Rajiv-0920 initial files 11 minutes ago

2045 lines (1369 loc) · 30.1 KB

Preview Code Blame Raw

Loop Control Instruction

1. WAP. to print first 10 natural numbers.

> Natural numbers

First 10 Natural Numbers are: *1, 2, 3, 4, 5, 6, 7, 8, 9 and 10.*

> Expected Output

```
1 2 3 4 5 6 7 8 9 10
```

> Source Code

```c
#include <stdio.h>

int main(){

    for(int i = 1; i <= 10; i++){
        printf("%d ", i);
    }

    return 0;
}
```

2. WAP. to print first 10 natural numbers in reverse order.

> Natural numbers

First 10 Natural Numbers are: *1, 2, 3, 4, 5, 6, 7, 8, 9 and 10.*

> Expected Output

```
10 9 8 7 6 5 4 3 2 1
```

> Source Code

```c
#include <stdio.h>

int main(){

    for(int i = 1; i >= 1; i--){
        printf("%d ", i);
    }

    return 0;
}
```

3. WAP. to print first 10 odd natural numbers

> Odd numbers

Odd numbers are ***not multiples of 2***.

> Expected Output

```
1 3 5 7 9 11 13 15 17 19
```

> Source Code

```c
#include <stdio.h>

int main(){

    int count = 0;

    for(int i = 1; i <= 100; i++){
        if(i % 2 != 0){
            printf("%d ", i);
            count++;
        }
        if(count == 10)
            break;
    }

    return 0;
}
```

4. WAP. to print first 10 even natural numbers

> Even numbers

Even numbers are ***multiples of 2***.

> Expected Output

```
2 4 6 8 10 12 14 16 18 20
```

> Source Code

```c
#include <stdio.h>

int main(){

    int count = 0;

    for(int i = 1; i<= 100; i++){
        if(i % 2 == 0){
            printf("%d ", i);
            count++;
        }
        if(count == 10)
            break;
    }

    return 0;
}
```

5. WAP. to print first N natural numbers. Value of N is given by user.

> Test Data

```
Enter n number: 15
```

> Expected Output

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

> Source Code

```c
#include <stdio.h>

int main(){
    int n;

    printf("\nEnter n number: ");
    scanf("%d", &n);

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

6. WAP. to print first N odd natural numbers in reverse order. Value of N is given by user.

> Test Data

```
Enter n number: 25
```

> Expected Output

```
First 25 Odd numbers in reverse order = 25 23 21 19 17 15 13 11 9 7 5 3 1
```

> Source Code

```c
#include <stdio.h>

int main(){
    int n;

    printf("Enter n number: ");
    scanf("%d", &n);

    printf("\nFirst %d Odd numbers in reverse order = ", n);
    for(int i = n; i >= 1; i--){
        if(i % 2 != 0)
            printf("%d ", i);
    }

    return 0;
}
```

7. WAP. to calculate sum of first N natural numbers. Value of N is given by user.

> Test Data

```
Enter n number: 5
```

> Expected Output

```
Sum of 5 Natural number = 15
```

> Source Code

```c
#include <stdio.h>

int main(){
    int n, sum = 0;

    printf("Enter n number: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++){
        sum += i;
    }

    printf("\nSum of %d Natural number = %d", n, sum);
    return 0;
}
```

8. WAP. to calculate factorial of a numbers.

> Factorial number

the factorial of 6 is $1 * 2 * 3 * 4 * 5 * 6 = 720$

> Test Data

```
Enter n number: 6
```

> Expected Output

```
Factorial of 6 = 720
```

> Source Code

```c
#include <stdio.h>

int main(){
    int n, fact = 1;

    printf("\nEnter a number:");
    scanf("%d", &n);

    for(int i = n; i >= 1; i--){
        fact *= i;
    }

    printf("Factorial of %d = %d", n, fact);

    return 0;
}
```

9. WAP. to calculate x^y. Values of x and y are given by user.

> Test Data

```
Enter x and y: 3 5
```

> Expected Output

```
Value of x and y = 243
```

> Source Code

```c
#include <stdio.h>

int main(){
    int x, y, value = 1;

    printf("Enter x and y: ");
    scanf("%d %d", &x, &y);
```

```
    for(int i = 1; i <= y; i++){
        value *= x;
    }

    printf("Value of x and y = %d", value);
    return 0;
}
```

10. WAP. to count number of digits in a given number.

> Test Data

