tax is: $%.2f\n", 8.0, total);

return 0;
}
float Tax(float amount, float taxRate) {
    float total = amount + (amount * taxRate / 100);
    return roundf(total * 100) / 100;
}

```

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

```

#include<stdio.h>
int main(){
    int n, i, isPrime = 1;
    printf("Enter a Number::");
    scanf("%d", &n);

    if (n <= 1) {
        printf("NOT PRIME\n");
        return 0;
    }

    if (n == 2) {
        printf("PRIME\n");
        return 0;
    }

    for (i = 2; i <= n / 2; i++) {
        if (n % i == 0) {
            isPrime = 0;
            break;
        }
    }

    if (isPrime) {
        printf("PRIME\n");
    } else {
        printf("NOT PRIME\n");
    }
}

```

```
    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 n, i, countp = 0, countn = 0;
    printf("Enter the no:of elements in array: ");
    scanf("%d", &n);
    int arr[n];
    printf("\nEnter %d numbers (positive and negative) into array:\n",
n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    for (i = 0; i < n; i++) {
        if (arr[i] > 0) {
            countp++;
        } else if (arr[i] < 0) {
            countn++;
        }
    }
    printf("\nPositive Numbers::%d Negative
Numbers::%d",countp,countn);

    return 0;
}
```

1.HCF using Recursion

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

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

```

        printf("Enter the second number: ");
        scanf("%d", &num2);
        printf("The HCF of %d and %d is %d.\n", num1, num2, hcf(num1,
num2));
        return 0;
    }
int hcf(int a, int b) {
    if (b == 0)
        return a;
    return hcf(b, a % b);
}

```

2.LCM using Recursion

```

#include <stdio.h>
int hcf(int a, int b);
int lcm(int a, int b);
int main() {
    int num1, num2;
    printf("Enter the first number: ");
    scanf("%d", &num1);
    printf("Enter the second number: ");
    scanf("%d", &num2);
    printf("The LCM of %d and %d is %d.\n", num1, num2, lcm(num1,
num2));
    return 0;
}
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);
}

```

3.GCD using Recursion

```

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

int main() {

```

```

    int num1, num2;
    printf("Enter the first number: ");
    scanf("%d", &num1);
    printf("Enter the second number: ");
    scanf("%d", &num2);
    printf("The GCD of %d and %d is %d.\n", num1, num2, gcd(num1,
num2));
    return 0;
}
int gcd(int a, int b) {
    if (b == 0)
        return a;
    return gcd(b, a % b);
}

```

4.Decimal to Binary using Recursion

```

#include <stdio.h>
void decimalToBinary(int n);
int main() {
    int num;
    printf("Enter a decimal number: ");
    scanf("%d", &num);
    if (num == 0) {
        printf("Binary: 0\n");
    } else {
        printf("Binary: ");
        decimalToBinary(num);
        printf("\n");
    }

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

```


5.Binary to GrayCode

```
#include <stdio.h>
int binaryToGray(int binary);
int main() {
    int binary;
    printf("Enter a binary number (as an integer): ");
    scanf("%d", &binary);
    printf("Gray code equivalent: %d\n", binaryToGray(binary));
    return 0;
}
int binaryToGray(int binary) {
    return binary ^ (binary >> 1);
}
```

6.Binary to Gray code using recursion

```
#include <stdio.h>
int binaryToGray(int binary);
int main() {
    int binary;
    printf("Enter a binary number (as an integer): ");
    scanf("%d", &binary);
    int gray = binaryToGray(binary);
    printf("Gray code representation: %d\n", gray);

    return 0;
}
int binaryToGray(int binary) {
    if (binary < 2) {
        return binary;
    }
    return binary ^ (binary >> 1);
}
```

7.print pattern

```

*****
****  ****
***   ***
**    **
*     *

```

```

#include <stdio.h>
int main() {
    int i, j;
    int n ;
    printf("\nEnter no:of rows::");
    scanf("%d",&n);

