

**1)Exercise 1: Write a c 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>
```

```
int main() {
```

```
    int choice;
```

```
    double value, convertedValue;
```

```
    printf("Unit Conversion Program\n");
```

```
    printf("1. Miles to Kilometers\n");
```

```
    printf("2. Gallons to Liters\n");
```

```
    printf("3. Pounds to Kilograms\n");
```

```
    printf("Enter your choice (1-3): ");
```

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

```
    switch (choice) {
```

```
        case 1:
```

```
            printf("Enter value in miles: ");
```

```
            scanf("%lf", &value);
```

```
            convertedValue = value * 1.60934;
```

```
            printf("%.2lf miles is equal to %.2lf kilometers.\n", value, convertedValue);
```

```
            break;
```

```
        case 2:
```

```
            printf("Enter value in gallons: ");
```

```
            scanf("%lf", &value);
```

```
            convertedValue = value * 3.78541;
```

```
            printf("%.2lf gallons is equal to %.2lf liters.\n", value, convertedValue);
```

```
            break;
```

```
        case 3:
```

```
            printf("Enter value in pounds: ");
```

```
            scanf("%lf", &value);
```

```
            convertedValue = value * 0.453592;
```

```
            printf("%.2lf pounds is equal to %.2lf kilograms.\n", value, convertedValue);
```

```
            break;
```

```
        default:
```

```
            printf("Invalid choice! Please select a valid option.\n");
```

```
            break;
```

```
    }
```

```
    return 0;
```

```
}
```

**2) 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);

int countDays(int day, int month, int year);

int daysBetween(int day1, int month1, int year1, int day2, int month2, int year2);

int main() {

    int day1, month1, year1;

    int day2, month2, year2;

    int days;

    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);

    days = daysBetween (day1, month1, year1, day2, month2, year2);

    printf("Number of days: %d", days);

    return 0;

    int isLeapYear(int year) {

        return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

    }

    int countDays(int day, int month, int year) (

        static int monthDays [12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

        int days = year * 365 + day;

        for (int i = 0; i < month - 1; i++) {

            days += monthDays[i];
```

```

}

days += year / 4 - year / 100 + year / 400;

if (month > 2 && isLeapYear (year)) {

days++;

}

return days;

}

Int daysBetween(int day1, int month1, int year1, int day2, int month2, int year2)
{
return countDays (day2, month2, year2) - countDays (day1, month1, year1);
}

```

o/p

Enter the first date (dd mm yyyy): 23 02 2001

Enter the second date (dd mm yyyy): 18 01 2006

Number of days: 1790

**3) 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 transmission_time(long file_size, int transmission_rate);

int main(){

long file_size=419430400;

int transmission_rate=960;

```

```

transmission_time(file_size, transmission_rate);

return 0;
}

void transmission_time(long file_size, int transmission_rate) {

double time_in_second=file_size/transmission_rate;

double time_in_minutes=time_in_second/60;

double time_in_hours = time_in_minutes/60;

int time_in_days=time_in_hours/24;

printf("Time Required for a file size of %ld bytes is \n", file_size);

printf("Seconds: %.2f\n", time_in_second);

printf("Minutes %.2f\n",time_in_minutes);

printf("Hours: %.2f\n", time_in_hours);

printf("Days: %d\n", time_in_days);
}

```

o/p

Time Required for a file size of 419430400 bytes is :

Seconds 436906.00

Minutes: 7281.77

Hours: 121.36

Days: 5

**4) 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() {
    double amount, taxRate = 0.08, total;

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

    total = amount + (amount * taxRate);
    total = round(total * 100) / 100;

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

    return 0;
}

```

o/p

Enter the amount in dollars: 100.00

The total amount after adding 8% sales tax is: \$108.00

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

```

#include <stdio.h>

int isPrime(int number);

int main() {
    int number;

    printf("Enter a number: ");

    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 number) {

if (number <= 1) {

return 0;

}

for (int i = 2; i <= number / 2; i++) {

if (number % i == 0) {

return 0;

}

}

return 1;
}

```

Enter a number: 11

11 is a prime number.

**6) 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 positiveCount = 0, negativeCount = 0;

    printf("Enter numbers (enter 0 to stop): \n");

    while (1) {

        scanf("%d", &num);

```

```

    if (num == 0) {
        break;
    }

    if (num > 0) {
        positiveCount++;
    } else if (num < 0) {
        negativeCount++;
    }
}

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

return 0;
}

```

o/p

Enter numbers (enter 0 to stop):

5  
-3  
8  
-2  
0

Number of positive values: 2  
Number of negative values: 2

## 7) C program to find the HCF of given numbers using recursion

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

```
int hcf(int n,int m);
```

```
int main(){
```

```
int a,b;
```

```
printf("Enter first number: ");
```

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

```
printf("Enter second number: ");
```

```

scanf("%d",&b);

printf("HCF of %d and %d = %d", a, b, hcf(a,b));

return 0;
}

int hcf(int n,int m) {

if(m==0){

return n;

}

else{

return hcf(m,n%m);

}
}

```

o/p

Enter first number :10

Enter second number :20

HCF of 10 and 20 = 10

### 8) C program to find the LCM of given numbers using recursion

```

#include<stdio.h>

int hcf(int n,int m);

int lcm(int n,int m);

int main(){

int a,b;

printf("Enter first number: ");

scanf("%d",&a);

printf("Enter second number: ");

```



```

scanf("%d",&b);

printf("LCM of %d and %d = %d",a,b,lcm(a,b));

return 0;