```
Enter a number: 123
```

> Expected Output

```
3 number of digits in a given number.
```

> Source Code

```
#include <stdio.h>

int main(){
    int count = 0, num;

    printf("Enter a number: ");
    scanf("%d", &num);

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

    printf("%d number of digits in a given number.", count);

    return 0;
}
```

11. WAP. to calculate sum of the digits of a given number.

> Test Data

```
Enter a number: 352
```

> Expected Output

```
Sum = 10
```

> Source Code

```c
#include <stdio.h>

int main(){
    int num, sum = 0;

    printf("Enter a number: ");
    scanf("%d", &num);

    while(num > 0){
        sum += num % 10;
        num /= 10;
    }

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

12. WAP. to reverse a number.

> Test Data

```
Enter a number: 352
```

> Expected Output

```
Reverse number = 253
```

> Source Code

```c
#include <stdio.h>

int main(){
    int num, rev = 0, rem;

    printf("Enter a number: ");
    scanf("%d", &num);

    while(num > 0){
        rem = num % 10;
        rev = rev * 10 + rem;
        num /= 10;
    }

    printf("Reverse number = %d", rev);

    return 0;
}
```

13. WAP. to check whether the two given numbers are reverse of each other or not.

> Test Data

```
Enter two number: 123 321
```

> Expected Output

```
YES
```

> Source Code

```c
#include <stdio.h>

int main(){
    int num1, num2, rev = 0;

    printf("Enter a number: ");
    scanf("%d %d", &num1, &num2);

    while(num1 > 0){
        rev = rev * 10 + num1 % 10;
        num1 /= 10;
    }

    (rev == num2) ? printf("YES") : printf("NO");
    return 0;
}
```

14. WAP. to check whether a given number is Prime or not.

> Test Data

```
Enter a number: 5
```

> Expected Output

```
Prime number
```

> Source Code

```c
#include <stdio.h>

int main(){
    int num, flag = 1;

    printf("Enter a number: ");
    scanf("%d", &num);

    for(int i = 2; i <= num / 2; i++){
        if(num % i == 0){
            flag = 0;
            break;
        }
    }

    if(num == 1)
```

```
        flag = 0;

    if(flag == 1)
        printf("Prime number");
    else
        printf("Not a Prime number");
    return 0;
}
```

15. WAP. to print all prime numbers between two given numbers.

> Test Data

```
Enter the range between two number: 25 50
```

> Expected Output

```
29 31 37 41 43 47
```

> Source Code

```
#include <stdio.h>

int main(){
    int n1, n2, flag;

    printf("Enter the range between two numbers: ");
    scanf("%d %d", &n1, &n2);

    printf("\nPrime numbers in the given range are = ");
    for(int i = n1; i <= n2; i++){
        flag = 1;
        for(int j = 2; j <= i / 2; j++){
            if(i % j == 0){
                flag = 0;
                break;
            }
        }
        if(i == 1)
            flag = 0;
        if(flag == 1)
            printf("%d ", i);
    }
    return 0;
}
```

16. WAP. to print all prime numbers in first 1000 natural numbers.

> Source Code

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

```c
int main(){
    int flag;

    printf("\nPrime numbers = ");
    for(int i = 1; i <= 1000; i++){
        flag = 1;
        for(int j = 2; j <= i / 2; j++){
            if(i % j == 0){
                flag = 0;
                break;
            }
        }
        if(i == 1)
            flag = 0;
        if(flag == 1)
            printf("%d ", i);
    }

    return 0;
}
```

17. WAP. to find GCD of two numbers.

> Greatest Common Divisor - GCD

Find the ***GCD*** of ***15 and 20.***

```
Factor of 15 = 1, 3, 5, 15
Factor of 20 = 1, 2, 4, 5, 10, 20

GCD of 15 and 20 = 5
```

> Test Data

```
Enter two numbers:  36 60
```

> Expected Output

```
GCD = 12
```

> Source Code

```c
#include <stdio.h>

int main(){
    int num1, num2, smaller, gcd;

    printf("\nEnter two numbers: ");
    scanf("%d %d", &num1, &num2);

    smaller = num1;

    if(num1 > num2)
        smaller = num2;
```

```
    while(smaller != 0){
        if(num1 % smaller == 0 && num2 % smaller == 0){
            gcd = smaller;
            break;
        }
        smaller--;
    }

    printf("\nGCD = %d", gcd);

    return 0;
}
```

