**Assignment 6**

// Function Declarations (Type 3 - void return, with parameters using pointers)

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

// Function Declarations (Type 3 - void return, with parameters using pointers)

// Assignment 1

void assignment\_1();

void fahrenheit\_to\_celsius(int \*celsius);

void find\_area\_and\_perimeter(int \*choice, int \*length, int \*width, int \*radius);

void input\_three\_digit\_number(int \*no);

void check\_even\_or\_odd(int \*no);

void calculate\_total\_salary(double \*basic);

void check\_marriage\_eligibility(int \*male\_age, int \*female\_age);

// Assignment 2

void assignment\_2();

void item\_price\_with\_discount(int \*price);

void find\_greatest\_of\_three(int \*a, int \*b, int \*c);

void calculator\_with\_operator(int \*num1, int \*num2, char \*op);

void display\_menu(int \*choice, int \*number, double \*basic);

void check\_student\_discount(int \*price, int \*isStudent);

// Assignment 3

void assignment\_3();

void print\_1\_to\_10();

void print\_table\_for\_number(int \*num);

void sum\_in\_range(int \*start, int \*end);

void check\_prime\_number(int \*num);

void check\_armstrong\_number(int \*num);

void check\_perfect\_number(int \*num);

void factorial\_of\_number(int \*num);

void check\_strong\_number(int \*num);

void check\_palindrome\_number(int \*num);

void add\_first\_and\_last\_digits(int \*num);

// Assignment 4

void assignment\_4();

void prime\_numbers\_in\_range(int \*start, int \*end);

void armstrong\_numbers\_in\_range(int \*start, int \*end);

void perfect\_numbers\_in\_range(int \*start, int \*end);

void strong\_numbers\_in\_range(int \*start, int \*end);

void generate\_fibonacci\_series(int \*n);

int main()

{

    int choice;

    while (1)

    {

        printf("\n================ MAIN MENU ================\n");

        printf("1) Assignment 1\n2) Assignment 2\n3) Assignment 3\n4) Assignment 4\n0) Exit\n");

        printf("Enter Your Assignment No: ");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            assignment\_1();

            break;

        case 2:

            assignment\_2();

            break;

        case 3:

            assignment\_3();

            break;

        case 4:

            assignment\_4();

            break;

        case 0:

            exit(0);

        default:

            printf("Invalid Choice. Try again.\n");

        }

    }

    return 0;

}

void assignment\_1()

{

    int ch;

    while (1)

    {

        printf("\n---------------- Assignment 1 ----------------\n");

        printf("1) Celsius to Fahrenheit\n");

        printf("2) Area and Perimeter\n");

        printf("3) Input 3-digit number\n");

        printf("4) Even or Odd\n");

        printf("5) Total Salary Calculation\n");

        printf("6) Marriage Eligibility\n");

        printf("0) Back to Main Menu\n");

        printf("Enter your choice: ");

        scanf("%d", &ch);

        int num, choice, length, width, radius;

        double basic;

        int male\_age, female\_age;

        switch (ch)

        {

        case 1:

            printf("Enter temperature in Celsius: ");

            scanf("%d", &num);

            fahrenheit\_to\_celsius(&num);

            break;

        case 2:

            printf("1) Rectangle\n2) Circle\nEnter choice: ");

            scanf("%d", &choice);

            if (choice == 1)

            {

                printf("Enter length: ");

                scanf("%d", &length);

                printf("Enter width: ");

                scanf("%d", &width);

            }

            else if (choice == 2)

            {

                printf("Enter radius: ");

                scanf("%d", &radius);

            }

            find\_area\_and\_perimeter(&choice, &length, &width, &radius);

            break;

        case 3:

            printf("Enter a 3-digit number: ");

            scanf("%d", &num);

            input\_three\_digit\_number(&num);

            break;

        case 4:

            printf("Enter a number: ");

            scanf("%d", &num);

            check\_even\_or\_odd(&num);

            break;

        case 5:

            printf("Enter Basic Salary: ");

            scanf("%lf", &basic);

            calculate\_total\_salary(&basic);

            break;

        case 6:

            printf("Enter Male Age: ");

            scanf("%d", &male\_age);

            printf("Enter Female Age: ");

            scanf("%d", &female\_age);

            check\_marriage\_eligibility(&male\_age, &female\_age);

            break;

        case 0:

            return;

        default:

            printf("Invalid choice. Try again.\n");