}

int hcf(int n, int m){

if(m==0){

return n;

}

else{

return hcf(m,n%m);

}

}

int lcm(int n, int m){

int lcm_of_numbers=(n*m)/hcf(n,m);

return lcm_of_numbers;

}

```

o/p

Enter first number :10

Enter second number :20

LCM of 10 and 20 = 20

### 9) C program to find the GCD of given numbers 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);

printf("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);

}

```

o/p

Enter two numbers: 4

8

GCD of 4 and 8 is 4

## 10) C program to convert a Decimal number to Binary using Recursion

```

#include <stdio.h>

void decimal_to_binary(int num);

int main() {

int n;

printf("Enter a Number: ");

scanf("%d", &n);

```

```

if (n == 0) {

printf("0\n");

} else {

printf("%d to Binary = ", n);

decimal_to_binary(n);

printf("\n");
}

return 0;
}

void decimal_to_binary(int num) {

if (num > 0) {

decimal_to_binary (num / 2);

printf("%d", num % 2);

}
}

```

o/p

Enter a Number: 5  
5 to Binary 101

### 11) C program to convert Binary Number to Gray Code

```

#include <stdio.h>

int binaryToGray (int num);

int main() {

int binary;

printf("Enter a binary number: ");

scanf("%d", &binary);

```

```

int gray = binaryToGray (binary);

printf("Binary %d to Gray Code = %d\n", binary, gray);

return 0;
}

int binaryToGray (int num) {

return num ^ (num >> 1);
}

```

o/p

Enter a binary number: 8

Binary 8 to Gray Code = 12

## 12) C program to convert Binary Number to Gray Code using Recursion

```

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

int bintogray(int bin) {
    if (bin == 0 || bin == 1) {
        return bin;
    }

    int last_digit = bin % 10;

    int remaining_bin = bin / 10;

    int gray = bintogray(remaining_bin);

    return (last_digit ^ (gray % 10)) * pow(10, (int)log10(gray) + 1) + gray / 10;
}

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;
}

```

o/p

Enter a binary number: 100  
The Gray code of 100 is 1

13)

7. C program to print following Pyramid:

```

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

```

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

```
int main() {
    int rows = 5; // Number of rows in the pattern
```

```
    // Loop through each row
    for (int i = 1; i <= rows; i++) {
        // Print the first set of stars
        for (int j = 1; j <= rows - i + 1; j++) {
            printf("*");
        }
    }

```

```
    // Print spaces in between
    for (int j = 1; j <= 2 * (i - 1); j++) {
        printf(" ");
    }

```

```
    // Print the second set of stars
    for (int j = 1; j <= rows - i + 1; j++) {
        printf("*");
    }

```

```
    // Move to the next line

```

```

        printf("\n");
    }

    return 0;
}

```

**14) C program to find the sum of Natural Number/Factorial of Number of all natural number from 1 to N.**

**Series:  $1/1! + 2/2! + 3/3! + 4/4! + \dots N/N!$**

```

#include <stdio.h>

int main() {

    int N;

    printf("Enter the value of N: ");

    scanf("%d", &N);

    double sum = 0.0;

    double factorial = 1.0;

    for (int i = 1 ; i < N; i++) {

        factorial *= i;

        sum += (double)i / factorial;

    }

    printf("The sum of the series is: %.2f\n", sum);

    return 0;
}

```

15)

9. C program to find sum of following series:

$1 + 3^2/3^3 + 5^2/5^3 + 7^2/7^3 + \dots$  till N terms

```

#include <stdio.h>

```

```

#include <math.h>

int main(){

int limits;

printf("Enter the limits:");

scanf("%d",&limits);

float sum = 1;

printf("Series is 1 + ");

for(int i=2;i<=limits;i++){

if(i%2 != 0){

float term = pow(i, 2) / pow(i, 3);

sum += term;

printf("%d^2 / %d^3 +",i,i);

}

}

printf("Sum of series = %.2f", sum);

return 0;
}

```

**16) 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 Number of elements needed in an array");

scanf("%d",&n);

```

```

int arr[n];

printf("Enter %d Elements\n",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;

}

else{

arr[i]=1;

}

}

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

printf("arr[%d] = %d \n",i,arr[i]);

}

return 0;

}

```

### 17) C program to read a matrix and print diagonals

```

#include<stdio.h>

int main(){

int m,n;

printf("Enter number of rows (n) and columns(n) in a matrix:\n");

scanf("%d %d", &m,&n);

```



```

int arr[m][n];

printf("Enter Elements for Matrix\n");

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

for(int j=0;j<m;j++){

printf("arr[%d][%d] = \t",i,j);

scanf("%d", &arr[i][j]);

}

}

printf("Diagonal Elements are: \n");

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

for(int j=0;j<m;j++){

if(i==j){

printf("arr[%d] [%d]=%d\n", i,j,arr[i][j]);

}

}

}
}

```

o/p

Enter number of rows (n) and columns(n) in a matrix:

2 2

Enter Elements for Matrix

arr[0][0] = 1

arr[0][1] = 2

arr[1][0] = 3

arr[1][1] = 4

Diagonal Elements are:

arr[0][0]=1

arr[1][1]=4

### 18) 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("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;
}
```

Enter the elements of the 3x3 matrix:

1  
2  
3  
4  
5  
6  
7  
8  
9

Upper Triangular Portion of the Matrix:

1 2 3  
5 6  
9