18. WAP. to find LCM of two numbers.

> Test Data

```
Enter two number: 12 52
```

> Expected Output

```
LCM = 156
```

> Source Code

```
#include <stdio.h>

int main(){
    int num1, num2, greater, lcm;

    printf("Enter two number: ");
    scanf("%d %d", &num1, &num2);

    greater = num1;

    if(num1 < num2)
        greater = num2;

    while(1){
        if(greater % num1 == 0 && greater % num2 == 0){
            lcm = greater;
            break;
        }
        greater++;
    }

    printf("LCM = %d", lcm);

    return 0;
}
```

19. WAP. to print the following 1, 4, 7, 10 ..... 40.

> Expected Output

```
1 4 7 10 13 16 19 22 25 28 31 34 37 40
```

> Source Code

```c
#include <stdio.h>

int main(){
    int i = 1;

    while(i <= 40){
        printf("%d ", i);
        i += 3;
    }

    return 0;
}
```

20. WAP. to print the following 1, -4, 7, -10 ..... -40.

> Expected Output

```
1 -4 7 -10 13 -16 19 -22 25 -28 31 -34 37 -40
```

> Source Code

```c
#include <stdio.h>

int main(){
    int i = 1, s = 1;

    while(i <= 40){
        printf("%d ",i * s);
        i += 3;
        s *= -1;
    }

    return 0;
}
```

21. WAP. to print table of a given number.

> Test Data

```
Enter a number: 19
```

> Expected Output

```
19 x 1 = 19
19 x 2 = 38
19 x 3 = 57
19 x 4 = 76
19 x 5 = 95
19 x 6 = 114
19 x 7 = 133
19 x 8 = 152
19 x 9 = 171
19 x 10 = 190
```

> Source Code

```c
#include <stdio.h>

int main(){
    int n, i = 1;

    printf("Enter a number: ");
    scanf("%d", &n);

    while(i <= 10){
        printf("\n%d x %d = %d", n, i, n * i);
        i++;
    }

    return 0;
}
```

22. WAP. to find a 3 digit number is Armstrong or not. (without using pow function)

> Armstrong number

0, 1, 153, 370, 371 and 407 are the Armstrong numbers.

$153 = 1 * 1 * 1 + 5 * 5 * 5 + 3 * 3 * 3$

$153 = 1 + 125 + 27$

$153 = 153$

> Test Data

```
Enter a number: 153
```

> Expected Output

```
153 is a Armstrong number
```

> Source Code

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

```c
int main(){
    int num, rem, arm = 0, temp;

    printf("Enter a number: ");
    scanf("%d", &num);

    temp = num;

    while(temp != 0){
        rem = temp % 10;
        arm += rem * rem * rem;
        temp /= 10;
    }

    if(num == arm)
        printf("%d is a Armstrong number.", num);
    else
        printf("%d is not a Armstrong number.", num);


    return 0;
}
```

23. WAP. to Find Number Is Armstrong Or Not

> Armstrong number

0, 1, 153, 370, 371 and 407 are the Armstrong numbers.

$153 = 1 * 1 * 1 + 5 * 5 * 5 + 3 * 3 * 3$

$153 = 1 + 125 + 27$

$153 = 153$

> Test Data

```
Enter a number: 9474
```

> Expected Output

```
9474 is a Armstrong number
```

> Source Code

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

int main(){
    int n, count = 0, temp, arm = 0, rem;

    printf("Enter a number: ");
    scanf("%d", &n);

    temp = n;
```

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

    temp = n;

    while(temp != 0){
        rem = temp % 10;
        arm += pow(rem, count);
        temp /= 10;
    }

    printf("Arm = %d n = %d", arm, n);

    if(arm == n)
        printf("%d is a Armstrong number.", n);
    else
        printf("%d is not an Armstrong number.", n);

    return 0;
}
```

24. WAP. to check whether it is a Palindrome number or not.

> Palindrome number

that remains the ***same when its digits are reversed.***

> Test Data

```
Enter a number: 35153
```

> Expected Output

```
35153 is a Palindrome number
```

> Source Code

```
#include <stdio.h>

int main(){
    int num, rev = 0, rem, temp;

    printf("Enter a number: ");
    scanf("%d", &num);

    temp = num;

    while(temp != 0){
        rem = temp % 10;
        rev = rev * 10 + rem;
        temp /= 10;
    }

    if(rev == num)
```

```
        printf("%d is a Palindrome number.", num);
    else
        printf("%d is not a Palindrome number.", num);

    return 0;
}
```