        }

    }

}

// ================= Assignment 1: Functions with Pointers =================

void fahrenheit\_to\_celsius(int \*celsius)

{

    int fahrenheit = (\*celsius \* 9 / 5) + 32;

    printf("Temperature in Fahrenheit: %d°F\n", fahrenheit);

}

void find\_area\_and\_perimeter(int \*choice, int \*length, int \*width, int \*radius)

{

    float area, perimeter;

    const float pi = 3.14f;

    if (\*choice == 1)

    {

        area = (\*length) \* (\*width);

        perimeter = 2 \* ((\*length) + (\*width));

        printf("Rectangle Area: %.2f\nRectangle Perimeter: %.2f\n", area, perimeter);

    }

    else if (\*choice == 2)

    {

        area = pi \* (\*radius) \* (\*radius);

        perimeter = 2 \* pi \* (\*radius);

        printf("Circle Area: %.2f\nCircle Perimeter: %.2f\n", area, perimeter);

    }

    else

    {

        printf("Invalid shape choice.\n");

    }

}

void input\_three\_digit\_number(int \*no)

{

    if (\*no >= 100 && \*no <= 999)

    {

        int n1 = \*no / 100;

        int n2 = (\*no / 10) % 10;

        int n3 = \*no % 10;

        int sum = n1 + n2 + n3;

        printf("Digits: %d, %d, %d\n", n1, n2, n3);

        printf("Sum of digits: %d\n", sum);

        printf("Reverse of number: %d%d%d\n", n3, n2, n1);

    }

    else

    {

        printf("Invalid input. Not a 3-digit number.\n");

    }

}

void check\_even\_or\_odd(int \*no)

{

    if (\*no % 2 == 0)

        printf("%d is Even\n", \*no);

    else

        printf("%d is Odd\n", \*no);

}

void calculate\_total\_salary(double \*basic)

{

    double da, ta, hra, totalSalary;

    if (\*basic <= 5000)

    {

        da = \*basic \* 0.10;

        ta = \*basic \* 0.20;

        hra = \*basic \* 0.25;

    }

    else

    {

        da = \*basic \* 0.15;

        ta = \*basic \* 0.25;

        hra = \*basic \* 0.30;

    }

    totalSalary = \*basic + da + ta + hra;

    printf("Total Salary = %.2lf\n", totalSalary);

}

void check\_marriage\_eligibility(int \*male\_age, int \*female\_age)

{

    if (\*male\_age >= 21)

        printf("Male is eligible for marriage.\n");

    else

        printf("Male is NOT eligible for marriage.\n");

    if (\*female\_age >= 18)

        printf("Female is eligible for marriage.\n");

    else

        printf("Female is NOT eligible for marriage.\n");

}

#include <stdio.h>

// ================= Assignment 2: Functions with Pointers =================

void assignment\_2()

{

    int ch;

    while (1)

    {

        printf("\n---------------- Assignment 2 ----------------\n");

        printf("1) Item Price with Discount\n");

        printf("2) Greatest of Three Numbers\n");

        printf("3) Calculator with Operator\n");

        printf("4) Display Menu\n");

        printf("5) Check Student Discount\n");

        printf("0) Back to Main Menu\n");

        printf("Enter your choice: ");

        scanf("%d", &ch);

        if (ch == 0)

            return;

        switch (ch)

        {

        case 1:

        {

            int price;

            printf("Enter the price of the item: ");

            scanf("%d", &price);

            item\_price\_with\_discount(&price);

            break;

        }

        case 2:

        {

            int a, b, c;

            printf("Enter three numbers: ");

            scanf("%d %d %d", &a, &b, &c);

            find\_greatest\_of\_three(&a, &b, &c);

            break;

        }

        case 3:

        {

            int num1, num2;

            char op;

            printf("Enter two numbers: ");

            scanf("%d %d", &num1, &num2);

            printf("Enter an operator (+, -, \*, /, %%): ");

            scanf(" %c", &op);

            calculator\_with\_operator(&num1, &num2, &op);

            break;

        }

        case 4:

        {

            int choice, number = 0;

            double basic = 0;

            printf("1. Check Even/Odd\n2. Calculate Basic Salary\nEnter your choice: ");

            scanf("%d", &choice);

            if (choice == 1)