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

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

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

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

        printf("\n");
    }

    return 0;
}

```

8.Sum of the series $1/1!+2/2!+.....N/N!$

```

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

```

```

float sum=0.0;
int f;
printf("\nEnter value of n::");
scanf("%d",&n);
for(int i=1;i<=n;i++){
    f=fact(i);
    sum+=(float)i/f;
}
printf("\nThe Sum of the series is ::%f",sum);
}
int fact(int a){
    int fact=1;
    for(int i=1;i<=a;i++){
        fact=fact*i;
    }
    return fact;
}

```

9.Sum of the series $1^2/1^3 + 3^2/3^3 + 5^2/5^3 + \dots$

```

#include<stdio.h>
int main(){
    int n,count=0;
    float sum=0.0;
    printf("\nEnter value of n::");
    scanf("%d",&n);
    int i=1;
    while(count!=n){
        sum+=(float)(i*i)/(float)(i*i*i);
        i+=2;
        count++;
    }
    printf("\nSum of the series is::%f",sum);
}

```

10.Change array values even by 0 and odd by 1.

```

#include<stdio.h>
int main(){
    int n;
    printf("Enter size of array::");
    scanf("%d",&n);
    int a[n];
    printf("\nEnter %d values to array::",n);
}

```

```

for(int i=0;i<n;i++){
    scanf("%d",&a[i]);
}
for(int i=0;i<n;i++){
    if(a[i]%2==0){
        a[i]=0;
    }
    else{
        a[i]=1;
    }
}
printf("\nAfter Changing::\n");
for(int i=0;i<n;i++){
    printf("%d ",a[i]);
}
}

```

11.Read matrix and print diagonal

```

#include<stdio.h>
int main(){
    int row,col;
    printf("\nEnter row and coloumn for matrix(try square matrix for Diagonals)::");
    scanf("%d %d",&row,&col);
    if(row!=col){
        printf("\nNo proper diagonals");
        return 1;
    }
    int mat[row][col];
    printf("\nEnter values into the %dx%d matrix::\n",row,col);
    for(int i=0;i<row;i++){
        for(int j=0;j<col;j++){
            printf("a[%d][%d]=\t",i,j);
            scanf("%d",&mat[i][j]);
        }
    }
    printf("\nDiagonal Elements are::");
    for(int i=0;i<row;i++){
        printf("%d\t",mat[i][i]);
    }
}

```

12.Upper Triangle of 3x3 matrix.

```
#include<stdio.h>
int main(){
    int mat[3][3];
    printf("\nEnter Values into the 3x3 matrix::");
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            scanf("%d",&mat[i][j]);
        }
    }
    printf("\nUpper Triangle of matrix is::\n");
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            if(j>=i){
                printf("%d\t",mat[i][j]);
            }
            else{
                printf("\t");
            }
        }
        printf("\n");
    }
}
```

13.Input and print text using dynamic memory allocation

```
#include <stdlib.h>

int main() {
    int size;
    printf("Enter the length of the string: ");
    scanf("%d", &size);
    char *str = (char *)malloc((size + 1) * sizeof(char));

    if (str == NULL) {
        printf("\nAllocation Error!\n");
        return 1;
    }
    printf("\nEnter the string: ");
    scanf("%s", str);
    printf("\nString is :: %s\n",str);
    free(str);
}
```

```
    return 0;
}
```

14. Read an array and print the sum using dynamic allocation

```
#include<stdio.h>
#include<stdlib.h>
int main(){
    int n;
    printf("\nEnter size of array::");
    scanf("%d",&n);
    int *arr=(int*)malloc(n*sizeof(int));
    if(arr==NULL){
        printf("\nAllocation Error");
    }
    printf("\nEnter %d elements into the array::",n);
    for(int i=0;i<n;i++){
        scanf("%d",arr+i);
    }
    int sum=0;
    for(int i=0;i<n;i++){
        sum+=*(arr+i);
    }
    printf("\nThe SUM is ::%d",sum);
    for(int i=0;i<n;i++){
        free(arr);
    }

    return 0;
}
```