25. WAP. to print Armstrong number between 1 to n.

> Test Data

```
Enter the value of n: 10000
```

> Expected Output

```
1 2 3 4 5 6 7 8 9 153 370 371 407 1634 8208 9474
```

> Source Code

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

int main(){
    int n, count = 0, temp, rem, arm = 0;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++){
        arm = 0;
        count = 0;
        temp = i;
        while(temp != 0){
            count++;
            temp /= 10;
        }
        temp = i;
        while(temp != 0){
            rem = temp % 10;
            arm += pow(rem, count);
            temp /= 10;
        }

        if(i == arm)
            printf("%d ", i);

    }
    return 0;
}
```

26. WAP. to print n fibonacci series.

> Fibonacci numbers

The Fibonacci numbers are the numbers in the following integer sequence: ***0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ……..***

> Test Data

```
Enter the value of n: 5
```

> Expected Output

```
0 1 1 2 3
```

> Source Code

```c
#include <stdio.h>

int main(){
    int n, n1 = 0, n2 = 1, sum = 0;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    for(int i = 0; i <= n; i++){
        if(i <= 1)
            printf("%d ", i);
        else{
            sum = n1 + n2;
            n1 = n2;
            n2 = sum;
            printf("%d ", sum);
        }
    }

    return 0;
}
```

27. WAP. to print n Square number series ex:- 1, 4, 9, 16, 25, 36, 48;

> Test Data

```
Enter the value of n: 6
```

> Expected Output

```
1, 4, 9, 16, 25, 36
```

> Source Code

```c
#include <stdio.h>

int main(){
    int n;
```

```
    printf("Enter the value of n: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++){
        printf("%d ", i * i);
    }

    return 0;
}
```

28. WAP. to print the following series 1, 8, 27, 64, 125 , 216....

> Test Data

```
Enter the value of n: 5
```

> Expected Output

```
1 8 27 64 125
```

> Source Code

```
#include <stdio.h>

int main(){
    int n;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++){
        printf("%d ", i * i * i);
    }

    return 0;
}
```

29. WAP. to print the following 1, -1, 1, -1, 1, ...

> Test Data

```
Enter the value of n: 5
```

> Expected Output

```
1 -1 1 -1 1
```

> Source Code

```c
#include <stdio.h>

int main(){
    int n, num = 1;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++){
        printf("%d ", num);
        num *= -1;
    }

    return 0;
}
```

30. WAP. to print the following 1, 1, 2, 4, 7, 13, 24, ... (Lucas series)

> Test Data

```
Enter the value of n: 5
```

> Expected Output

```
1 1 2 4 7
```

> Source Code

```c
#include <stdio.h>

int main(){
    int n, n1 = 1, n2 = 1, n3 = 0, temp;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++){
        if(i <= 2){
            printf("%d ", 1);
        } else {
            temp = n1 + n2 + n3;
            n3 = n2;
            n2 = n1;
            n1 = temp;

            printf("%d ", n1);
        }
    }
    return 0;
}
```

31. WAP. to print 1 to n Triangular number sequence.

> Triangular number

```
1, 3, 6, 10, 15, 21, 28, 36, 45,...
```

> Test Data

```
Enter the value of n: 10
```

> Expected Output

```
1 3 6 10 15 21 28 36 45 55
```

> Source Code

```c
#include <stdio.h>

int main(){
    int sum = 0, n;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++){
        sum += i;
        printf("%d ", sum);
    }

    return 0;
}
```

32. WAP. to print the following sequence 1, 2, 0, 3, -1, 4, -2, 5, -3, 6, -4, 7, ...n

> Test Data

```
Enter the value of n: 10
```

> Expected Output

```
1 2 0 3 -1 4 -2 5 -3 6
```

> Source Code

```c
#include <stdio.h>

int main(){
    int n, n1 = 2, n2 = 1;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++){
```

```
        if(i % 2 == 0)
            printf("%d ", n1++);
        else
            printf("%d ", n2--);
    }
    return 0;
}
```

33. WAP. to print the following sequence 1, 5, 2, 4, 3, 3, 4, 2, 5, 1

> Test Data