            {

                printf("Enter a number: ");

                scanf("%d", &number);

            }

            else if (choice == 2)

            {

                printf("Enter Basic Salary: ");

                scanf("%lf", &basic);

            }

            display\_menu(&choice, &number, &basic);

            break;

        }

        case 5:

        {

            int price, isStudent;

            printf("Enter the total purchase amount: ");

            scanf("%d", &price);

            printf("Are you a student? (1 = Yes, 2 = No): ");

            scanf("%d", &isStudent);

            check\_student\_discount(&price, &isStudent);

            break;

        }

        default:

            printf("Invalid choice. Try again.\n");

        }

    }

}

void item\_price\_with\_discount(int \*price)

{

    int discountRate, discount, finalPrice;

    if (\*price <= 1000)

        discountRate = 5;

    else if (\*price <= 5000)

        discountRate = 10;

    else

        discountRate = 20;

    discount = (\*price \* discountRate) / 100;

    finalPrice = \*price - discount;

    printf("Discount: %d\n", discount);

    printf("Final Price after discount: %d\n", finalPrice);

}

void find\_greatest\_of\_three(int \*a, int \*b, int \*c)

{

    if (\*a == \*b && \*b == \*c)

    {

        printf("All numbers are equal.\n");

        return;

    }

    if (\*a >= \*b && \*a >= \*c)

        printf("Greatest number: %d\n", \*a);

    else if (\*b >= \*c)

        printf("Greatest number: %d\n", \*b);

    else

        printf("Greatest number: %d\n", \*c);

}

void calculator\_with\_operator(int \*num1, int \*num2, char \*op)

{

    int result;

    switch (\*op)

    {

    case '+':

        result = \*num1 + \*num2;

        printf("Result: %d\n", result);

        break;

    case '-':

        result = \*num1 - \*num2;

        printf("Result: %d\n", result);

        break;

    case '\*':

        result = \*num1 \* \*num2;

        printf("Result: %d\n", result);

        break;

    case '/':

        if (\*num2 != 0)

            printf("Result: %d\n", \*num1 / \*num2);

        else

            printf("Error: Division by zero is not allowed.\n");

        break;

    case '%':

        if (\*num2 != 0)

            printf("Result: %d\n", \*num1 % \*num2);

        else

            printf("Error: Division by zero is not allowed.\n");

        break;

    default:

        printf("Invalid operator!\n");

    }

}

void display\_menu(int \*choice, int \*number, double \*basic)

{

    if (\*choice == 1)

    {

        if (\*number % 2 == 0)

            printf("%d is Even.\n", \*number);

        else

            printf("%d is Odd.\n", \*number);

    }

    else if (\*choice == 2)

    {

        double hra = 0.20 \* (\*basic);

        double da = 0.50 \* (\*basic);

        double gross = \*basic + hra + da;

        printf("Gross Salary: %.2lf\n", gross);

    }

    else

    {

        printf("Invalid choice!\n");

    }

}

void check\_student\_discount(int \*price, int \*isStudent)

{

    double discount = 0.0;

    if (\*isStudent == 1)

    {

        if (\*price > 500)

        {

            discount = 0.20 \* (\*price);

            printf("20%% student discount applied.\n");

        }

        else

        {

            discount = 0.10 \* (\*price);

            printf("10%% student discount applied.\n");

        }

    }

    else if (\*isStudent == 2)

    {

        if (\*price > 600)

        {

            discount = 0.15 \* (\*price);

            printf("15%% non-student discount applied.\n");

        }

        else

        {

            printf("No discount applicable.\n");

        }

    }

    else

    {

        printf("Invalid input for student status.\n");

        return;

    }

    double finalPrice = \*price - discount;

    printf("Final price after discount: %.2lf\n", finalPrice);

}

// ================= Assignment 3: Functions with Pointers =================

void assignment\_3()

{

    int ch, num, start, end;

    while (1)

    {

        printf("\n---------------- Assignment 3 ----------------\n");

        printf("1) Print 1 to 10\n");

        printf("2) Print Table for Number\n");

        printf("3) Sum in Range\n");

        printf("4) Check Prime Number\n");

        printf("5) Check Armstrong Number\n");

        printf("6) Check Perfect Number\n");

        printf("7) Factorial of Number\n");

        printf("8) Check Strong Number\n");

        printf("9) Check Palindrome Number\n");

        printf("10) Add First and Last Digits\n");

        printf("0) Back to Main Menu\n");

        printf("Enter your choice: ");

        scanf("%d", &ch);

        switch (ch)

