

Questions

1. Write a C program to print positive integers from 1 to 10.
2. Write a C program to print numbers in reverse order with a difference of 2.

Example: Enter the upper value: 10

Output: 10 8 6 4 2 1

3. Write a C program to count the number of digits in a given integer.

Example: Input: 3452

Output: Number of digits = 4

4. Write a C program to reverse a given integer.

Example: Input: 876

Output: 678

5. Write a C program to print the sum of digits of a number using a for loop.

Example: Given number: 14892

Output: $1 + 4 + 8 + 9 + 2 = 24$

6. Write a C program to check whether a number is a Palindrome or not.

Example: Input: 151

Output: Palindrome Number

7. Write a C program to generate Fibonacci series.

Example: Enter number of elements: 15

Output:

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377

8. Write a C program to generate prime numbers between 1 and 10000.

9. Write a C program to check whether a given number is an Armstrong number or not.

Note:

An Armstrong number is a number that is equal to the sum of the cubes of its digits.

Examples: 0, 1, 153, 370, 371, 407

10. Write a C program to check whether a given number is a Perfect number or not.

Note:

A Perfect number is a positive integer equal to the sum of its proper positive divisors.

Example:

6 → Proper divisors: 1, 2, 3

Sum = $1 + 2 + 3 = 6$

Therefore, 6 is a Perfect number.

Other examples: 28, 496

1. Print Numbers from 1 to 10

Question Explanation:

This program prints positive integers from 1 to 10 using a for loop.

Program Code:

```
#include <stdio.h>
int main() {
    for(int i = 1; i <= 10; i++) {
        printf("%d ", i);
    }
    return 0;
}
```

Code Explanation:

The loop starts from 1 and runs until 10. Each iteration prints the value of i.

2. Reverse Order with Difference of 2

Question Explanation:

This program prints numbers in reverse order with a difference of 2.

Program Code:

```
#include <stdio.h>
int main() {
    int n;
    printf("Enter upper value: ");
    scanf("%d", &n);

    for(int i = n; i >= 1; i -= 2) {
        printf("%d ", i);
    }
    return 0;
}
```

Code Explanation:

The loop starts from n and decreases by 2 each time until it reaches 1.

3. Count Number of Digits

Question Explanation:

This program counts the number of digits in a given integer.

Program Code:

```
#include <stdio.h>

int main() {
    int num, count = 0;
    printf("Enter an integer: ");
    scanf("%d", &num);

    while(num != 0) {
        num = num / 10;
        count++;
    }

    printf("Number of digits = %d", count);
    return 0;
}
```

Code Explanation:

Each division by 10 removes the last digit. The counter increases until the number becomes 0.

4. Reverse a Number

Question Explanation:

This program reverses a given integer.

Program Code:

```
#include <stdio.h>
int main() {
    int num, reverse = 0;
    printf("Enter number: ");
    scanf("%d", &num);

    while(num != 0) {
        reverse = reverse * 10 + num % 10;
        num = num / 10;
    }

    printf("Reversed number = %d", reverse);
    return 0;
}
```

Code Explanation:

The last digit is extracted using modulus and added to the reversed number.

5. Sum of Digits (Using for loop)

Question Explanation:

This program calculates the sum of digits of a number using a for loop.

Program Code:

```
#include <stdio.h>
int main() {
    int num, sum = 0;
    printf("Enter number: ");
    scanf("%d", &num);

    for(; num != 0; num = num / 10) {
        sum += num % 10;
    }

    printf("Sum of digits = %d", sum);
    return 0;
}
```

Code Explanation:

The loop extracts each digit using modulus and adds it to sum.

6. Palindrome Number

Question Explanation:

This program checks whether a number is palindrome or not.

Program Code:

```
#include <stdio.h>

int main() {
    int num, original, reverse = 0;
    printf("Enter number: ");
    scanf("%d", &num);
    original = num;

    while(num != 0) {
        reverse = reverse * 10 + num % 10;
        num = num / 10;
    }

    if(original == reverse)
        printf("Palindrome number");
    else
        printf("Not a palindrome number");

    return 0;
}
```

Code Explanation:

A number is palindrome if original number equals reversed number.

7. Fibonacci Series

Question Explanation:

This program generates Fibonacci series up to n terms.

Program Code:

```
#include <stdio.h>
int main() {
    int n, a = 0, b = 1, next;
    printf("Enter number of elements: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++) {
        printf("%d ", a);
        next = a + b;
        a = b;
        b = next;
    }
    return 0;
}
```

Code Explanation:

Each term is the sum of previous two numbers.

8. Prime Numbers from 1 to 10000

Question Explanation:

This program prints prime numbers between 1 and 10000.

Program Code:

```
#include <stdio.h>
int main() {
    int i, j, isPrime;

    for(i = 2; i <= 10000; i++) {
        isPrime = 1;
        for(j = 2; j*j <= i; j++) {
            if(i % j == 0) {
                isPrime = 0;
                break;
            }
        }
        if(isPrime)
            printf("%d ", i);
    }
    return 0;
}
```

Code Explanation:

A number is prime if it has no divisors other than 1 and itself.

9. Armstrong Number

Question Explanation:

This program checks whether a number is an Armstrong number.

Program Code:

```
#include <stdio.h>

int main() {
    int num, original, sum = 0, digit;
    printf("Enter number: ");
    scanf("%d", &num);
    original = num;

    while(num != 0) {
        digit = num % 10;
        sum += digit * digit * digit;
        num = num / 10;
    }

    if(original == sum)
        printf("Armstrong number");
    else
        printf("Not an Armstrong number");

    return 0;
}
```

Code Explanation:

Sum of cubes of digits is compared with original number.

10. Perfect Number

Question Explanation:

This program checks whether a number is perfect.

Program Code:

```
#include <stdio.h>
int main() {
    int num, sum = 0;
    printf("Enter number: ");
    scanf("%d", &num);

    for(int i = 1; i < num; i++) {
        if(num % i == 0)
            sum += i;
    }

    if(sum == num)
        printf("Perfect number");
    else
        printf("Not a perfect number");

    return 0;
}
```

Code Explanation:

A perfect number equals the sum of its proper divisors.