```
Enter the value of n: 20
```

> Expected Output

```
1 10 2 9 3 8 4 7 5 6 6 5 7 4 8 3 9 2 10 1
```

> Source Code

```
#include <stdio.h>

int main(){
    int n, n1 = 1, n2;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    n2 = n / 2;

    for(int i = 1; i <= n; i++){
        if(i % 2 == 0)
            printf("%d ", n2--);
        else
            printf("%d ", n1++);
    }

    return 0;
}
```

34. WAP. to print the following sequence 1, 5, 2, 6, 3, 7, 4, 8, 5, 9

> Test Data

```
Enter the value of n: 20
```

> Expected Output

```
1 10 2 11 3 12 4 13 5 14 6 15 7 16 8 17 9 18 10 19
```

> Source Code

```c
#include <stdio.h>

int main(){
    int n, n1, n2;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    n1 = 1;
    n2 = n / 2;

    for(int i = 1; i <= n; i++){
        if(i % 2 == 0)
            printf("%d ", n2++);
        else
            printf("%d ", n1++);
    }

    return 0;
}
```

35. WAP. to input a number and print sum of its even and odd digits.

> Test Data

```
Enter any number: 3875698
```

> Expected Output

```
Sum of even number = 22 and odd number = 24
```

> Source Code

```c
#include <stdio.h>

int main(){
    int evenSum = 0, oddSum = 0, num, rem;

    printf("Enter any number: ");
    scanf("%d", &num);

    while(num != 0){
        rem = num % 10;
        rem % 2 == 0 ? evenSum += rem : (oddSum += rem);
        num /= 10;
    }

    printf("Sum of even number = %d and odd number = %d", evenSum, oddSum);

    return 0;
}
```

36. WAP. to input any number and print its factors.

> Test Data

```
Enter any number: 36
```

> Expected Output

```
1 2 3 4 6 9 12 18 36
```

> Source Code

```c
#include <stdio.h>

int main(){
    int num;

    printf("Enter any number: ");
    scanf("%d", &num);

    for(int i = 1; i <= num; i++){
        if(num % i == 0)
            printf("%d ", i);
    }
    return 0;
}
```

37. WAP. to input a number and check it for Strong number.

> Strong number

Strong number is a special number ***whose sum of the factorial of digits is equal to the original number.*** For Example: ***145 is strong number.*** Since, *1! + 4! + 5! = 145.*

> Test Data

```
Enter any number: 40585
```

> Expected Output

```
40585 is a strong number
```

> Source Code

```c
#include <stdio.h>

int main(){
    int num, sum = 0, temp, fact = 1, rem;

    printf("Enter any number: ");
    scanf("%d", &num);
```

```
    temp = num;

    while(temp != 0){
        fact = 1;
        rem = temp % 10;
        for(int i = rem; i >= 1; i--){
            fact *= i;
        }
        sum += fact;
        temp /= 10;
    }

    if(sum == num)
        printf("%d is a strong number", num);
    else
        printf("%d is not a strong number", num);
    return 0;
}
```

38. WAP. to input n number and print all strong numbers in between 1 to n.

> Strong number

Strong number is a special number ***whose sum of the factorial of digits is equal to the original number.*** For Example: ***145 is strong number.*** Since, *1! + 4! + 5! = 145.*

> Test Data

```
Enter the value of n: 50000
```

> Expected Output

```
1 2 145 40585
```

> Source Code

```
#include <stdio.h>

int main(){
    int n, temp, fact, sum, rem;

    printf("\nEnter the value of n: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++){
        temp = i;
        sum = 0;
        while(temp != 0){
            fact = 1;
            rem = temp % 10;
            for(int j = rem; j >= 1; j--)
                fact *= j;
            sum += fact;
            temp /= 10;
        }
```

```
        if(sum == i)
            printf("%d ", i);
    }
    return 0;
}
```

39. WAP. to input a number and print the min and max digit of the number.

> Test Data

```
Enter a number: 312
```

> Expected Output

```
Min = 1 Max = 3
```

> Source Code

```
#include <stdio.h>

int main(){
    int num, min = 9, max = 0, rem;

    printf("Enter a number: ");
    scanf("%d", &num);

    while(num != 0){
        rem = num % 10;
        if(rem > max)
            max = rem;
        if(rem < min)
            min = rem;
        num /= 10;
    }

    printf("Min = %d Max = %d", min, max);
    return 0;
}
```

40. WAP. to input a number and make a new number by adding 1 with individual digit.

> Test Data