        {

        case 1:

            print\_1\_to\_10();

            break;

        case 2:

            printf("Enter a number: ");

            scanf("%d", &num);

            print\_table\_for\_number(&num);

            break;

        case 3:

            printf("Enter start and end range: ");

            scanf("%d %d", &start, &end);

            sum\_in\_range(&start, &end);

            break;

        case 4:

            printf("Enter a number: ");

            scanf("%d", &num);

            check\_prime\_number(&num);

            break;

        case 5:

            printf("Enter a number: ");

            scanf("%d", &num);

            check\_armstrong\_number(&num);

            break;

        case 6:

            printf("Enter a number: ");

            scanf("%d", &num);

            check\_perfect\_number(&num);

            break;

        case 7:

            printf("Enter a number: ");

            scanf("%d", &num);

            factorial\_of\_number(&num);

            break;

        case 8:

            printf("Enter a number: ");

            scanf("%d", &num);

            check\_strong\_number(&num);

            break;

        case 9:

            printf("Enter a number: ");

            scanf("%d", &num);

            check\_palindrome\_number(&num);

            break;

        case 10:

            printf("Enter a number: ");

            scanf("%d", &num);

            add\_first\_and\_last\_digits(&num);

            break;

        case 0:

            return;

        default:

            printf("Invalid choice. Try again.\n");

        }

    }

}

void print\_1\_to\_10()

{

    printf("\nQ1. Numbers from 1 to 10:\n");

    for (int i = 1; i <= 10; i++)

    {

        printf("%d ", i);

    }

    printf("\n");

}

void print\_table\_for\_number(int \*num)

{

    printf("\nQ2. Table of %d:\n", \*num);

    for (int i = 1; i <= 10; i++)

    {

        printf("%d x %d = %d\n", \*num, i, (\*num) \* i);

    }

}

void sum\_in\_range(int \*start, int \*end)

{

    int sum = 0;

    for (int i = \*start; i <= \*end; i++)

    {

        sum += i;

    }

    printf("Sum from %d to %d is: %d\n", \*start, \*end, sum);

}

void check\_prime\_number(int \*num)

{

    int isPrime = 1;

    if (\*num <= 1)

        isPrime = 0;

    else

    {

        for (int i = 2; i <= (\*num) / 2; i++)

        {

            if (\*num % i == 0)

            {

                isPrime = 0;

                break;

            }

        }

    }

    printf("%d is %s Prime number.\n", \*num, isPrime ? "a" : "not a");

}

void check\_armstrong\_number(int \*num)

{

    int original = \*num, temp = \*num, digits = 0, sum = 0;

    while (temp > 0)

    {

        temp /= 10;

        digits++;

    }

    temp = \*num;

    while (temp > 0)

    {

        int rem = temp % 10, power = 1;

        for (int i = 0; i < digits; i++)

        {

            power \*= rem;

        }

        sum += power;

        temp /= 10;

    }

    printf("%d is %s an Armstrong number.\n", original, (original == sum) ? "" : "not");

}

void check\_perfect\_number(int \*num)

{

    int sum = 0;

    for (int i = 1; i <= (\*num) / 2; i++)

    {

        if (\*num % i == 0)

            sum += i;

    }

    printf("%d is %s a Perfect number.\n", \*num, (sum == \*num) ? "" : "not");

}

void factorial\_of\_number(int \*num)

{

    int factorial = 1;

    for (int i = 1; i <= \*num; i++)

    {

        factorial \*= i;

    }

    printf("Factorial of %d is %d\n", \*num, factorial);

}

void check\_strong\_number(int \*num)

{

    int original = \*num, sum = 0;

    int n = \*num;

    while (n > 0)

    {

        int digit = n % 10, fact = 1;

        for (int i = 1; i <= digit; i++)

        {

            fact \*= i;

        }

        sum += fact;

        n /= 10;

    }

    printf("%d is %s a Strong number.\n", original, (sum == original) ? "" : "not");

}

void check\_palindrome\_number(int \*num)

{

    int original = \*num, reversed = 0;

    int n = \*num;

    while (n > 0)