```
Enter a number: 239
```

> Expected Output

```
Sum = 350
```

> Source Code

```c
#include <stdio.h>

int main(){
    int num, count = 0, sum, temp;

    printf("Enter a number: ");
    scanf("%d", &num);

    temp = num;

    while(temp != 0){
        count = count * 10 + 1;
        temp /= 10;
    }

    sum = num + count;

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

41. WAP. to check perfect number or not.

> Perfect number

A perfect number is a positive integer that is ***equal to the sum of its factors except for the number itself.***

The smallest perfect number is 6, which is the sum of its factors: 1, 2, and 3.

Note that this sum does not include the number itself which is also a factor of itself.

> Test Data

```
Enter a number: 496
```

> Expected Output

```
Perfect number
```

> Source Code

```c
#include <stdio.h>

int main(){
    int num, sum = 0;

    printf("Enter a number: ");
    scanf("%d", &num);

    for(int i = 1; i <= num / 2; i++){
        if(num % i == 0)
            sum += i;
```

```
    }

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

    return 0;
}
```

42. WAP. to print all integers that are not divisible by either 2 or 3 and lie between 1 and n.

> Test Data

```
Enter the value of n: 20
```

> Expected Output

```
1 5 7 11 13 17 19
```

> Source Code

```
#include <stdio.h>

int main(){
    int n;

    printf("Enter the value of n: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++){
        if(i % 2 != 0 && i % 3 != 0)
            printf("%d ", i);
    }
    return 0;
}
```

43. WAP. to input a number and print the position of individual digit.

> Test Data

```
Enter a number: 123
```

> Expected Output

```
100 20 3
```

> Source Code

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

int main(){
    int count = 0, rem, num, t;

    printf("Enter a number: ");
    scanf("%d", &num);

    t = num;

    while(t){
        count++;
        t /= 10;
    }

    for(int i = count - 1; i >= 0; i--){
        rem = num / pow(10, i);
        rem = rem * pow(10, i);
        printf("%d ", rem);
        num = num % (int)pow(10, i);
    }

    return 0;
}
```

44. WAP. to express a given number as a sum of two prime numbers. Print all possible solutions.

> Test Data

```
Enter a number: 34
```

> Expected Output

```
3 + 31 = 34
5 + 29 = 34
11 + 23 = 34
15 + 19 = 34
17 + 17 = 34
```

> Source Code

```c
#include <stdio.h>

int main(){
    int num, c, c1, subVal;

    printf("Enter a number: ");
    scanf("%d", &num);

    for(int i = 2; i <= num / 2; i++){
        c = 1;
        if(i == 2){
            c = 1;
```

```
        } else if(i % 2 == 0){
            c = 0;
        }

        if(c == 1){
            subVal = num - i;
            c1 = 1;
            for(int j = 2; j <= subVal/2; j++){
                if(subVal % j == 0){
                    c1 = 0;
                    break;
                }
            }
            if(c1 == 1){
                printf("\n%d + %d = %d", i, subVal, i + subVal);
            }
        }
    }
    if(c1 != 1){
        printf("Sum of Prime possible number is 0");
    }
    return 0;
}
```

45. WAP. to input a number and check it is square of 2 or not

> Test Data

```
Enter any number: 64
```

> Expected Output

```
64 is a square of 2
```

> Source Code

```
#include <stdio.h>

int main(){
    int n, s = 1;

    printf("Enter any number: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++){
        s *= 2;
        if(s == n){
            break;
        }
    }

    if(s == n){
        printf("%d is a square of 2", n);
    } else{
        printf("%d is not a square of 2", n);
    }
```

```
    return 0;
}
```

46. WAP. to swap first and last digit of a number

> Test Data

```
Enter any number: 452
```

> Expected Output

```
252
```

> Source Code

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

int main(){
    int num, temp, count = 0, swappedNum, firstDigit, lastDigit;

    printf("Enter any number: ");
    scanf("%d", &num);

    temp = num;

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

    lastDigit = num % 10;
    firstDigit = num / (int)pow(10, count - 1);
    swappedNum = num % (int)pow(10, count - 1);
    swappedNum -= lastDigit;
    swappedNum += firstDigit;
    swappedNum += lastDigit * (int)pow(10, count - 1);

    printf("%d", swappedNum);
    return 0;
}
```