    {

        reversed = reversed \* 10 + (n % 10);

        n /= 10;

    }

    printf("%d is %s a Palindrome number.\n", original, (original == reversed) ? "" : "not");

}

void add\_first\_and\_last\_digits(int \*num)

{

    int lastDigit = \*num % 10;

    int firstDigit = \*num;

    while (firstDigit >= 10)

    {

        firstDigit /= 10;

    }

    printf("Sum of first and last digit: %d + %d = %d\n", firstDigit, lastDigit, firstDigit + lastDigit);

}

// ================= Assignment 4: Functions with Pointers =================

void assignment\_4()

{

    int ch;

    while (1)

    {

        printf("\n---------------- Assignment 4 ----------------\n");

        printf("1) Prime Numbers in Range\n");

        printf("2) Armstrong Numbers in Range\n");

        printf("3) Perfect Numbers in Range\n");

        printf("4) Strong Numbers in Range\n");

        printf("5) Generate Fibonacci Series\n");

        printf("0) Back to Main Menu\n");

        printf("Enter your choice: ");

        scanf("%d", &ch);

        int start = 1, end = 1000, n = 10;

        switch (ch)

        {

        case 1:

            prime\_numbers\_in\_range(&start, &end);

            break;

        case 2:

            armstrong\_numbers\_in\_range(&start, &end);

            break;

        case 3:

            perfect\_numbers\_in\_range(&start, &end);

            break;

        case 4:

            strong\_numbers\_in\_range(&start, &end);

            break;

        case 5:

            generate\_fibonacci\_series(&n);

            break;

        case 0:

            return;

        default:

            printf("Invalid choice. Try again.\n");

        }

    }

}

void prime\_numbers\_in\_range(int \*start, int \*end)

{

    printf("\nQ1. Prime numbers between %d and %d:\n", \*start, \*end);

    for (int num = \*start; num <= \*end; num++)

    {

        if (num < 2)

            continue;

        int is\_prime = 1;

        for (int i = 2; i \* i <= num; i++)

        {

            if (num % i == 0)

            {

                is\_prime = 0;

                break;

            }

        }

        if (is\_prime)

        {

            printf("%d ", num);

        }

    }

    printf("\n--------------------------------\n");

}

void armstrong\_numbers\_in\_range(int \*start, int \*end)

{

    printf("\nQ2. Armstrong numbers between %d and %d:\n", \*start, \*end);

    for (int num = \*start; num <= \*end; num++)

    {

        int original = num, sum = 0, digits = 0, temp = num;

        while (temp > 0)

        {

            temp /= 10;

            digits++;

        }

        temp = num;

        while (temp > 0)

        {

            int digit = temp % 10;

            sum += pow(digit, digits);

            temp /= 10;

        }

        if (sum == original)

        {

            printf("%d ", num);

        }

    }

    printf("\n--------------------------------\n");

}

void perfect\_numbers\_in\_range(int \*start, int \*end)

{

    printf("\nQ3. Perfect numbers between %d and %d:\n", \*start, \*end);

    for (int num = \*start; num <= \*end; num++)

    {

        int sum = 0;

        for (int i = 1; i <= num / 2; i++)

        {

            if (num % i == 0)

            {

                sum += i;

            }

        }

        if (sum == num)

        {

            printf("%d ", num);

        }

    }

    printf("\n--------------------------------\n");

}

void strong\_numbers\_in\_range(int \*start, int \*end)

{

    printf("\nQ4. Strong numbers between %d and %d:\n", \*start, \*end);

    for (int num = \*start; num <= \*end; num++)

    {

        int original = num, sum = 0, temp = num;

        while (temp > 0)

        {

            int digit = temp % 10;

            int factorial = 1;

            for (int i = 1; i <= digit; i++)

            {

                factorial \*= i;

            }

            sum += factorial;

            temp /= 10;

        }

        if (sum == original)

        {

            printf("%d ", num);

        }

    }

    printf("\n--------------------------------\n");

}

void generate\_fibonacci\_series(int \*n)

{

    int first = 0, second = 1;

    printf("\nQ5. First %d terms of Fibonacci Series:\n", \*n);

    for (int i = 0; i < \*n; i++)

    {

        printf("%d ", first);

        int next = first + second;

        first = second;

        second = next;

    }

    printf("\n--------------------------------\n");

}