47. WAP. to find all prime factors of a number.

> Test Data

```
Enter any number: 88
```

> Expected Output

```
2 11
```

Source Code

```c
#include <stdio.h>

int main(){
    int num, count;

    printf("Enter any number: ");
    scanf("%d", &num);

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

        if(count == 2){
            printf("%d ", i);
        }
    }
    return 0;
}
```

48. WAP. to convert Decimal to Binary number system.

Test Data

```
Enter any number: 10
```

Expected Output

```
Decimal number = 2
```

Source Code

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

int main(){
    int dec = 0, i = 0, rem;
    long long bin;
    printf("Enter a number: ");
    scanf("%lld", &bin);

    while(bin != 0){
        rem = bin % 10;
        dec += rem * (int)pow(2, i);
        i++;
```

```
        bin /= 10;
    }

    printf("Decimal number = %d", dec);
    return 0;
}
```

49. WAP. to convert Octal to Decimal number system.

> Test Data

```
Enter any number: 16
```

> Expected Output

```
Decimal number = 14
```

> Source Code

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

int main(){
    int octal, dec = 0, i = 0, rem;

    printf("Enter a number: ");
    scanf("%d", &octal);

    while(octal != 0){
        rem = octal % 10;
        dec += rem * (int)pow(8, i);
        i++;
        octal /= 10;
    }

    printf("Decimal number = %d", dec);
    return 0;
}
```

50. WAP. to convert Decimal to Binary number system.

> Test Data

```
Enter any number: 10
```

> Expected Output

```
Binary number = 1010
```

> Source Code

```c
#include <stdio.h>

int main(){
    int dec, rem, i = 1;
    long long bin = 0;

    printf("Enter any number: ");
    scanf("%d", &dec);

    while(dec != 0){
        rem = dec % 2;
        bin += rem * i;
        i *= 10;
        dec /= 2;
    }

    printf("Binary number = %d", bin);
    return 0;
}
```

51. WAP. to convert Decimal to Octal number system.

> Test Data

```
Enter any number: 14
```

> Expected Output

```
Octal number = 16
```

> Source Code

```c
#include <stdio.h>

int main(){
    int dec, rem, i = 1, octal = 0;

    printf("Enter any number: ");
    scanf("%d", &dec);

    while(dec != 0){
        rem = dec % 8;
        octal += rem * i;
        i *= 10;
        dec /= 8;
    }

    printf("Octal number = %d", octal);
    return 0;
}
```

52. Write a C program to enter a number and print it in words.

Test Data

```
Enter a number: 1050
```

Expected Output

```
One Zero Five Zero
```

Source Code

```c
#include <stdio.h>

int main()
{
    int reverse(int);
    int totalDigit(int);
    int num, n, td, ne;

    printf("Enter a number: ");
    scanf("%d", &num);

    td = totalDigit(num);
    n = reverse(num);
    ne = totalDigit(n);

    while (n != 0)
    {
        int temp = n % 10;
        switch (temp)
        {
        case 0:
            printf("Zero ");
            break;
        case 1:
            printf("One ");
            break;
        case 2:
            printf("Two ");
            break;
        case 3:
            printf("Three ");
            break;
        case 4:
            printf("Four ");
            break;
        case 5:
            printf("Five ");
            break;
        case 6:
            printf("Six ");
            break;
        case 7:
            printf("Seven ");
            break;
        case 8:
            printf("Eight ");
            break;
        case 9:
            printf("Nine ");
```

```
            break;
        default:
            printf("Not in the list");
        }
        n /= 10;
    }
    while (td > ne)
    {
        printf("Zero ");
        ne++;
    }
    return 0;
}

int reverse(int num)
{
    int rev = 0, rem;
    while (num != 0)
    {
        rem = num % 10;
        rev = rev * 10 + rem;
        num /= 10;
    }
    return rev;
}
int totalDigit(int num)
{
    int count = 0;
    while (num)
    {
        count++;
        num /= 10;
    }
    return count;
}
```

Hi 👋, I'm Rajiv Kumar

A passionate frontend developer from India

🌐 Connect with me:

GITHUB TWITTER LINKEDIN

💻 Tech Stack: